

ThinkDSP. Лабораторная 2. Гармоники.

Шерепа Никита

26 апреля 2021 г.

Содержание

1	Упражнение 2.2	5
2	Упражнение 2.3	9
3	Упражнение 2.4	11
4	Упражнение 2.5	13
5	Упражнение 2.6	15
6	Вывод	18

Список иллюстраций

1	Пилообразный звук	6
2	Спектр пилообразного сигнала	6
3	Сравнение гармоник пилообразного сигнала (синий) и прямоугольного сигнала (красный)	7
4	Сравнение гармоник пилообразного сигнала (синий) и треугольного сигнала (красный)	8
5	Спектр созданного сигнала	9
6	Спектр созданного сигнала	11
7	Синий - первоначальный <i>Spectrum.hs[0]</i> . Красный - <i>Spectrum.hs[0]</i> = 100	12
8	Измененный спектр	14
9	Пилообразный спектр	15
10	Сравнение начального спектра и измененного спектра . . .	16
11	Получившийся результат	17

Листинги

1	Класс SawtoothSignal	5
2	Класс SawtoothSignal	5
3	Визуализация звука	5
4	Спектр пилообразного сигнала	6
5	Создание прямоугольного сигнала	7
6	Создание треугольного сигнала	7
7	Создание сигнала и его спектра	9
8	Прослушивание сигнала	9
9	Создание сигнала с частотой 300 Гц и его прослушивание .	10
10	Создание треугольного сигнала	11
11	Первый элемент спектра	12
12	Первый элемент спектра	12
13	Функция для изменения спектра	13
14	Вычисление первоначального спектра и его прослушивание	13
15	Изменение спектра и визуализация результата	13
16	Воспроизведение отфильтрованного спектра	14
17	Изменение спектра и визуализация результата	15
18	Изменение спектра	16
19	Получившийся результат	16

1 Упражнение 2.2

1. Задание

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?

2. Ход работы

Напишем класс *SawtoothSignal*

```
1     from thinkdsp import Sinusoid
2     from thinkdsp import normalize, unbias
3     import numpy as np
4
5     class SawtoothSignal(Sinusoid):
6         def evaluate(self, ts):
7             cycles = self.freq * ts + self.offset / np.pi / 2
8             frac, _ = np.modf(cycles)
9             ys = normalize(unbias(frac), self.amp)
10            return ys
```

Листинг 1: Класс *SawtoothSignal*

Теперь сделаем пилообразный звук и прослушаем его

```
1     sawtooth = SawtoothSignal().make_wave(duration=0.05,
2         framerate=40000)
3     sawtooth.make_audio()
```

Листинг 2: Класс *SawtoothSignal*

Получился очень острый, высокий и неприятный звук Визуализируем его

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

Листинг 3: Визуализация звука

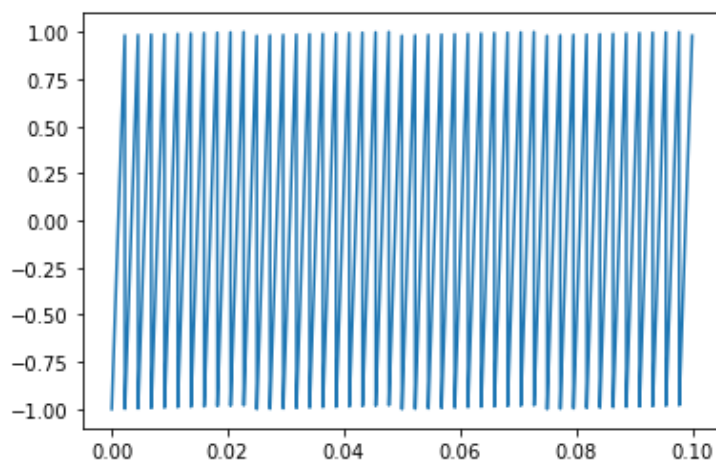


Рис. 1: Пилообразный звук

Теперь вычислим спектр пилообразного сигнала

```
1 sawtooth.make_spectrum().plot()
```

