Лабораторная работа №1
Звуки и сигналы

Кобыжев Александр

16 февраля 2021 г.

Оглавление

Список иллюстраций

Листинги

Глава 1

Упражнение 1.1

В данном упражнении нас просят открыть `chap01.ipynb`, прочитать пояснения, а также запустить примеры. Под самый конец блокнота стало жалко скрипку, точнее что с ней сделали виджеты IPython.

Глава 2

Упражнение 1.2

2.1 Скачивание звука и работа с ним

С предложенного нам сайта скачан звук работы какого-то механизма. Ссылка на соответствующий звук:

https://freesound.org/people/felix.blume/sounds/414062/.

Далее звук был загружен, прослушан, и получена его визуализация.

```
wave =
    thinkdsp.read_wave('sounds/414062__felix-blume__machine-gears.wav')
wave.normalize()
wave.make_audio()
```

Листинг 2.1: Загрузка и прослушивание звука

```
wave.plot()
```

Листинг 2.2: Визуализация звука

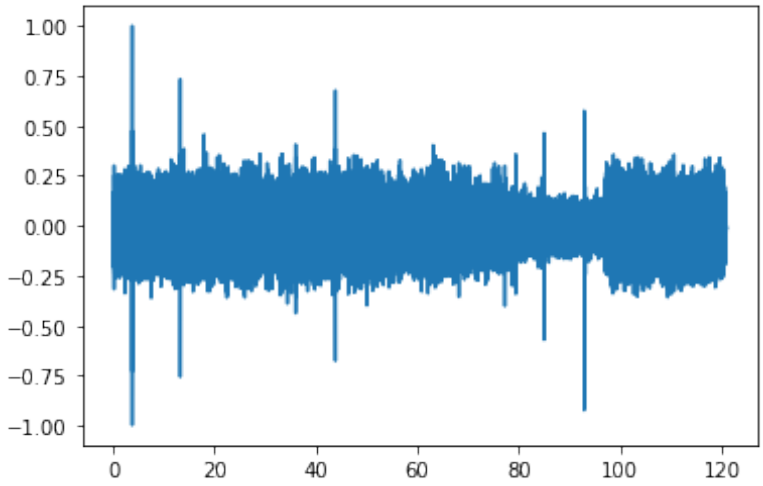

Рис. 2.1: Исходный звук

Далее возьмём полусекундный сегмент звука.

```
segment = wave.segment(start=25, duration=0.5)
segment.make_audio()
```

Листинг 2.3: Изменение и прослушивание звука

```
segment.plot()
```

Листинг 2.4: Визуализация укороченного звука

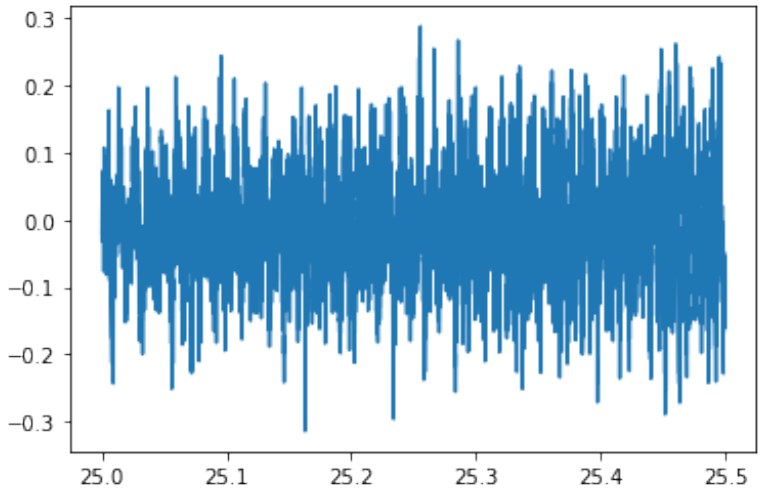

Рис. 2.2: Исходный звук

2.2 Спектр звука

Теперь рассмотрим спектр нашего полусекундного сегмента звука.

```
spectrum = segment.make_spectrum()
spectrum.plot(high=3000)
```

