

Санкт-Петербургский Государственный Политехнический Университет
Институт Компьютерных Наук и Технологий

**Высшая школа интеллектуальных систем и суперкомпьютерных
технологий**

Отчёт по лабораторной работе №2
на тему
Гармоники

Работу выполнил
Студент группы 3530901/80203
Курняков П. М.
Преподаватель
Богач Н.В.

Санкт-Петербург, 2021 год

1 Настройка проекта

Перед тем как выполнять задания необходимо настроить проект и сделать все необходимые импорты:

```
from __future__ import print_function, division

%matplotlib inline

import thinkdsp
import thinkplot
import numpy as np
import math

import warnings
warnings.filterwarnings('ignore')

from IPython.html.widgets import interact, interact_manual, fixed
from IPython.html import widgets
from IPython.display import display

PI2 = 2 * math.pi
```

Рис. 1: 2

2 Упражнение номер №1

1) Необходимо создать класс SawtoothSignal, расширяющий signal и предоставляющий evaluate для оценки пилообразного сигнала. 2) Необходимо вычислить спектр пилообразного сигнала и посмотреть как соотносится его гармоническая структура с треугольным и прямоугольным сигналами.

Пункт 1.1: Реализуем класс SawtoothSignal. Он расширяет Signal и предоставляет возможность сделать оценку пилообразного сигнала:

```
: class SawtoothSignal(thinkdsp.Sinusoid):
    def evaluate(self, ts):
        cycles = self.freq * ts + self.offset / PI2
        frac, _ = np.modf(cycles)
        ys = thinkdsp.normalize(thinkdsp.unbias(frac), self.amp)
        return ys
```

Рис. 2: 2

Пункт 1.2: Используя созданный класс создадим пилообразный сигнал:

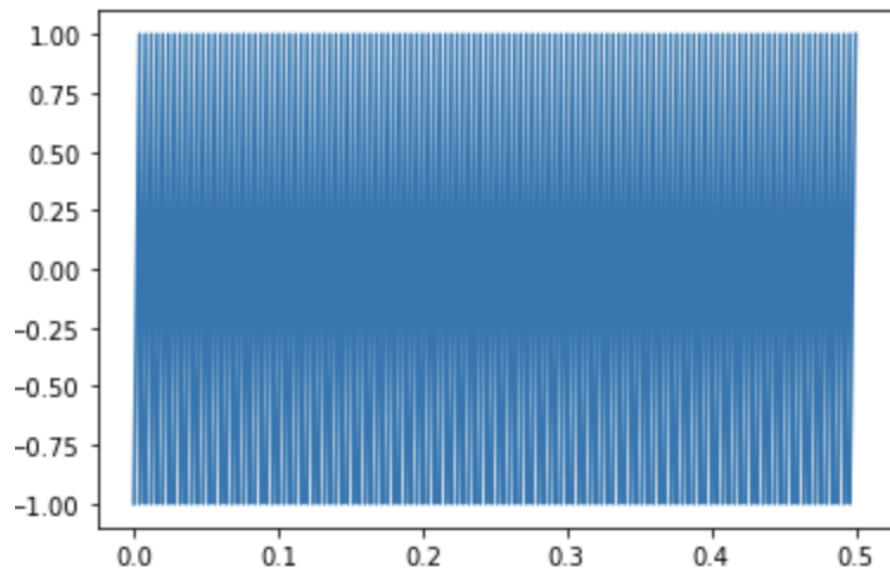


Рис. 3: 2

Распечатаем спектр:

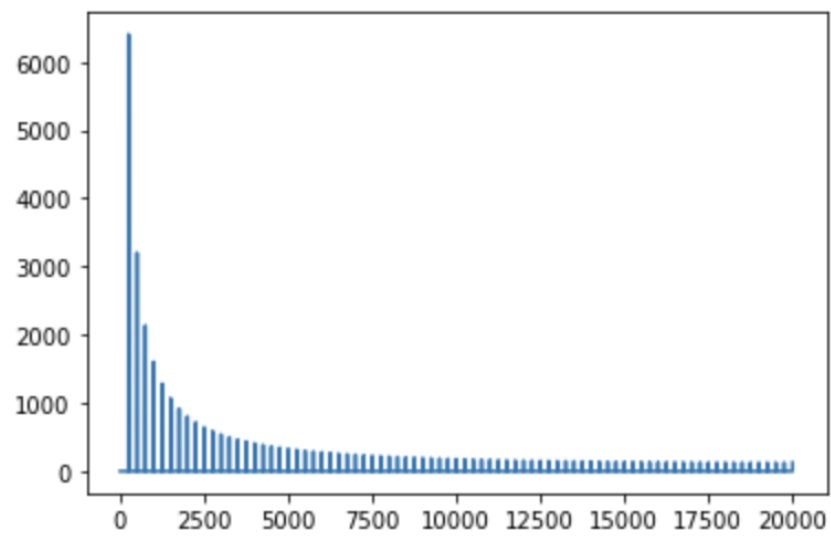


Рис. 4: 2

Сделаем наложение пилообразного сигнала и прямоугольного:

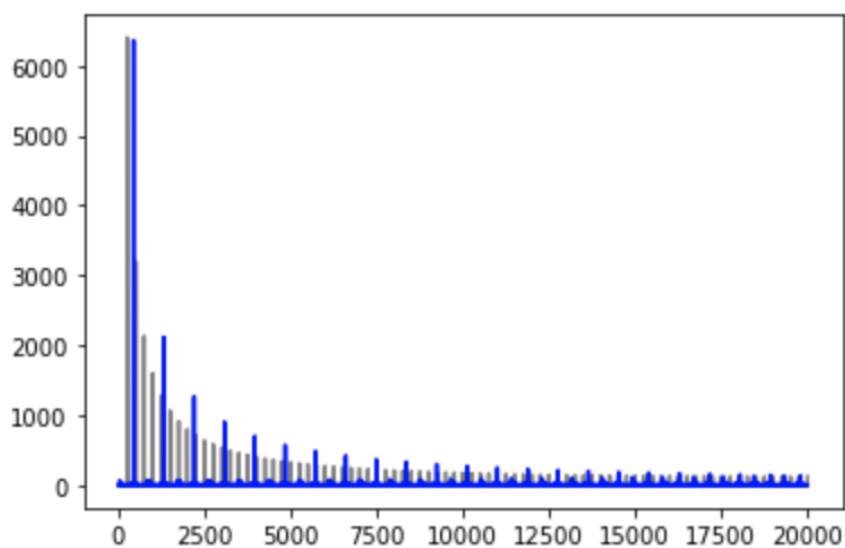


Рис. 5: 2

Из графика видно, что спад пилообразных происходит аналогично спаду прямоугольных. Так же стоит отметить, что пилообразный сигнал включает как четные, так и нечетные гармоники. Сделаем наложение пилообразного сигнала и треугольного:

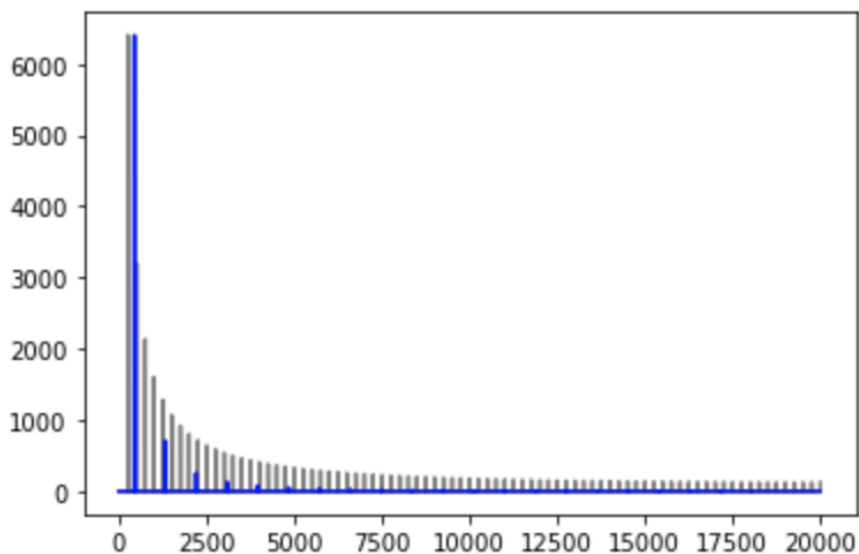


Рис. 6: 2

В отличие от пилообразных гармоник, у треугольных спад протекает значительно быстрее.

3 Упражнение номер №2

Необходимо создать прямоугольный сигнал 1100Гц и вычислить wave с выборками 10 000 кадров в секунду. Построить спектр и убедиться, что большинство гармоник "завернуты" из-за биений. Проверить слышны ли последствия этого при проигрывании.