Листинг 4: Спектр пилообразного сигнала

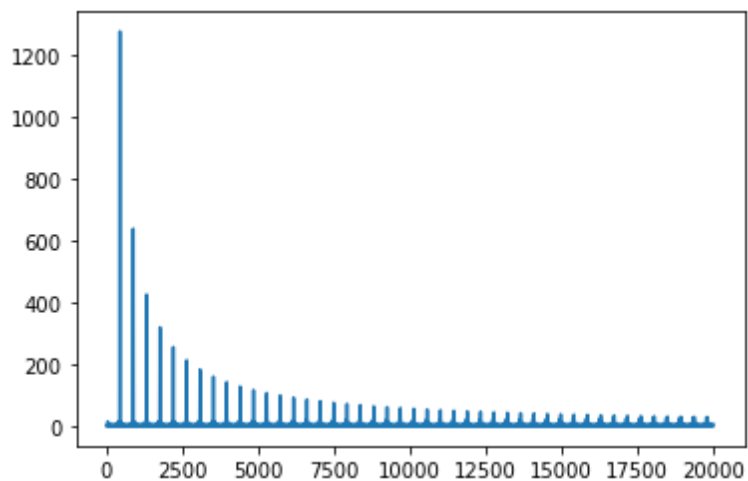


Рис. 2: Спектр пилообразного сигнала

Теперь составим прямоугольный сигнал и сравним его гармоническую структуру с гармонической структурой пилообразного сигнала

```

1 from thinkdsp import SquareSignal
2
3 sawtooth.make_spectrum().plot(color='red')
4 square = SquareSignal(amp=0.5).make_wave(duration=0.1,
5         framerate=40000)
6 square.make_spectrum().plot()

```

Листинг 5: Создание прямоугольного сигнала

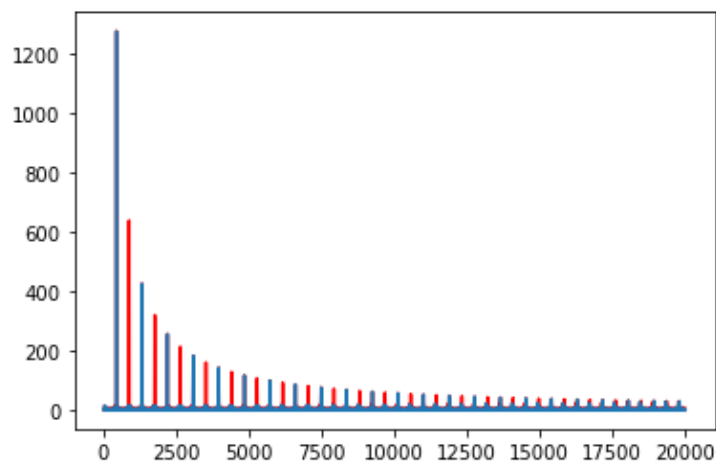


Рис. 3: Сравнение гармоник пилообразного сигнала (синий) и прямоугольного сигнала (красный)

Как видим, гармоники прямоугольного сигнала убывают аналогично гармоникам пилообразного сигнала. При этом у прямоугольного сигнала присутствуют как четные, так и нечетные гармоники.

Теперь составим треугольный сигнал и сравним его гармоническую структуру с гармонической структурой пилообразного сигнала

```

1 from thinkdsp import SquareSignal
2
3 sawtooth.make_spectrum().plot(color='red')
4 square = SquareSignal(amp=0.5).make_wave(duration=0.1,
5         framerate=40000)
6 square.make_spectrum().plot()