Листинг 2.5: Спектр сегмента звука

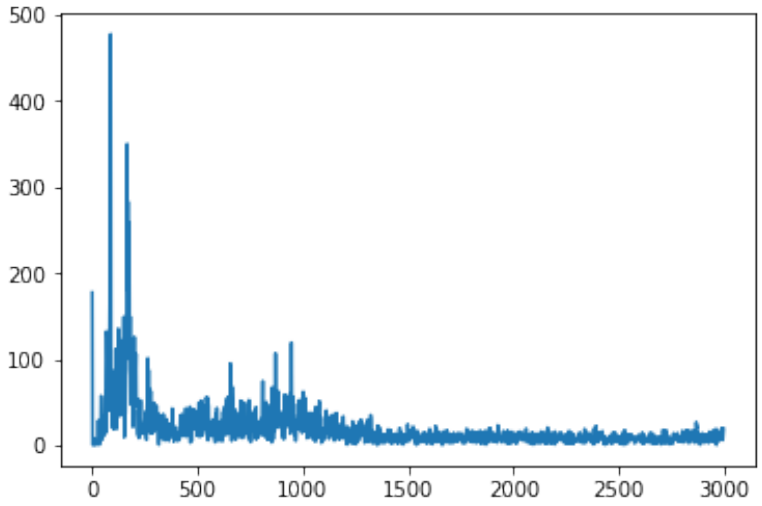

Рис. 2.3: Спектр сегмента звука

Теперь давайте рассмотрим основные и доминирующие частоты.

```
spectrum = segment.make_spectrum()
spectrum.plot(high=1000)
```

Листинг 2.6: Основные и доминирующие частоты

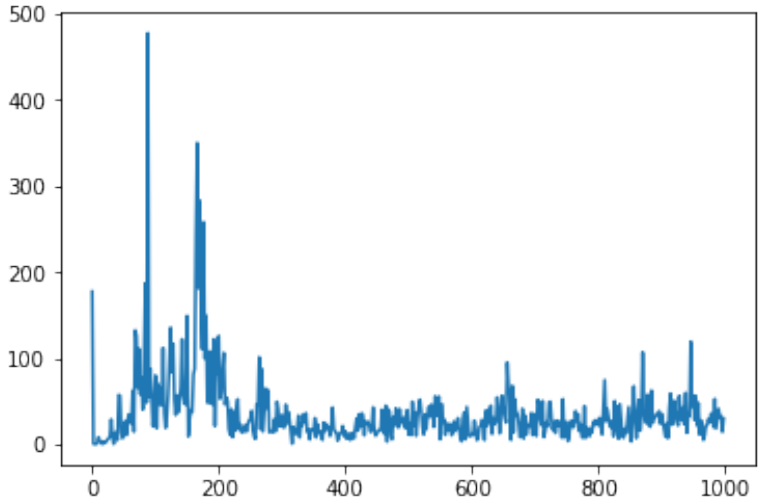

Рис. 2.4: Основные и доминирующие частоты

При помощи метода `spectrum.peaks()[:30]` определим доминирующий пик, который находится находится на частоте 88 Гц.

```
[(477.12651815782436, 88.0),
 (349.78356868530363, 166.0),
 (282.7682770271516, 170.0),
 (257.9959018541909, 176.0),
 (241.34058048533615, 164.0),
 (186.61794990338055, 84.0),
 (181.12766751136527, 168.0),
 (177.34814169151826, 0.0),
 (157.95871927000582, 172.0),
 (149.43185101975794, 180.0),
 (149.14523507428788, 150.0),
 (135.43157096550385, 124.0),
 (132.31096277693203, 68.0),
 (126.03280864487395, 70.0),
 (125.78404373338199, 200.0),
 (122.03771045731844, 192.0),
 (121.9978844128709, 142.0),
 (119.22228552637446, 946.0),
 (116.56985643634354, 128.0),
 (111.97150490905403, 112.0),
 (110.96738906342934, 74.0),
 (110.30567932667302, 174.0),
 (107.97413765938968, 186.0),
```

```
(106.82983478456184, 870.0),
(106.16611169214379, 208.0),
(101.22634883171585, 264.0),
(100.31004880713746, 82.0),
(99.62009343933141, 178.0),
(99.24076695835758, 122.0),
(95.08575904720958, 656.0)]
```