Создадим прямоугольный сигнал 1100Гц:

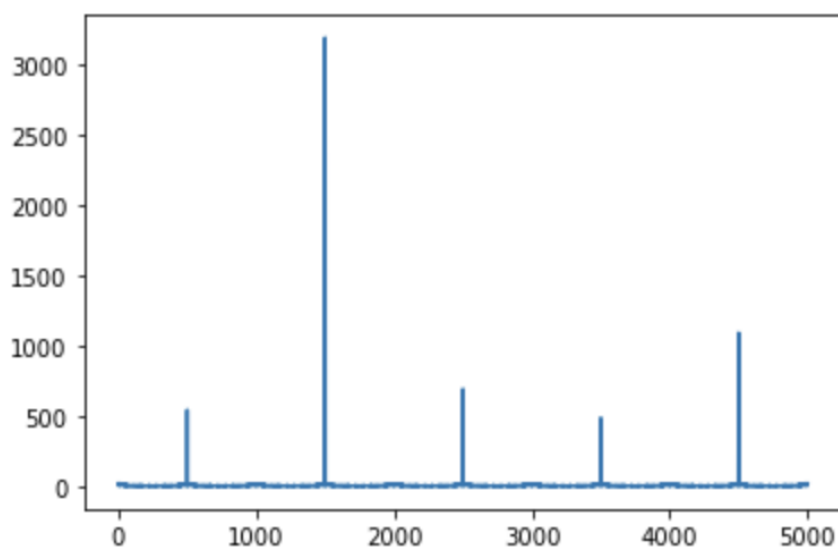


Рис. 7: 2

Как видно из графика, первая гармоника находится в нужном месте. А вот вторая гармоника (5500Hz) совмещена с гармоникой 4500Hz. Следующая гармоника совмещена с гармоникой на 2300Hz.

4 Упражнение номер №3

Необходимо взять объект `spectrum` и распечатать несколько первых значений `spectrum.fs`. Убедиться что они начинаются с нуля. Затем нужно провести эксперименты: 1) Создать треугольный сигнал с частотой 440Гц и `wave` длительностью 0.01 секунд. Распечатать сигнал. 2) Создать объект `spectrum` и распечатать `spectrum.hs[0]`. Посмотреть каковы амплитуда и фазы этого компонента. 3) Установить `spectrum.hs[0] = 100`. Проверить как эта операция влияет на сигнал.

Создали треугольный сигнал и распечатали `wave`:

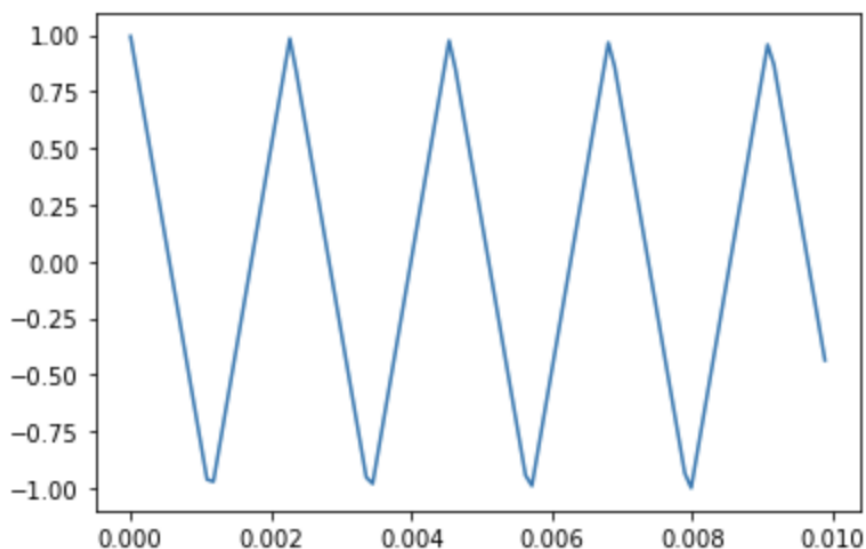


Рис. 8: 2

Каждый элемент массива `hs` объекта `Spectrum` представляет собой комплексное число и соответствует частотной компоненте: размах пропорционален амплитуде соответствующей компоненты, а

угол - это фаза. Как видно из результатов выполнения кода, первый элемент массива `hs` - комплексное число, близкое к нулю, мнимая часть равна нулю. Присвоим первый элемент 100 и посмотрим, что из этого выйдет.