```

Листинг 6: Создание треугольного сигнала

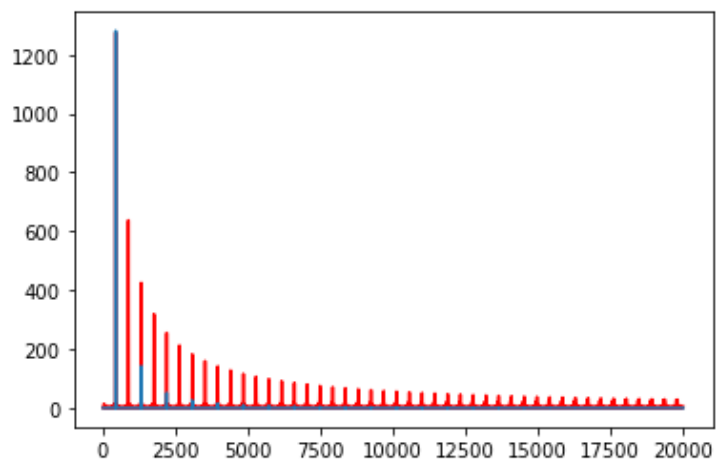


Рис. 4: Сравнение гармоник пилообразного сигнала (синий) и треугольного сигнала (красный)

Как видим, гармоники треугольного сигнала убывают не так резко по сравнению с гармониками пилообразного сигнала. При этом у треугольного сигнала присутствуют как четные, так и нечетные гармоники.

2 Упражнение 2.3

1. Задание

Создайте прямоугольный сигнал 1100 Гц и вычислите *wave* с выборками 10 000 кадров в секунду. Постройте спектр и убедитесь, что большинство гармоник "завернуты" из-за биений. Слышны ли последствия этого при проигрывании?

2. Ход работы

Создадим указанный сигнал и построим его спектр

```
1 square = SquareSignal(1100).make_wave(duration=0.5,  
    framerate=10000)  
2 square.make_spectrum().plot()
```

Листинг 7: Создание сигнала и его спектра

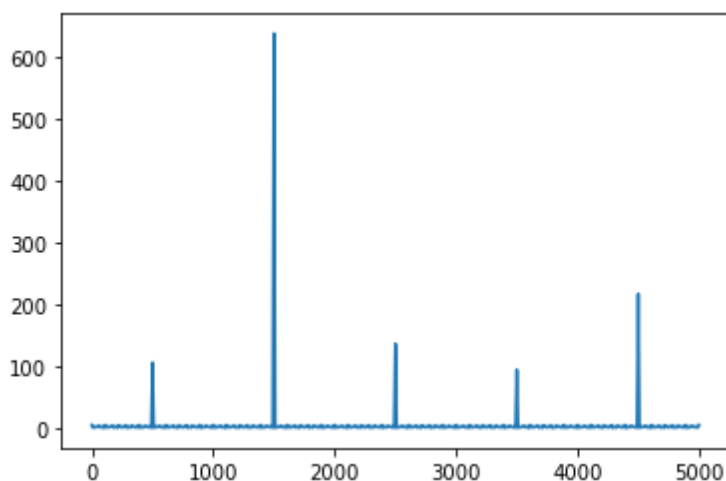


Рис. 5: Спектр созданного сигнала

Основная гармоника - 1100 Гц Первая гармоника - 3300 Гц Вторая гармоника - 4500 Гц, хотя должна быть на 5500 Гц Третья гармоника - 2300 Гц, хотя должна быть на 7700 Гц

Теперь прослушаем получившийся сигнал

```
1 square.make_audio()
```

Листинг 8: Прослушивание сигнала

Теперь как раз можно услышать эти сместившиеся гармоники. Теперь создадим сигнал с частотой 300 Гц и прослушаем.

```
1     from thinkdsp import SinSignal
2
3     SinSignal(300).make_wave(duration=0.1,
        framerate=10000).make_audio()
```

Листинг 9: Создание сигнала с частотой 300 Гц и его прослушивание

Разница есть. Частоту можно отчетливо прослушать.

