

Санкт-Петербургский государственный политехнический
университет Петра Великого.

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

ЛАБОРАТОРНАЯ РАБОТА

Гармоники.

Работу выполнила студентка:

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

Преподаватель лабораторных
работ:

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

Санкт-Петербург, 2021 г.

Суть работы 2.1:

Прочитать пояснения и запустить примеры.

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

Суть работы 2.2:

Пилообразный сигнал линейно нарастает от -1 до 1, а затем резко падает до -1 и повторяется. Напишите класс, называемый `SawtoothSignal`, расширяющий `signal` и предоставляющий `evaluate` для оценки пилообразного сигнала. Вычислите спектр пилообразного сигнала. Как соотносится его гармоническая структура с треугольным и прямоугольным сигналами?

Создадим класс `SawtoothSignal`. Он будет наследником класса `Sinusoid`, изменен только метод `evaluate`:

```
In [1]: from thinkdsp import Sinusoid
        from thinkdsp import normalize, unbias
        import numpy as np

        class SawtoothSignal(Sinusoid):
            def evaluate(self, ts):
                cycles = self.freq * ts + self.offset / np.pi / 2
                frac, _ = np.modf(cycles)
                ys = normalize(unbias(frac), self.amp)
                return ys
```

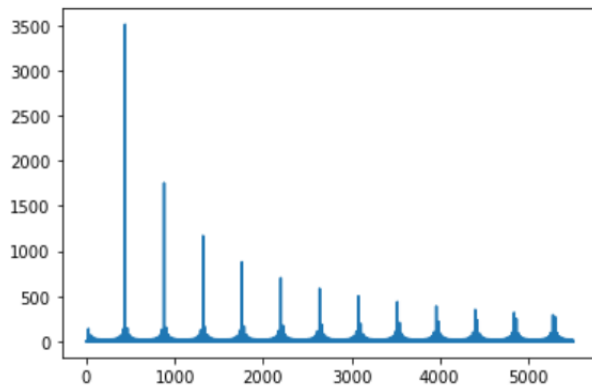
Создадим объект этого класса (длина - 1 с, частота - 30000) и прослушаем полученный звук:

```
In [2]: sawtooth = SawtoothSignal()
        sawtooth = sawtooth.make_wave(1, 30000)
        sawtooth.make_audio()
```

Out[2]:

▶ 0:00 / 0:01 — 🔊 ⋮

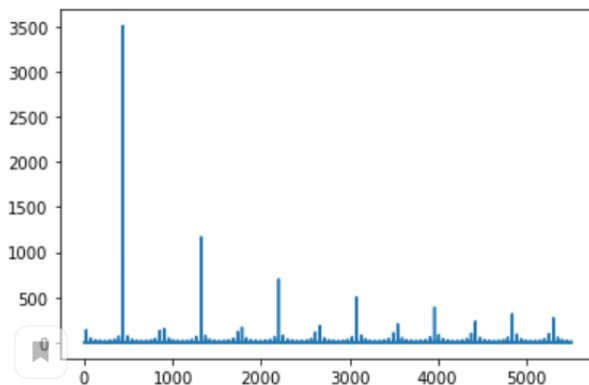
Также необходимо посмотреть как выглядит спектр:



Теперь необходимо сравнить его гармоническую структуру с прямоугольным и треугольным сигналом. Построим спектр прямоугольного сигнала:

```
In [4]: from thinkdsp import SquareSignal

square = SquareSignal(amp=0.5).make_wave(1, 30000)
square.make_spectrum().plot()
```

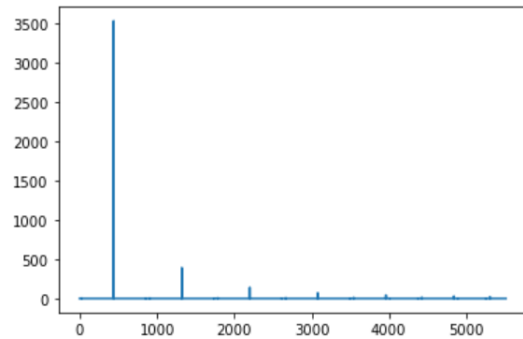


Падают они обе одинаково, но главное отличие в том, что у прямоугольного сигнала только нечетные гармоники.

Треугольная же волна падает быстрее пилообразной и включает также только нечетные гармоники.

```
In [5]: from thinkdsp import TriangleSignal

triangle = TriangleSignal(amp=0.79).make_wave(1, 30000)
triangle.make_spectrum().plot()
```



Суть работы 2.3:

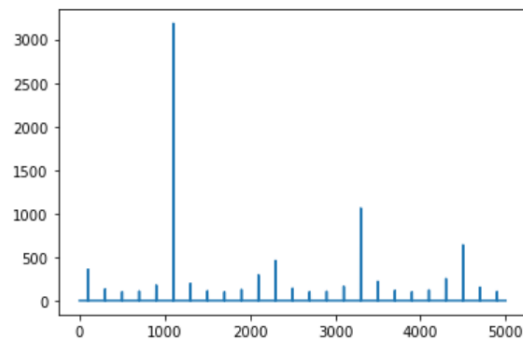
Создайте прямоугольный сигнал 1100 Гц и вычислите wave с выборками 10000 кадров в секунду. Постройте спектр и убедитесь, что большинство гармоник "завернуты" из-за биений. Слышны ли последствия этого при проигрывании?