Листинг 2.7: Отфильтрованный массив пиков

2.3 Фильтрация звука

Применим фильтр нижних частот.

```
spectrum.low_pass(2000)
spectrum.make_wave().make_audio()
```

Листинг 2.8: Фильтрация и воспроизведение звука

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

Листинг 2.9: Визуализация фильтрации

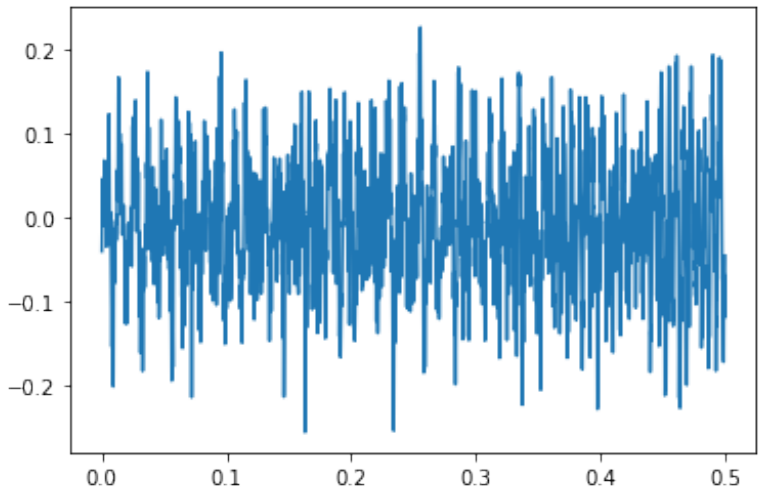

Рис. 2.5: Спектр сегмента звука

Как видно из графика, сигнал после фильтрации значительно поредел, а звук стал похож на воспроизведение его под водой или за стеной.

Глава 3

Упражнение 1.3

3.1 Создание сложного сигнала

Создадим сложный сигнал из объектов `SinSignal` и `CosSignal`.

```
signal = (thinkdsp.SinSignal(freq=140, amp=0.7) +
          thinkdsp.SinSignal(freq=440, amp=0.2) +
          thinkdsp.CosSignal(freq=640, amp=0.8))
signal.plot()
```

Листинг 3.1: Создание сложного сигнала

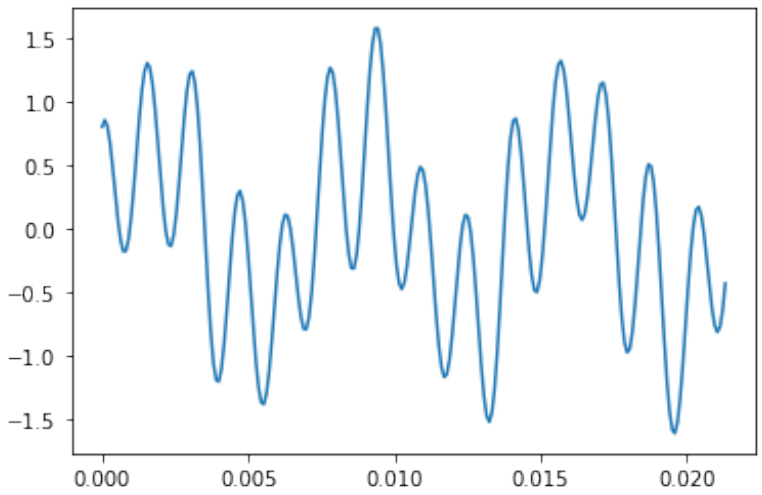

Рис. 3.1: Спектр сегмента звука

Достаточно интересный звук у нас получился, теперь попробуем его прослушать.

```
wave2 = signal.make_wave(duration=1)
wave2.apodize()
wave2.make_audio()
```

Листинг 3.2: Воспроизведение сложного сигнала

Полученный звук схож с чем-то инопланетным, будто я попал в какую-то вселенную Звёздных Войн. Выведем спектр полученного звука.

```
spectrum = wave2.make_spectrum()
spectrum.plot(high=2000)
```