3 Упражнение 2.4

1. Задание

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:

- (a) Создайте треугольный сигнал с частотой 440 Гц и *wave* длительностью 0.01 секунд. Распечатайте сигнал
- (b) Создайте объект *Spectrum* и распечатайте *Spectrum.hs[0]*. Каковы амплитуда и фаза этого компонента?
- (c) Установите *Spectrum.hs[0] = 100*. Как эта операция повлияет на сигнал? Подсказка: *Spectrum* дает метод, называемый *make_wave*, вычисляющий *wave*, соответствующий *Spectrum*.

2. Ход работы

- (a) Создадим треугольный сигнал и распечатаем его.

```
1         triangle =  
            TriangleSignal(440).make_wave(duration=0.01)  
2         triangle.plot()
```

Листинг 10: Создание треугольного сигнала

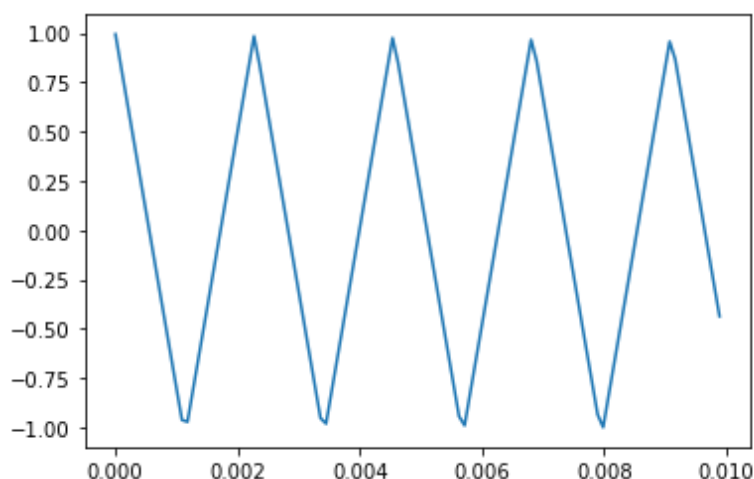


Рис. 6: Спектр созданного сигнала

(b) Теперь выведем первый элемент спектра

```
1 spectrum = triangle.make_spectrum()  
2 spectrum.hs[0]  
3  
4 (1.0436096431476471e-14+0j)
```

Листинг 11: Первый элемент спектра

Получили комплексное число

Если добавить в компонент нулевой частоты какое-то число -
добавиться вертикальное смещение спектра

(c) Установим $Spectrum = 100$

```
1 spectrum.hs[0] = 100  
2 triangle.plot(color='red')  
3 spectrum.make_wave().plot()
```

Листинг 12: Первый элемент спектра

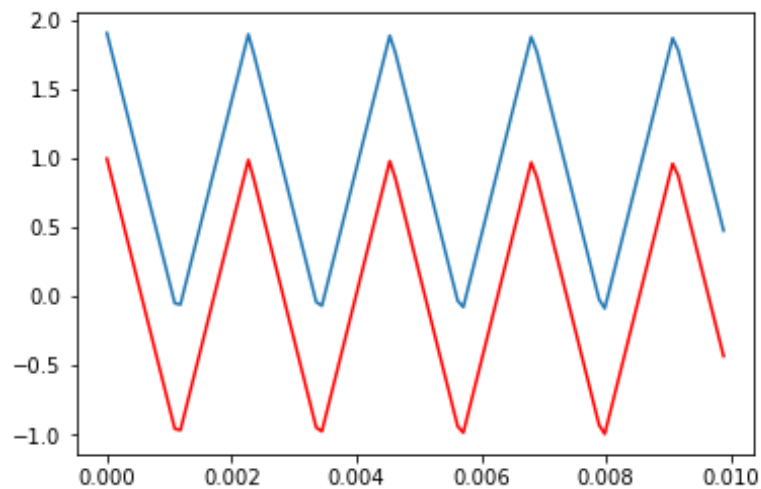


Рис. 7: Синий - первоначальный $Spectrum.hs[0]$. Красный - $Spectrum.hs[0] = 100$

Видим, что спектр сместился по вертикали относительно своего первоначального положения (красный график)

4 Упражнение 2.5

1. Задание

Напишите функцию, принимающую *Spectrum* как параметр и изменяющую его делением каждого элемента *hs* на соответствующую частоту из *fs*. Подсказка: поскольку деление на ноль не определено, надо задать *Spectrum.hs[0] = 0*.