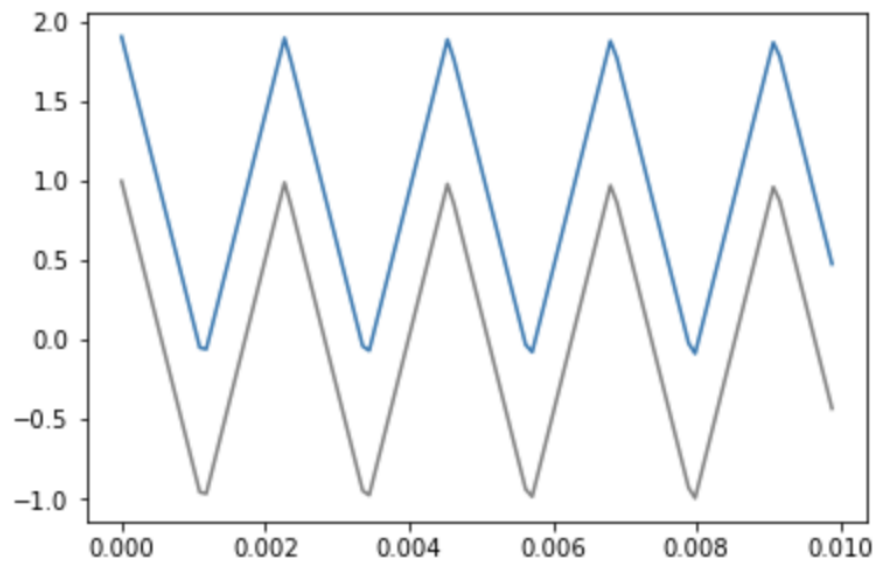


Рис. 9: 2

Как мы можем заметить это привело к вертикальному смещению волны

5 Упражнение номер №4

Необходимо реализовать функцию, которая в качестве аргумента принимает `spectrum` и изменяет его делением каждого элемента `hs` на соответствующую частоту из `fs`. Проверить эту функцию на прямоугольном, треугольном и пилообразном сигналах: 1) Вычислить `spectrum` и распечатать его 2) Изменить `spectrum` и распечатать его 3) Использовать `spectrum.make_wave`, чтобы сделать `wave` из измененного `spectrum`, прослушать его. Посмотреть как изменился сигнал.

Реализуем метод:

```
: def filter_spectrum(spectrum):
    spectrum.hs[1:] /= spectrum.fs[1:]
    spectrum.hs[0] = 0
```

Рис. 10: 2

Создадим треугольный сигнал:

```
wave = thinkdsp.TriangleSignal(freq=440).make_wave(duration=0.5)
wave.make_audio()
```

Рис. 11: 2

Применим написанную нами функцию к созданному сигналу:

```
spectrum = wave.make_spectrum()
spectrum.plot(high=10000, color='gray')
filter_spectrum(spectrum)
spectrum.scale(440)
spectrum.plot(high=10000)
```

Рис. 12: 2

Распечатаем полученный спектр:

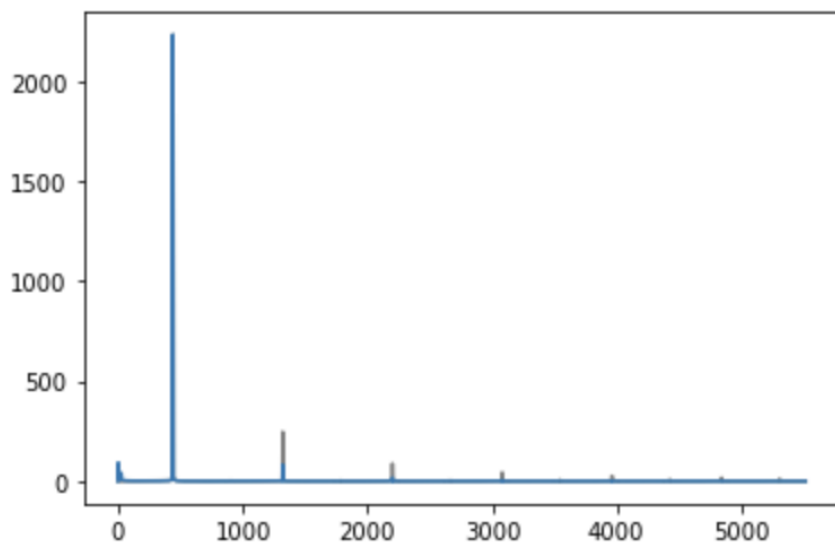


Рис. 13: 2

Фильтр подавляет гармоники, поэтому он действует как фильтр нижних частот.