Создадим прямоугольный сигнал такой, который необходим по заданию. Построим спектр и прослушаем его:

```
In [35]: from thinkdsp import SquareSignal

square = SquareSignal(1100).make_wave(duration = 0.5, framerate = 10000)
square.make_audio()

s = square.make_spectrum()
s.plot()
```



Видна основная гармоника (1100), а также 3300. Из-за биений гармоники

некоторые завернуты, поэтому можно наблюдать картину, что представлено выше.

Суть работы 2.4:

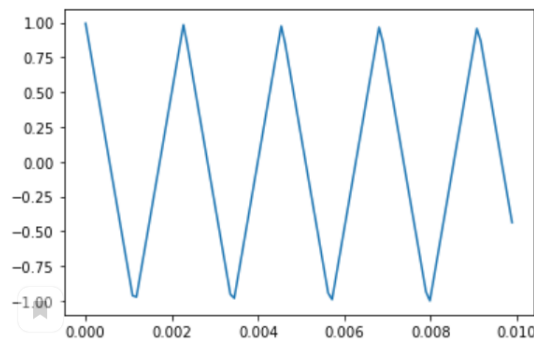
Возьмите объект Spectrum и распечатайте несколько первых значений spectrum.fs.

Убедитесь, что они начинаются с нуля, то есть Spectrum.hs[0] - амплитуда компоненты с частотой 0. Но что это значит? Проведите такой эксперимент:

1. Создайте треугольный сигнал с частотой 440 Гц и wave длительностью 0,01 секунд. Распечатайте сигнал.
2. Создайте объект Spectrum и распечатайте Spectrum.hs[0]. Каковы амплитуда и фаза этого компонента?
3. Установите Spectrum.hs[0] = 100. Как эта операция повлияет на сигнал?

Первым шагом создадим треугольный сигнал с частотой 440 Гц и wave длительностью 0,01 секунд:

```
In [15]: triangle = TriangleSignal().make_wave(0.01)
triangle.plot()
```



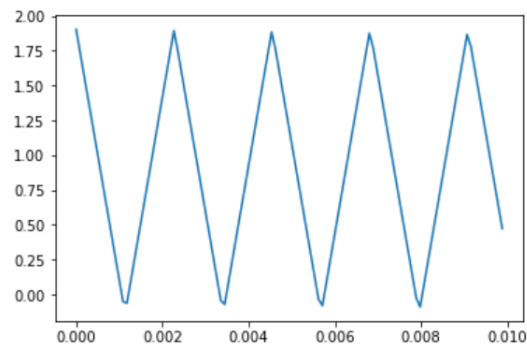
Убедимся, что первое число спектра - это 0 (комплексное число, близкое к нулю).

```
In [16]: s = triangle.make_spectrum()
s.hs[0]
```

```
Out[16]: (1.0436096431476471e-14+0j)
```

Последним шагом установим Spectrum.hs[0] = 100. По картинке можно подумать, что ничего не поменялось, но если посмотреть на вертикальную шкалу, то можно заметить, что произошло вертикальное смещение волны вверх.

```
In [17]: s.hs[0] = 100  
s.make_wave().plot()
```



Суть работы 2.5:

Напишите функцию, принимающую Spectrum как параметр и изменяющую его делением каждого элемента hs на соответствующую частоту из fs. Проверьте эту функцию, используя прямоугольный, треугольный или пилообразный сигналы:

1. Вычислите Spectrum и распечатайте его
2. Измените Spectrum, вновь используя свою функцию, и распечатайте его
3. Используйте Spectrum.makewave, чтобы сделать wave из измененного Spectrum, и прослушайте его. Как эта операция повлияла на сигнал?