Проверьте эту функцию, используя прямоугольный, треугольный или пилообразный сигналы:

- (a) Вычислите *Spectrum* и распечатайте его
- (b) Измените *Spectrum*, вновь используя свою функцию, и распечатайте его
- (c) Используйте *Spectrum.make_wave*, чтобы сделать *wave* из измененного *Spectrum*, и прослушайте его. Как эта операция повлияла на сигнал?

2. Ход работы

Напишем функцию для изменения спектра

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

Листинг 13: Функция для изменения спектра

Теперь вычислим первоначальный спектр и прослушаем

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

Листинг 14: Вычисление первоначального спектра и его прослушивание

Звучит как гудок в телефоне

Теперь, используя написанную функцию, изменим спектр

```
1 spectrum = wave.make_spectrum()
2 spectrum.plot(high=10000, color='red')
3 filter_spectrum(spectrum)
4 spectrum.scale(440)
5 spectrum.plot(high=10000)
```

Листинг 15: Изменение спектра и визуализация результата

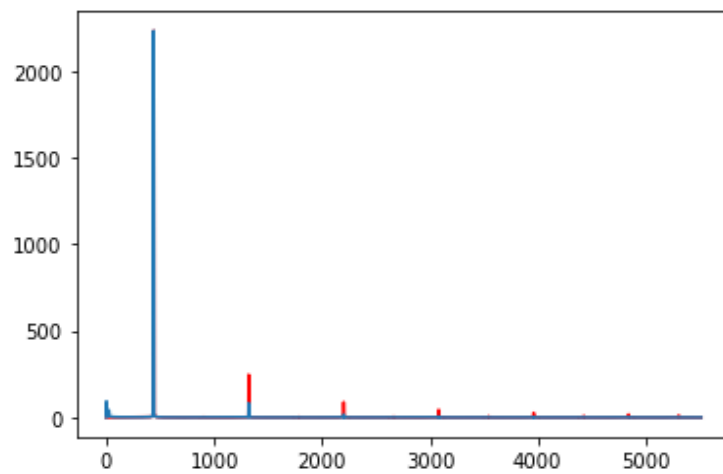


Рис. 8: Измененный спектр

И прослушаем получившийся результат

```
1     filtered = spectrum.make_wave()  
2     filtered.make_audio()
```

Листинг 16: Воспроизведение отфильтрованного спектра

Звук стал более глухой. По графику видно, что измененный спектр подавляет низкие гармоники исходного спектра и тем самым фильтрует низкие частоты.

5 Упражнение 2.6

1. Задание

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 $1/f^2$?

Подсказка: для этого есть два способа.

- (a) Можно собрать желаемый сигнал из синусоид
- (b) Можно взять сигнал со спектром, похожим на необходимый, и изменять его параметры

2. Ход работы

Создадим пилообразный спектр и визуализируем его

```
1     freq = 500
2     signal = SawtoothSignal(freq=freq)
3     wave = signal.make_wave(duration=0.5, framerate=20000)
4     wave.make_audio()
5     spectrum = wave.make_spectrum()
6     spectrum.plot()
```

