

Лабораторная работа №1
Звуки и сигналы

Крынский Павел

26 мая 2021 г.

Оглавление

1	Упражнение 1.1	4
2	Упражнение 1.2	5
2.1	Скачивание звука и работа с ним	5
2.2	Спектр звука	7
2.3	Фильтрация звука	7
3	Упражнение 1.3	9
3.1	Создание сложного сигнала	9
3.2	Добавление новой частоты	11
4	Упражнение 1.4	12
5	Выводы	14

Список иллюстраций

2.1	Исходный звук	6
2.2	Исходный звук	6
2.3	Спектр сегмента звука	7
2.4	Спектр сегмента звука	8
3.1	Спектр сегмента звука	10
3.2	Визуализация сегмента звука	11
4.1	Визуализация ускоренного звука	13

Листинги

2.1	Загрузка и прослушивание звука	5
2.2	Визуализация звука	5
2.3	Изменение и прослушивание звука	6
2.4	Визуализация укороченного звука	6
2.5	Спектр сегмента звука	7
2.6	Фильтрация и воспроизведение звука	7
2.7	Визуализация фильтрации	7
3.1	Создание сложного сигнала из 4 элементов	9
3.2	Воспроизведение сложного сигнала	10
3.3	Визуализация сигнала	10
3.4	Добавление новой частоты и воспроизведение	11
4.1	Загрузка и прослушивание звука	12
4.2	Функция stretch	12
4.3	Прослушивание ускоренного звука	12
4.4	Визуализация ускоренного звука	12

Глава 1

Упражнение 1.1

В данном упражнении нам нужно открыть `chap01.ipynb`, прочитать пояснения и запустить примеры. Поэтому здесь я изучил все примеры с комментариями и позапускал их.

Глава 2

Упражнение 1.2

2.1 Скачивание звука и работа с ним

С предложенного нам сайта скачан звук проезжающей машины. Ссылка на соответствующий звук:

<https://freesound.org/people/14FValtrovaT/sounds/419691/>.

Далее я загрузил звук, прослушал его , и получил его визуализацию.

```
1 wave = read_wave('419691__14fvaltrovat__car-driving.wav')
2 wave.normalize()
3 wave.make_audio()
```

Листинг 2.1: Загрузка и прослушивание звука

```
1 wave.plot()
```

Листинг 2.2: Визуализация звука

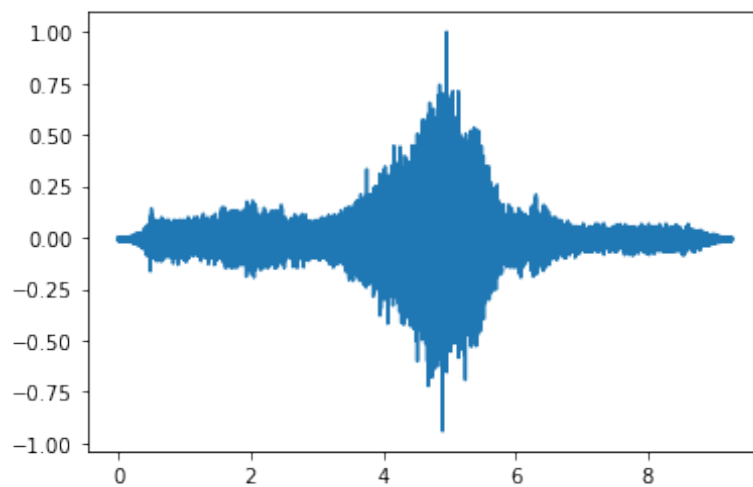


Рис. 2.1: Исходный звук

Берем полусекундный сегмент.

```
1 segment = wave.segment(start=8, duration=8)
2 segment.make_audio()
```

Листинг 2.3: Изменение и прослушивание звука

```
1 segment.plot()
```

Листинг 2.4: Визуализация укороченного звука

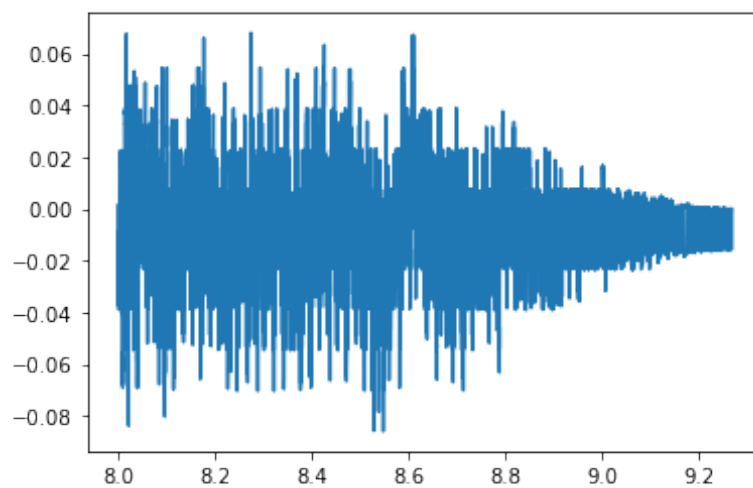


Рис. 2.2: Исходный звук

2.2 Спектр звука

Теперь рассмотрим спектр нашего полусекундного сегмента звука.

```
1 spectr = segment.make_spectrum()  
2 spectr.plot(high=500)
```

