

Санкт-Петербургский государственный политехнический
университет Петра Великого.

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

ЛАБОРАТОРНАЯ РАБОТА

Сигналы и Звуки.

Работу выполнила студентка:

_____ А. И. Луцкевич
« ____ » _____ 2021 г.

Преподаватель лабораторных
работ:

_____ Н. В. Богач
« ____ » _____ 2021 г.

Санкт-Петербург, 2021 г.

Суть работы 1.1:

Прочитать пояснения и запустить примеры.

Был прочитан весь параграф, были запущены все примеры и просмотрены все результаты.

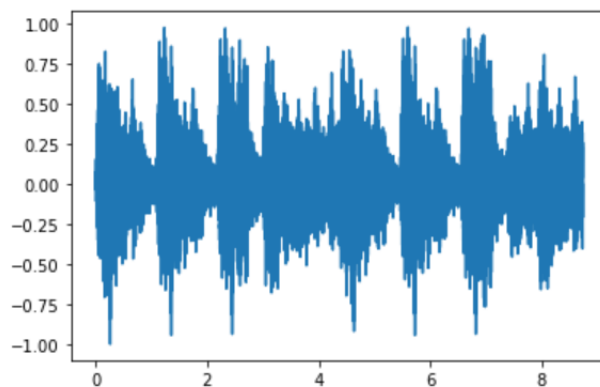
Суть работы 1.2:

Скачать с сайта образец звука, включающего музыку. Выделите примерно полусекундный сегмент, в котором высота постоянна. Вычислите, распечатайте спектр. Как связаны тембр звука и гармоническая структура, видимая в спектре? Используйте переменные для фильтрации тех или иных гармоник, а затем преобразуйте спектры обратно в сигнал и прослушайте его. Как звук соотносится с изменениями, сделанными в спектре?

Перед началом работы были подключены библиотеки, которые я смогу использовать для создания и обработки сигналов в данной лабораторной работе.

Далее мы читаем скачанную нами ранее мелодию. Посмотрим как выглядит `wave.plot()`:

```
In [2]: wave.plot()
```



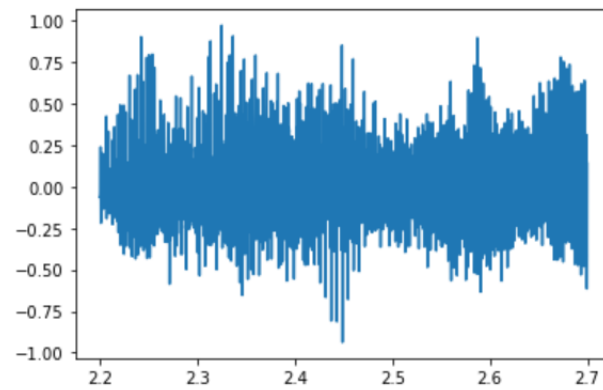
Выберем сегмент (начало - 2.2, длительность - 0.5 с) и построим его:

```
In [5]: segment = wave.segment(start=2.2, duration=0.5)
segment.make_audio()
```

Out[5]:

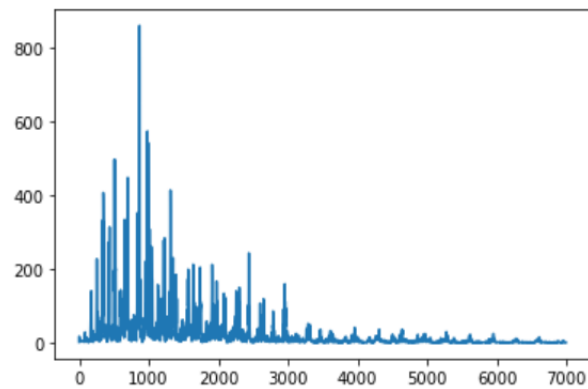


```
In [6]: segment.plot()
```



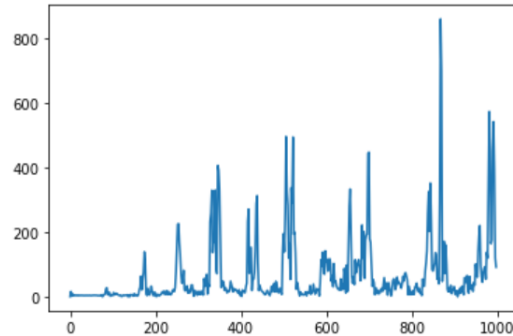
Посмотрим как выглядит спектр:

```
In [10]: spectrum = segment.make_spectrum()
spectrum.plot(high=7000)
```



Отфильтруем гармоники и получим спектр, изображенный ниже:

```
In [11]: spectrum = segment.make_spectrum()  
spectrum.plot(high=1000)
```

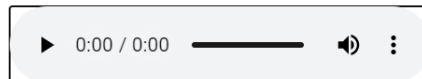


По заданию необходимо преобразовать спектр обратно в сигнал и прослушать то, что получилось:

```
In [13]: spectrum.low_pass(2000)
```

```
In [14]: spectrum.make_wave().make_audio()
```

Out[14]:



Вывод: в конце преобразований звук получился похожим, но чуть более тихим и приглушенным.