Листинг 17: Изменение спектра и визуализация результата

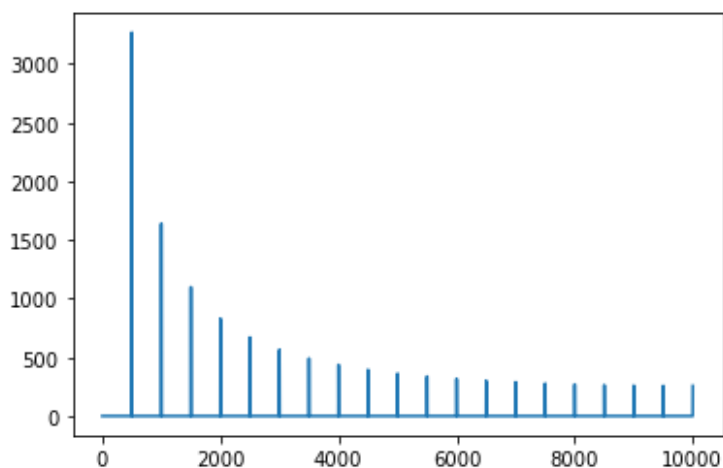


Рис. 9: Пилообразный спектр

Теперь воспользуемся функцией из прошлого упражнения, изменим полученный спектр и сравним результаты

```
1 spectrum.plot(color='red')
2 filter_spectrum(spectrum)
3 spectrum.scale(freq)
4 spectrum.plot()
```

Листинг 18: Изменение спектра

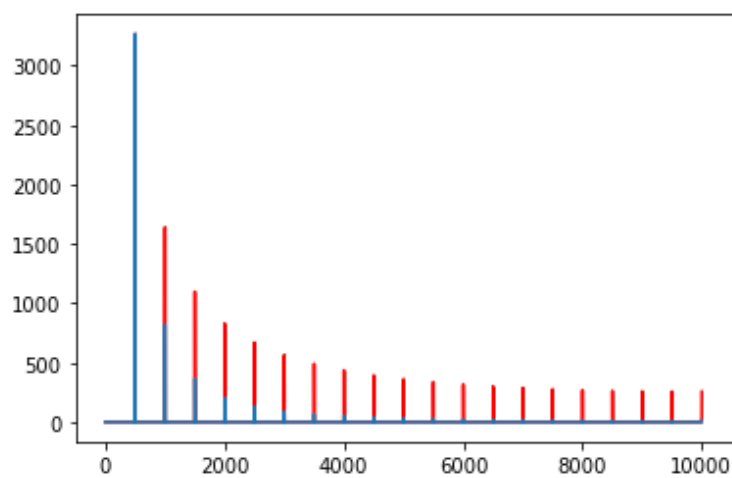


Рис. 10: Сравнение начального спектра и измененного спектра

Как видим, теперь гармоники получились схожими. Они спадают пропорционально $1/f^2$

Теперь посмотрим, как выглядит получившийся результат

```
1 wave.segment(duration=0.01).plot()
```

Листинг 19: Получившийся результат

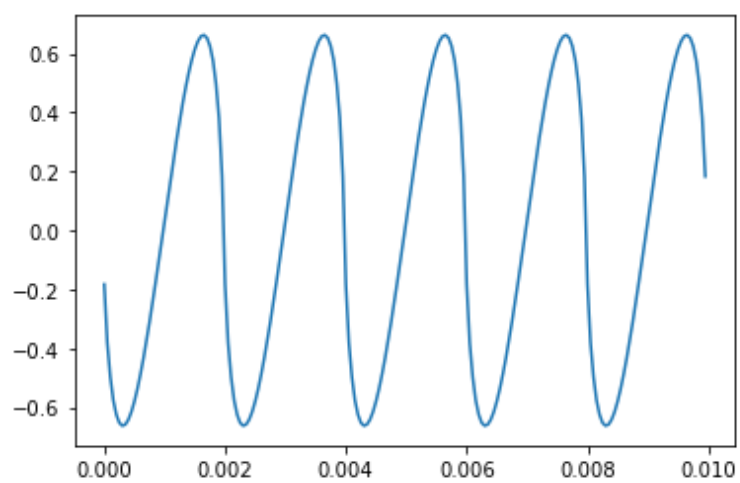


Рис. 11: Получившийся результат

Видим, что звук похож на синусоиду.

6 Вывод

В результате выполнения лабораторной работы получены навыки обработки пилообразных, прямоугольных, треугольных сигналов и их гармоник. Также в данной работе мы ознакомились с явлением смещения гармоник в цифровой обработке сигналов.