Листинг 2.5: Спектр сегмента звука

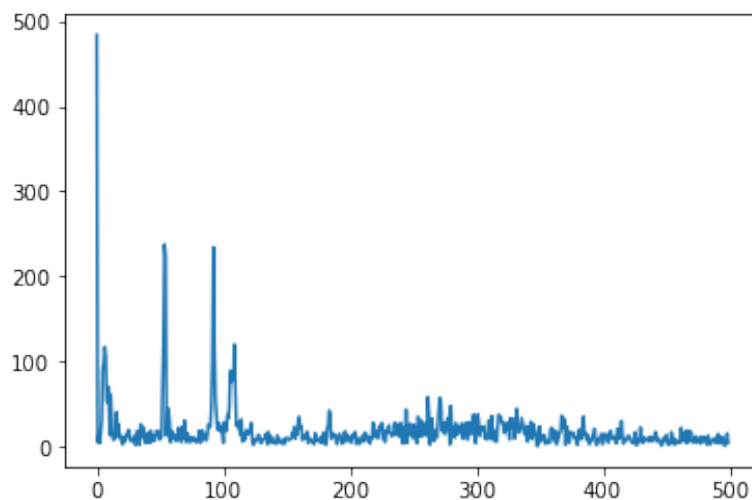


Рис. 2.3: Спектр сегмента звука

2.3 Фильтрация звука

Применим фильтр нижних частот.

```
1 spectrum.low_pass(500)  
2 spectrum.make_wave().make_audio()
```

Листинг 2.6: Фильтрация и воспроизведение звука

```
1 spectrum.make_wave().plot()
```

Листинг 2.7: Визуализация фильтрации

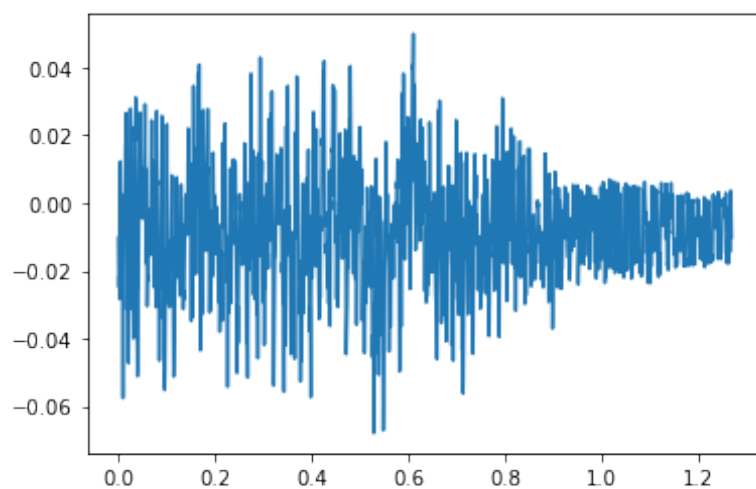


Рис. 2.4: Спектр сегмента звука

Видно, что график изменился, а звук стал, как из туннеля.

Глава 3

Упражнение 1.3

3.1 Создание сложного сигнала

Нужно создать сложный сигнал из объектов `SinSignal` и `CosSignal`.

```
1 from thinkdsp import CosSignal, SinSignal
2
3 signal = (thinkdsp.SinSignal(freq=200 , amp = 1.1)+
4 thinkdsp.SinSignal(freq=300 , amp = 1.5)+
5 thinkdsp.CosSignal(freq=400 , amp = 0.5)+
6 thinkdsp.SinSignal(freq=100 , amp = 1.1))
7 signal.plot()
```

