Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторным работам**

Дисциплина: Телекоммуникационные технологии.

Выполнил студент гр. 5130901/10101 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Л. Симоновский (подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Богач (подпись)

“30” октябрь 2024 г.

Санкт-Петербург

2024

Оглавление

[1. Лабораторная работа 1. Сигналы и звуки. 2](#_Toc157941480)

[1.1. Упражнение 1.2. 2](#_Toc157941481)

[1.2. Упражнение 1.3. 5](#_Toc157941482)

[1.3. Упражнение 1.4. 8](#_Toc157941483)

[2. Приложение: 9](#_Toc157941484)

# Лабораторная работа 1. Сигналы и звуки.

## Упражнение 1.2.

Скачаем с сайта [https://freesound.org/](https://freesound.org/%20) образец звука и различными способами исследуем его. Для удобной работы с сигналами здесь, и в дальнейших работах будем использовать библиотеку thinkdsp.

Откроем скачанный файл, нормализуем и выведем на экран. Код будет выглядеть следующим образом:



Результат выполнения кода выглядит следующим образом:

Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

Рис. 1.1. Спектрограмма аудио файла.

Данный отрезок слишком длинный, выделим из него отрезок длинной пол секунды, начиная с 40 секунды аудио файла. Выведем полученный сегмент на экран, используя следующий код:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

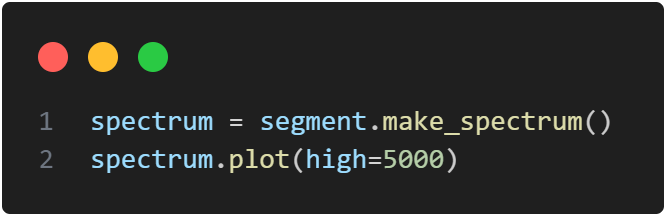
Спектрограмма заданного сегмента выглядит следующим образом:

Изображение выглядит как снимок экрана, текст, График, диаграмма

Автоматически созданное описание

Рис. 1.2. Спектрограмма аудио файла с 40.0 по 40.5 секунды.

Разложим полученный отрезок в спектр и выведем на экран. Код будет выглядеть следующем образом:



Этот код выведет спектр до 5000 частоты т.к. далее частоты равны примерно нулю:

Изображение выглядит как снимок экрана, График, диаграмма, линия

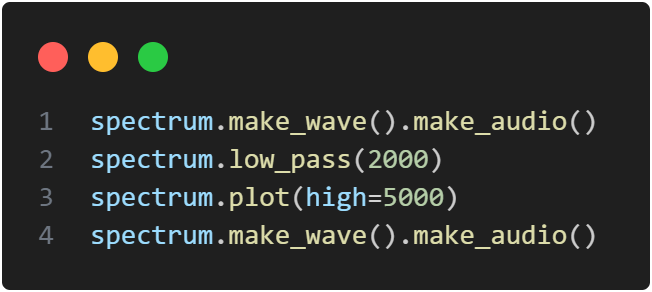
Автоматически созданное описание

Рис. 1.3. Результат разложения сегмента в спектр.

Доминантной частотой в этом отрывке является 98 Гц.

Теперь поэкспериментируем с функциями high\_pass, low\_pass и band\_stop, которые фильтруют гармоники.

Начнем с low\_pass:



Данный код сохраняет музыкальный фрагмент (для дальнейшего сравнения), после чего применяет функцию low\_pass и выводит его спектр на экран, а также опять сохраняет фрагмент.

Полученный спектр выглядит следующим образом:

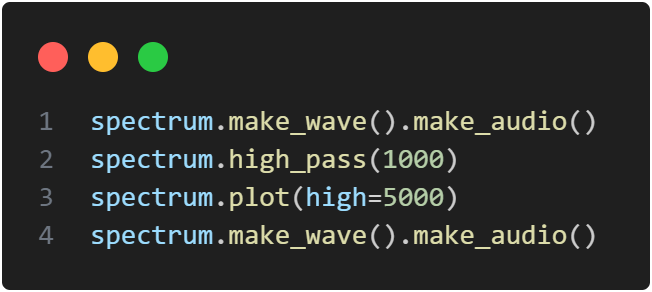
Изображение выглядит как снимок экрана, текст, График, диаграмма

Автоматически созданное описание

Рис. 1.4. Спектр фрагмента после применения low\_pass.

Как видно из рисунка выше, данная функция полностью убрала частоты, выше 2000. Таким образом звук стал более «глухим» и «отдаленным».

