ГУАП

КАФЕДРА 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук, доц.		О. О. Жаринов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2

ОСНОВЫ ОБРАБОТКИ АУДИОСИГНАЛОВ СРЕДСТВАМИ PYTHON

по курсу: Мультимедиа технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4016		А. А. Абрамочкин
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Получить представление о принципах формирования аудиосигналов и приобрести навыки работы со звуковыми файлами с использованием MATLAB.

Индивидуальный вариант

Номер варианта: **4.**

Текст задания: применить фильтр высоких частот с $f_{ниж}$ = 3400 Гц.

Теоретические сведения

Работа с аудиосигналами может показаться простой, однако это такая же сложная работа, как и оптика.

В обработке аудиосигналов приходится прибегать к фильтрации сигнала. Фильтр может использоваться в абсолютно разных целях. Например, в звукозаписи ‘low cut’ фильтрацию используют, чтобы отрезать ненужные и очень низкие звуки, дабы они не мешали голосу и не создавали помех. Обычно человеческий голос звучит от 120 Гц до 20 кГц, таким образом низкочастотный шум от 20 Гц до 120 Гц не несет никакого смысла.

Чаще всего для получения информации о частотных характеристиках аудиосигналов используют метод быстрого преобразования Фурье.

Точно так же его применяют и в эквализации. К примеру, в записанном голосе не хватает четкости и разборчивости слов. В таком случае прибегают к подъему громкости частот в полосе от 3 кГц до 5 кГц. Это область отвечает за четкость в человеческом голосе, хоть сама по себе и не является мощной или широкой.

Ход работы

В процессе выполнения первой лабораторной работы из данного курса пакет Matlab не показался мне удобной средой для работы. Поэтому я прибегу к работе над фильтрацией аудио на языке Python с использование таких библиотек, как ‘matplotlib’, ‘soundfile’ и ‘numpy’.

В качестве файла для фильтрации я взял свою любимую композицию под название ‘Kaytranada - You`re the one’. Ниже можно наблюдать программный код, использованный в

данной лабораторной работе, а также таблицу с использованными переменными для удобства проверки работы.

Таблица переменных

Таблица 1 - Список переменных

Название переменной	Описание
original_signal	Исходный сигнал, считанный из mp3 - файла
sample_rate	Частота дискретизации
equalizer	Функция визуализации АЧХ
signal	Исходный сигнал, пропущенный через функцию equalizer
title	Переменная для названий графиков
fft_spectrum	Амплитудный спектр сигнала, полученный после преобразования Фурье
amplitude_db	Амплитудный спектр в Дб
frequency_axis	Частотная ось
highpass_filter	Функция для фильтрации сигнала
min_frequency	Отсекаемая частота
min_frequency_index	Отсекаемая частота, вычисленная с помощью ‘sample_rate’ и длины входного сигнала
fft_input	Двумерное преобразование Фурье входного сигнала
fft_output	Двумерное преобразование Фурье выходного сигнала после фильтрации
output_signal	Аудио сигнал, полученный после применения ФВЧ