Листинг 3.1: Создание сложного сигнала из 4 элементов

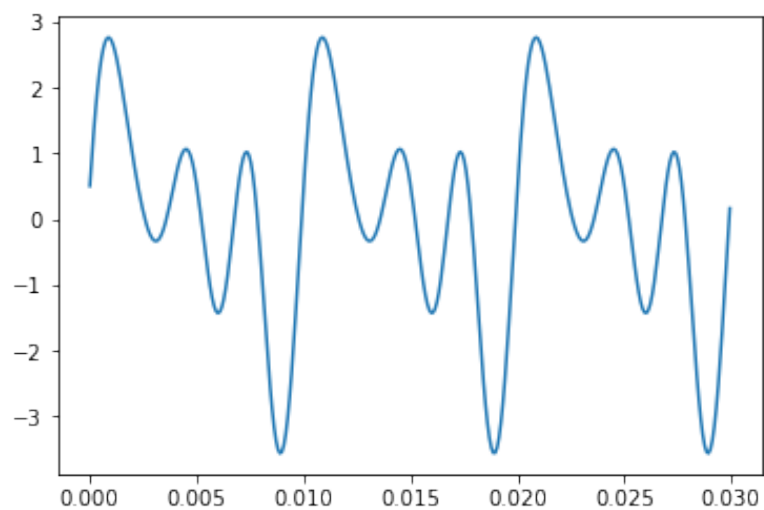


Рис. 3.1: Спектр сегмента звука

Теперь нужно получить звук.

```

1 wave = signal.make_wave(duration=2)
2 wave.apodize()
3 wave.make_audio()

```

Листинг 3.2: Воспроизведение сложного сигнала

Выведем спектр полученного звука.

```

1 spectrum = wave.make_spectrum()
2 spectrum.plot(high=1000)

```

Листинг 3.3: Визуализация сигнала

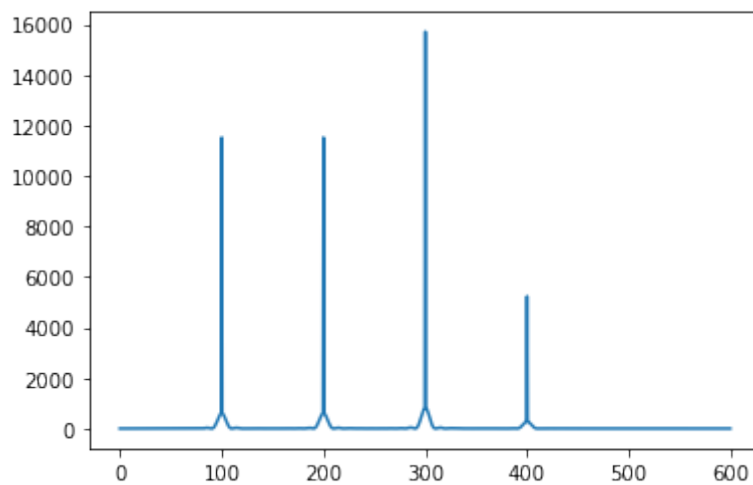


Рис. 3.2: Визуализация сегмента звука

3.2 Добавление новой частоты

Изменим наш сигнал.

```
1 signal = signal + thinkdsp.SinSignal(freq=1100)
2 signal.make_wave().make_audio()
```

Листинг 3.4: Добавление новой частоты и воспроизведение

Теперь слышно добавленную новую частоту, при чём более высокую, потому что `freq=1000`.

Глава 4

Упражнение 1.4

Подготовим звук.

```
1 wave = thinkdsp.read_wave('419691__14fvaltrovat__car-driving.wav')
2 wave.normalize()
3 wave.make_audio()
```

Листинг 4.1: Загрузка и прослушивание звука

Теперь сделаем функцию `stretch`.

```
1 def stretch(wave , factor):
2     wave.ts = wave.ts*factor
3     wave. framerate = wave. framerate/factor
```

Листинг 4.2: Функция `stretch`

Попробуем прослушать полученный звук, введя 0.5.

```
1 stretch(wave , 0.5)
2 wave.make_audio()
```

Листинг 4.3: Прослушивание ускоренного звука

По таймеру в колабе время сократился с 9 до 4 секунд.

```
1 wave.plot()
```

Листинг 4.4: Визуализация ускоренного звука

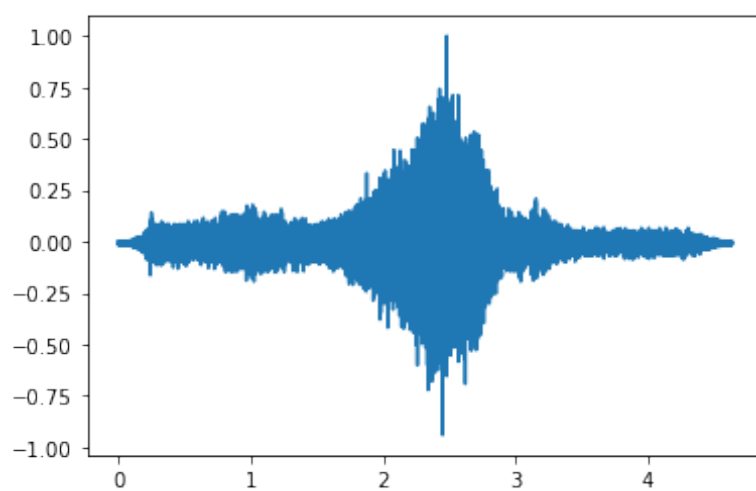


Рис. 4.1: Визуализация ускоренного звука

Глава 5

Выводы

Во время выполнения лабораторной работы получены навыки работы со звуками, волнами и спектрами.