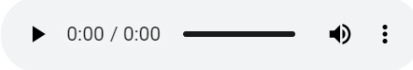
Сначала была создана функция, которая принимает Spectrum как параметр и делит каждый элемент hs на соответствующую fs:

```
In [25]: def del_spectrum(spectrum):  
         spectrum.hs[1:] /= spectrum.fs[1:]  
         spectrum.hs[0] = 0
```

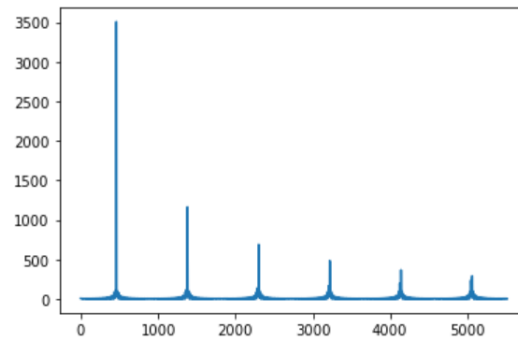
Мною был создан прямоугольный сигнал, прослушана аудиозапись и распечатан спектр:

```
In [27]: wave = SquareSignal(460).make_wave(0.5)
         wave.make_audio()
```

Out[27]:

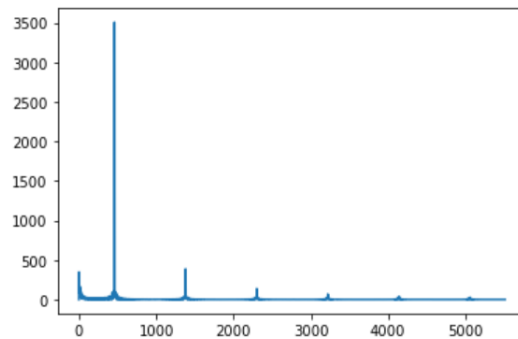


```
In [28]: spectrum = wave.make_spectrum()
         spectrum.plot()
```




Изменим Spectrum с помощью нашей функции, распечатаем спектр. Сделаем снова wave и прослушаем ее:

```
In [23]: del_spectrum(spectrum)
         spectrum.scale(460)
         spectrum.plot()
```



```
In [24]: new = spectrum.make_wave()
         new.make_audio()
```

Out[24]:



Наша функция глушит гармоники и из-за этого работает как фильтр низких частот. Звук стал звучать тише.

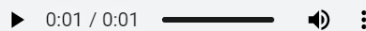
Суть работы 2.6:

У треугольных и прямоугольных сигналов есть только нечетные гармоники; в пилообразном сигнале есть и четные, и нечетные гармоники. Гармоники прямоугольных и пилообразных сигналов уменьшаются пропорционально $1/f$; гармоники треугольных сигналов - пропорционально $1/f^2$. Можно ли найти сигнал, состоящий из четных и нечетных гармоник, спадающих пропорционально $1/f^2$?

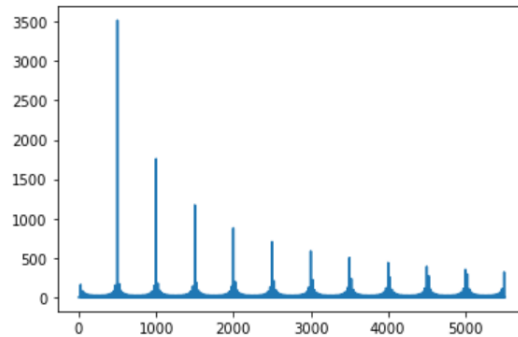
Сначала создадим пилообразный сигнал с частотой 500 Гц и длиной 1 секунда:

```
In [30]: signal = SawtoothSignal(500)
         wave = signal.make_wave(1, 20000)
         wave.make_audio()
```

Out[30]:



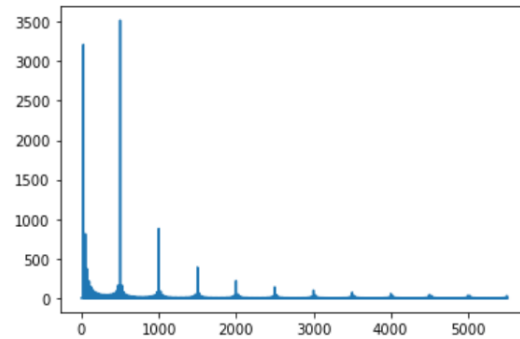
```
In [32]: spectrum = wave.make_spectrum()
         spectrum.plot()
```



Помним, что гармоники пилообразных сигналов уменьшаются $1/f$ (это можно посмотреть в пункте 2.2). Задача состоит в том, чтобы гармоники спадали пропорционально $1/f^2$.

Решение состоит в том, чтобы просто применить функцию, которая была написана в пункте 2.5, которая как раз и является делителем частоты. Проверим теорию, передав в качестве параметра нашей функции `spectrum`:


```
In [33]: del_spectrum(spectrum)
          spectrum.scale(500)
          spectrum.plot()
```



Все работает верно, гармоники теперь спадает быстрее, пропорционально $1/f^2$, как и было предложено по заданию.

Заключение:

Вывод данной лабораторной работы состоит в том, что были изучены новые сигналы разных форм: треугольные, прямоугольные и пилообразные. С помощью практических заданий мною была услышана разница между этими сигналами и изучена разница с точки зрения гармоник. Также были изучены биения, но с практической точки зрения посмотреть не получилось по тому заданию, которое было дано.