Листинг 3.3: Визуализация сложного сигнала

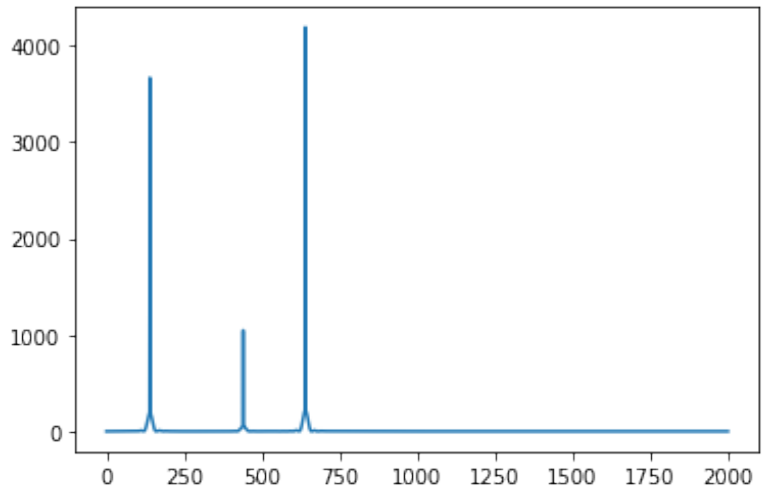

Рис. 3.2: Визуализация сегмента звука

3.2 Добавление новой частоты

Попробуем добавить новую частоту в наш имеющийся сигнал и прослушать полученный звук.

```
signal += thinkdsp.SinSignal(freq=1000)
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`. Теперь звук более похож на набор цифр при звонке через стационарный телефон.

Глава 4

Упражнение 1.4

Для выполнения данного упражнения возьмём скачанный звук из `Упражнения 1.2`.

```
wave3 =
    thinkdsp.read_wave('sounds/414062__felix-blume__machine-gears.wav')
wave3.normalize()
wave3.make_audio()
```

Листинг 4.1: Загрузка и прослушивание звука

Теперь сделаем функцию `stretch`.

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

Листинг 4.2: Функция stretch

Попробуем прослушать полученный звук, введя 0.25. По логике он должен ускориться в 4 раза.

```
stretch(wave3, 0.25)
wave3.make_audio()
```

Листинг 4.3: Прослушивание ускоренного звука

Теперь этот звук напоминает работающий блендер или мясорубку, что аж есть захотелось. Проверим работоспособность нашей функции визуализируя полученный звук.

```
wave3.plot()
```

Листинг 4.4: Визуализация ускоренного звука

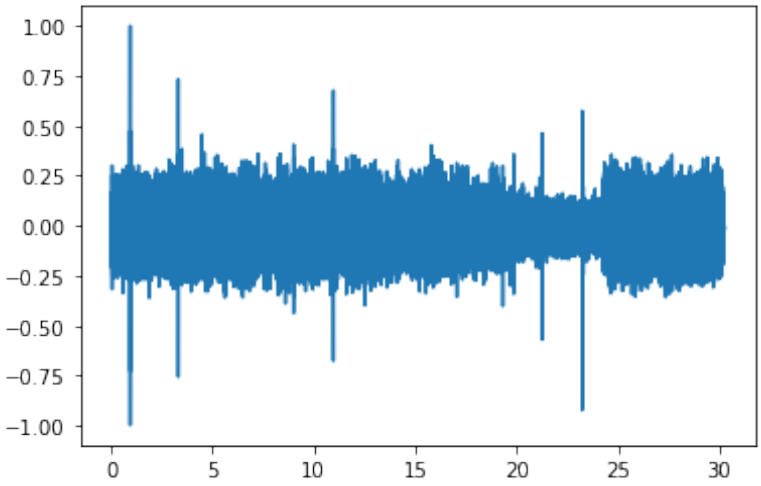

Рис. 4.1: Визуализация ускоренного звука

Глава 5

Выводы

Во время выполнения лабораторной работы получены навыки работаты со звуками, волнами и спектрами. Также я научился находить более высокие и фундаментальные пики, определять частоту, а также ускорять и замедлять звуки и строить графики.