Теперь к исходному сегменту применим метод high\_pass:



Полученный спектр имеет следующий вид:

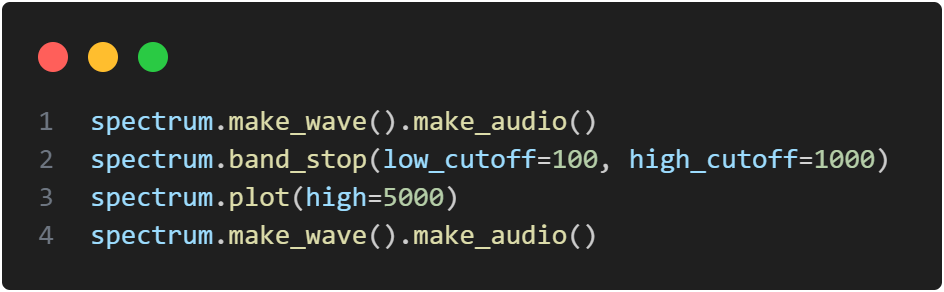
Изображение выглядит как снимок экрана, текст, График, диаграмма

Автоматически созданное описание

Рис. 1.5. Спектр фрагмента после применения high\_pass.

Как видно по спектру, эта функция убирает все частоты ниже заданной. Таким образом звук сильно поменял свое звучание, став более шипящим и менее глубоким.

И последняя функция band\_stop:



Полученный спектр выглядит следующим образом:

Изображение выглядит как текст, снимок экрана, График, диаграмма

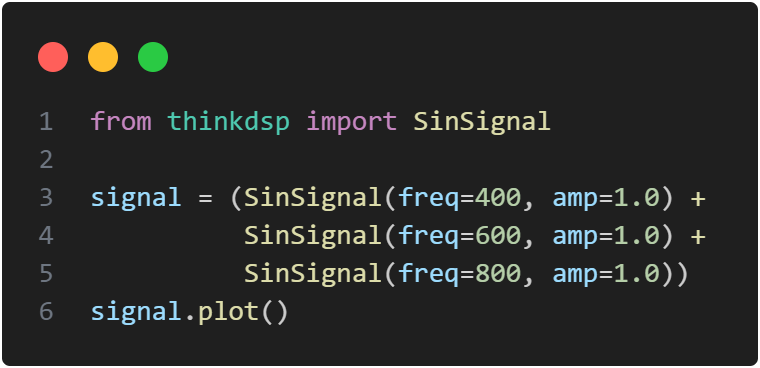
Автоматически созданное описание

Рис. 1.6. Спектр фрагмента после применения band\_stop.

Как мы видим, данная функция убирает частоты из заданного диапазона. Звук фрагмента при удалении частот со 100 Гц до 1000 Гц сильно изменился, в нем практически не слышны ударные.

## Упражнение 1.3.

Создадим сигнал, состоящий из синусов, разной частоты, однако кратных одному числу, например 200:



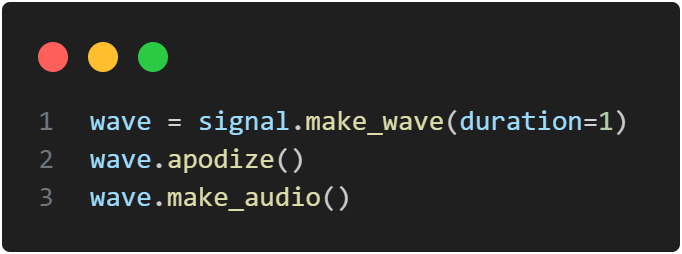
Полученный сигнал имеет следующий вид:

Изображение выглядит как График, линия, диаграмма, текст

Автоматически созданное описание

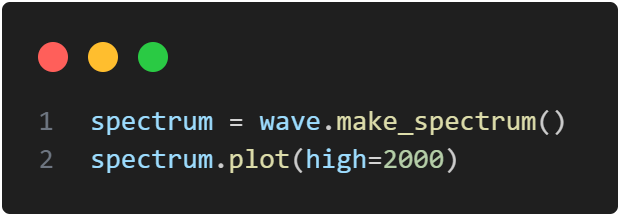
Рис. 1.7. Сигнал, полученный суммой синусов разной частоты.

Создадим файл для прослушивания этого звука, длинной 1 секунда:



Полученный звуковой файл является однотонным писком, похожим на звук гудка, но монотонного.