6 Упражнение номер №5

Проверить можно ли найти сигнал, состоящий из четных и нечетных гармоник, спадающих пропорционально $1/(f^2)$.

Нужно создать сигнал, состоящий из четных и нечетных гармоник, при этом, что эти гармоники падали пропорционально $1/(f^2)$. Для этого воспользуемся одним из способов: создадим пилообразный сигнал и выведем его спектр:

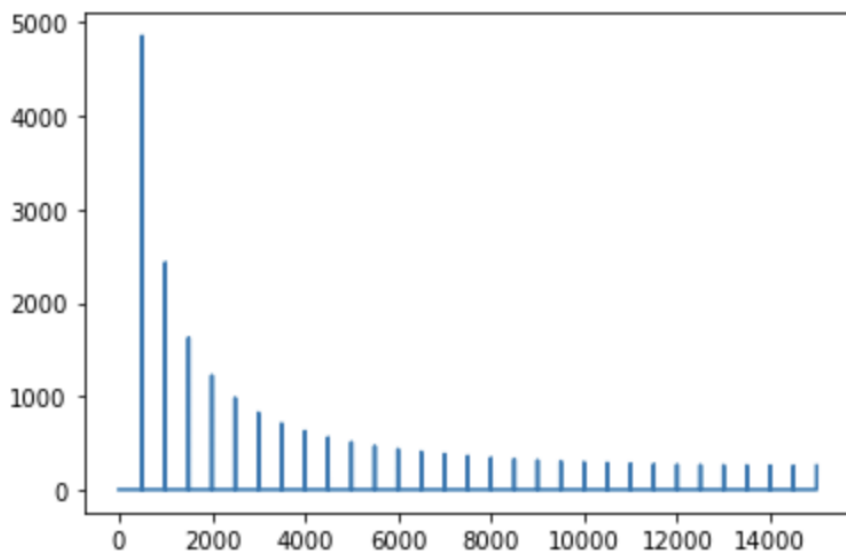


Рис. 14: 2

Используем функцию из предыдущего пункта и преобразуем наш сигнал, а после сделаем наложение спектров:

При наложении видно, что спад происходит как в условии

Отобразим полученную нами wave:

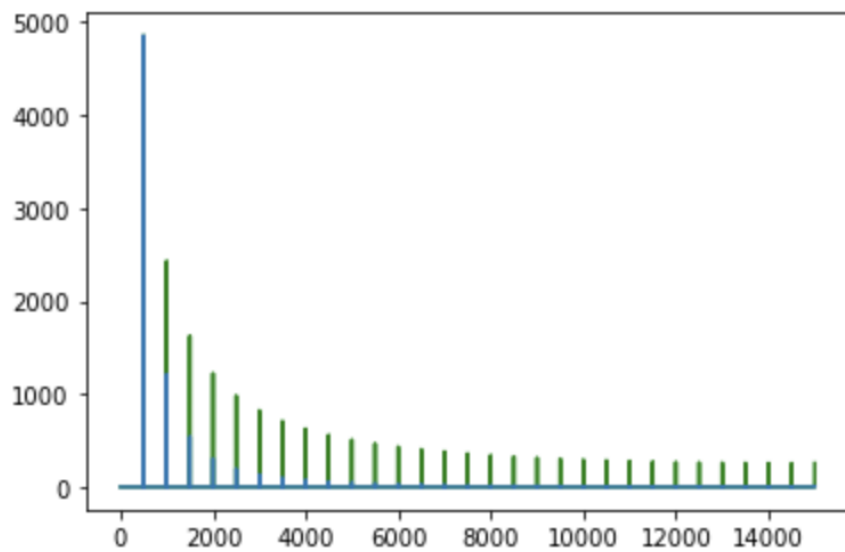


Рис. 15: 2

Как видно из графиков, полученных после обработки сигнала, спектр спадает пропорционально как сказано в условии и при этом имеет четные и нечетные графики. Как видно из последнего графика, сигнал перестал быть пилообразным, и стал похожим на синусоидный.