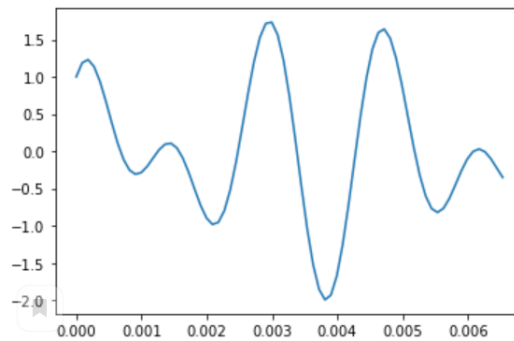
Суть работы 1.3:

Создать сложный сигнал из объектов SinSignal и CosSignal суммируя их. Следует обработать сигнал для получения волны и прослушать его. А также вычислить Spectrum и распечатать его.

Я взяла синусоидальный сигнал частотой 455 Hz и косинусоидальный сигнал частотой 655 Hz. Далее я просуммировала два сигнала и получила суммирующий сигнал. Построим его:

```
In [19]: from thinkdsp import SinSignal
from thinkdsp import CosSignal

signal = (SinSignal(freq=455, amp=1.0) +
          CosSignal(freq=655, amp=1.0))
signal.plot()
```

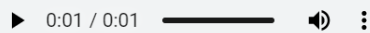


После данных действий звук был прослушен и построен спектр:

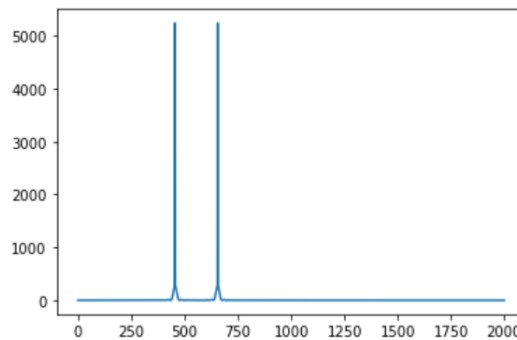
```
In [20]: wave2 = signal.make_wave(duration=1)
wave2.apodize()
```

```
In [21]: wave2.make_audio()
```

Out[21]:



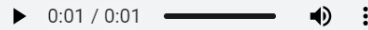
```
In [22]: spectrum = wave2.make_spectrum()
spectrum.plot(high=2000)
```



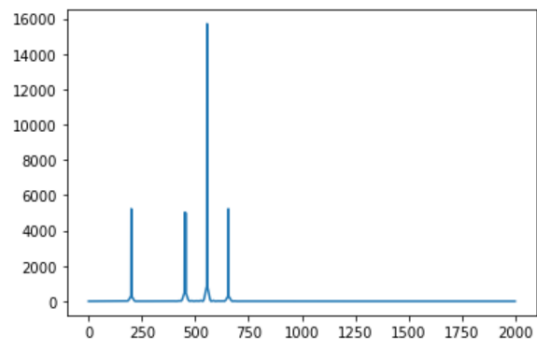
Я добавила к полученному сигналу синусоидальный сигнал с частотой 555 Hz. Прослушаем звук и построим спектр:

```
In [27]: signal += SinSignal(freq=555)
         signal.make_wave().make_audio()
```

Out[27]:



```
In [29]: wave3 = signal.make_wave(duration=1)
         wave3.apodize()
         spectrum2 = wave3.make_spectrum()
         spectrum2.plot(high=2000)
```



В результате прослушивания этого последнего сигнала я услышала, что сигнал стал звучать ниже и появились звуковые колебания.

Суть работы 1.4:

Следует написать функцию `stretch`, берущую `wave` и коэффициент изменения. Данная функция должна ускорять или замедлять сигнал изменением `ts` и `framerate`.

Сначала была создана волна, чтобы в дальнейшем произвести проверку данной функции.

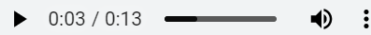
Далее была написана функция `stretch`, которая ускоряет или замедляет сигнал изменением коэффициентов `ts` и `framerate`. Сама функция представлена ниже:

```
In [32]: def stretch(wave, factor):
         wave.ts *= factor
         wave.framerate /= factor
```

Проверим работоспособность данной функции: передадим в качестве аргумента волну и коэффициент 1.5. Предполагаем, что темп замедлится в 1.5 раза.

```
In [33]: stretch(wave, 1.5)
         wave.make_audio()
```

Out[33]:



Действительно, мелодия стала медленней, а ее длительность увеличилась в полтора раза.

Заключение:

Вывод данной лабораторной работы состоит в том, что я научилась работать с сигналами, обрабатывать, фильтровать, воспроизводить их с помощью специальной библиотеки языка Python. Были получены начальные знания о всех необходимых далее понятиях.