Выведем спектр полученного сигнала:



Результат выглядит следующим образом:

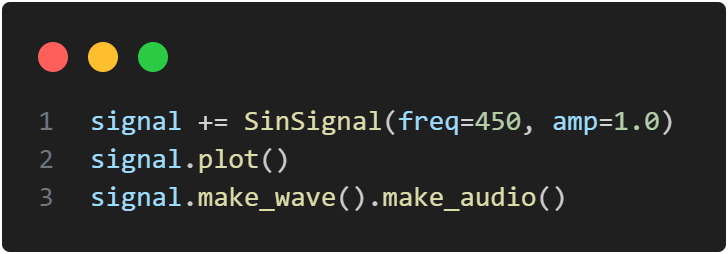
Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание

Рис. 1.8. Спектр сигнала, полученного суммой синусов разной частоты.

Как видим, спектр полностью соответствует ожидания, на нем пики находятся именно в тех частотах, которые мы указывали при создании.

Теперь изменим наш сигнал, добавив частоту, не кратную 200:



Полученный сигнал имеет следующий вид:

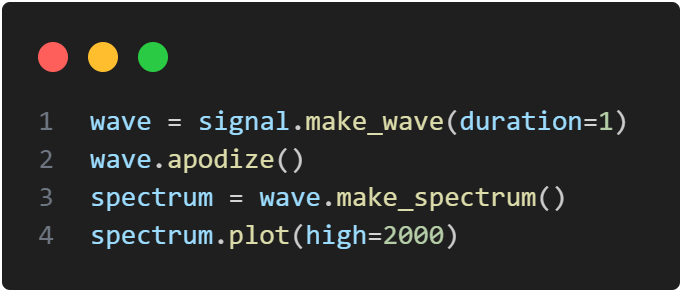
Изображение выглядит как График, диаграмма, линия, Шрифт

Автоматически созданное описание

Рис. 1.9. Сигнал, после добавления синуса не кратной частоты.

Полученный сигнал сильно отличается от того, который был ранее. Так же аудио файл тоже чуть-чуть отличается. В монотонном звуке гудка различим какой-то посторонний периодический сигнал.

Выведем спектр полученного сигнала:



Полученный спектр имеет следующий вид:

Изображение выглядит как текст, снимок экрана, линия, Прямоугольник

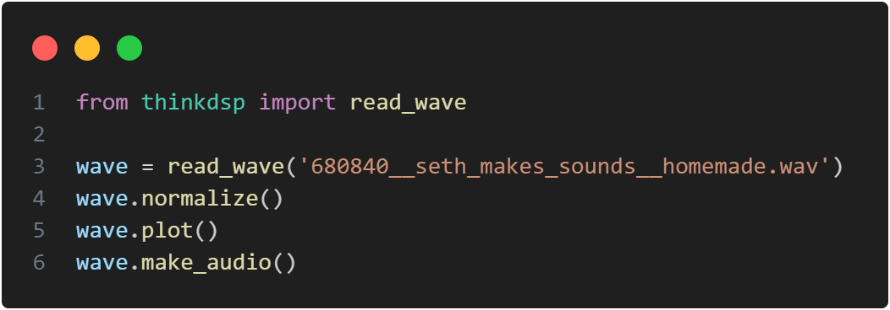
Автоматически созданное описание

Рис. 1.10. Спектр сигнала, после добавления синуса не кратной частоты.

Как и ожидалось, в спектре появился добавленный ранее сигнал.

## Упражнение 1.4.

Напишем функцию для ускорения и замедления аудио. Для начала прочитаем аудио фрагмент и выведем его на экран:



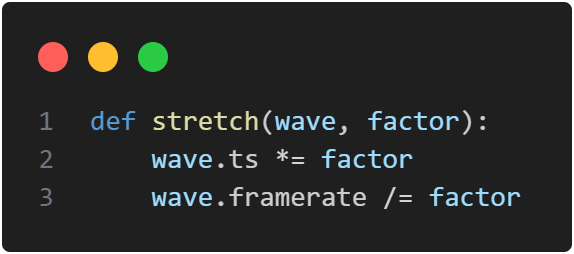
Спектрограмма будет выглядеть следующим образом:

Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

Рис. 1.11. Спектрограмма аудио файла.

Функция для ускорения будет выглядеть следующим образом:



Она изменяет ts (которое используется для корректного отображения временной шкалы в plot) и framerate, что, собственно, и ускоряет произведение.

Передадим функции значение 0.5, что эквивалентно ускорению в 2 раза:



После выполнения мы получили аудио файл, который ускорен в 2 раза, как и ожидалось. Посмотрим на полученную спектрограмму:

Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

Рис. 1.12. Спектрограмма аудио файла после ускорения.

Как мы видим, полученная спектрограмма не отличается от исходной ничем, кроме длительности аудио фрагмента, он меньше в 2 раза.

# Приложение:

Ссылка на репозиторий с исходными кодами: <https://github.com/DafterT/telecom_labs>