Программный код

Листинг 1 - Программный код

```
import soundfile as sf
import matplotlib.pyplot as plt
import numpy as np
```

```
# Инициализация функций

# Функция для визуализации частотных характеристик
def equalizer(sample_rate, signal, title: str):
    # Вычисление амплитудного спектра сигнала
    fft_spectrum = np.fft.fft(signal)
    # Преобразование сигнала в Дб
    amplitude_db = 20 * np.log10(np.abs(fft_spectrum))

    # Генерация частотной оси
    frequency_axis = np.fft.fftfreq(len(fft_spectrum), 1 / sample_rate)

    # Построение графика частотных характеристик (Эквалайзер)
    plt.figure()
    plt.plot(frequency_axis[:len(frequency_axis) // 2], amplitude_db[:len(frequency_axis) // 2])
    plt.xlabel('Частота, Гц)')
    plt.ylabel('Амплитуда, Дб')
    plt.title(f'Частотные характеристики - {title}')
    # Ограничение частотных характеристик свойствами слуха
    plt.xlim(20, 20000)
    plt.grid()
    # Создание логарфмической шкалы на оси X
    plt.semilogx()


# Функция фильтрации высоких частот
def highpass_filter(min_frequency, signal, sample_rate):
    # Переводим Гц в индексы массива
    min_frequency_index = int(len(signal) * min_frequency / sample_rate)

    # Создаем спектр входного и выходного сигнала
    fft_input = np.fft.fft2(signal)
    fft_output = np.zeros_like(fft_input) + 10 ** (-10 ** 10)

    # Реализация фильтра
    for i in range(1, min_frequency_index):
        fft_output[i, :] = fft_input[i, :]
        fft_output[len(signal) - i, :] = fft_input[len(signal) - i, :]
    fft_output[0, :] = fft_input[0, :]
    fft_output = fft_input - fft_output

    # Обратное преобразование в аудиосигнал
    output_signal = np.real(np.fft.ifft2(fft_output))

    return output_signal
```

```
# Применение функций

# Считывание файла и визуализация АЧХ
original_signal, sample_rate = sf.read("Kaytranada - You're The One (Ft. Syd)-.mp3")
equalizer(sample_rate, original_signal[:, 0], '(до фильтрации)')

# Фильтрация сигнала и визуализация АЧХ
filtered_signal = highpass_filter(3400, original_signal, sample_rate)
equalizer(sample_rate, filtered_signal[:, 0], '(после фильтрации)')

# Вывод частотных хар-ик до и после, сохранение отфильтрованного файла
plt.show()
sf.write("output_audio.wav", filtered_signal, sample_rate)
```

Визуализация

В ходе работы программы были получены графики АЧХ до и после фильтрации.

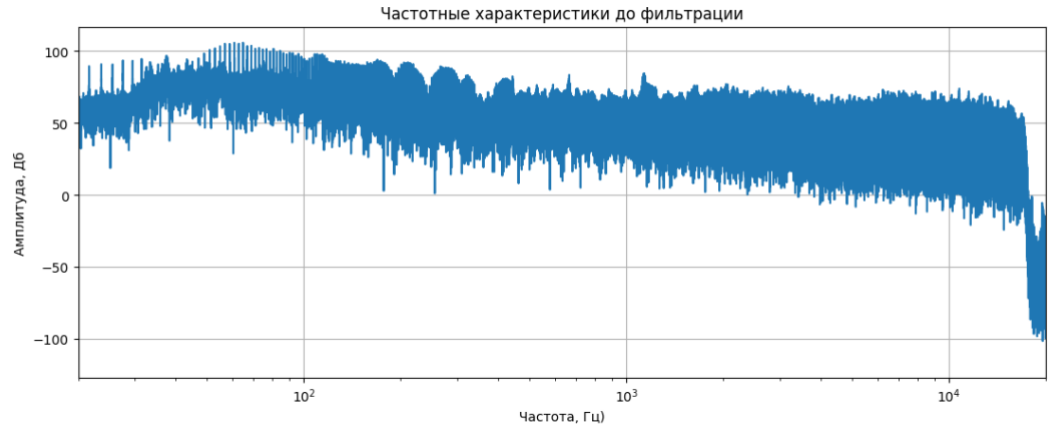

Рисунок 1 - Частотные характеристики сигнала до фильтрации

Рисунок 2 - Частотные характеристики сигнала после фильтрации

Взглянув на рисунки 1 и 2, сразу же делается вывод, что фильтрация прошла успешно. А прослушивая записанный сигнал, сразу можно выделить хруст и песочность высоких частот, так как низкие были отфильтрованы и больше не мешают фокусу на высоких частотах.

Заключение

В результате выполнения данной лабораторной работы были получены навыки работы с фильтрацией звука. Это было достигнуто путем применения метода преобразования Фурье, который встроен в библиотеку ‘numpy’.

Были разработаны две функции: по визуализации частотных характеристик сигнала и фильтрации сигнала. Графики, полученные в результате работы, ясно дают понять, что фильтрация прошла успешна. Записанный отфильтрованный сигнал так же отражает суть фильтрации.