Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Телекоммуникационные технологии

Отчёт по лабораторной работе №2

Работу выполнил:
Смирнов Л. Д.
Группа:
3530901/80202
Преподаватель:
Богач Н. В.

Санкт-Петербург
2021

Содержание

1. Теоретическая часть

В главе, на которой основана данная лабораторная работа рассматривается связь между формой сигналов и их спектрами на основе треугольных и прямоугольных сингналов, а также явление, называемое биением (aliasing). Треугольные сигналы, похожие на спрямленную синусоиду, имеют при разложении в спектр наибольшую гармонику на оновной частоте сигнала и допольнительные на **нечетных** кратных основной частотах. Спадание амплитуд гармоник идет с квадратичной скоростью. Соответствующая завимосвязь между амплитудами ближайших гармоник называется **гармонической зависимостью**. Прямоугольные сигналы, аналогично треугольным, имеют только нечетные грамоники, кратные основной, но спадают они с прямой скоростью, а не квадратичной. Биением называют явление, из-за которого можно перепутать сигналы, сумма частот которых равна выборке, например 6450 Гц и 3550 Гц при 10000 кадров в секунду при разложении в спектр будут выглядеть одинаково. Частоты после определенного значания "заворачиваются" и продолжают записываться в обратную сторону. Таким образом получается зигзаообразная лесенка из амплитуд.

2. Выполнение работы

2.1. Упражнение 1

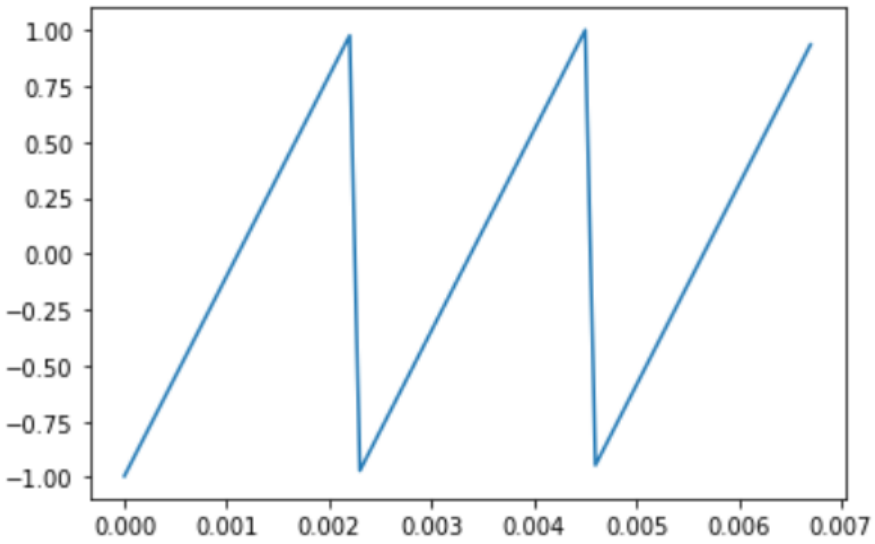
В качетсве выполнения первого упражнения я открыл и изучил примеры к изучаемой главе.

2.2. Упражнение 2

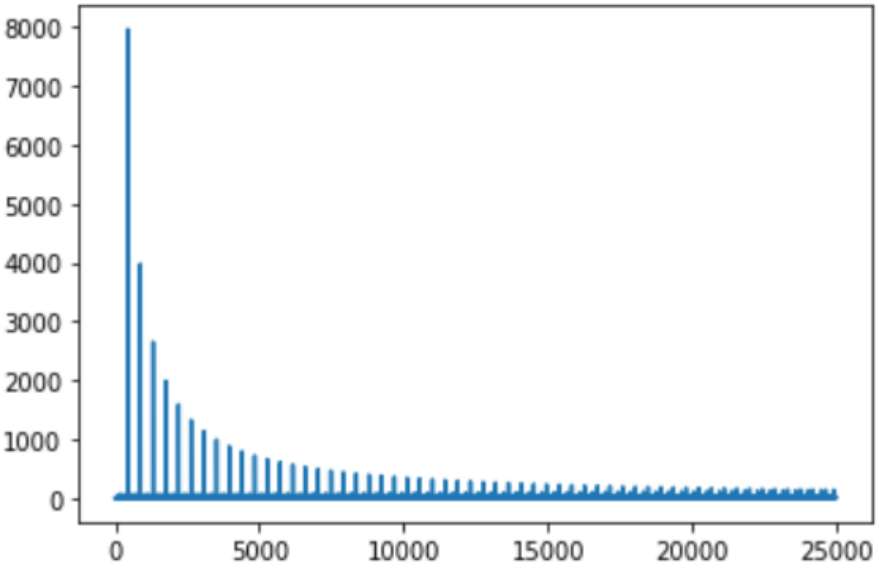
В первую очередь я написал класс для пилообразных сигналов.

```
from thinkdsp import Sinusoid
from thinkdsp import normalize, unbias

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
```

Чтобы убедиться в правильности созданного описания я вывел волну длительностью в 3 периода пилообразного сигнала.



Визуально это похоже на пилообразный сигнал, поэтому пошел дальше и вывел разложение в спектр.

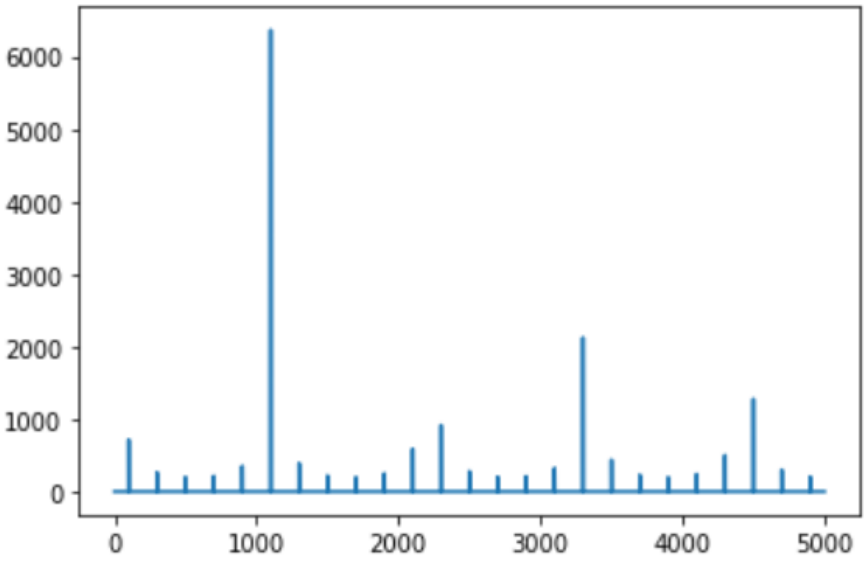


Как можно видеть, гармоники спектра данного сигнала похожи на гармоники для прямоугольного сигнала (убывают с прямой зависимостью) за исключением того, что у пилообразного их в 2 раза больше. Соответственно также, как и гармоники прямоугольного сигнала, гармоники пилообразного убывают медленнее, чем у треугольного (прямая зависимость против квадратичной соответственно).

2.3. Упражнение 3

Согласно указаниям в учебнике я создал прямоугольный сигнал, волну на его основе, и раложение в спектр (код и разложение приведены ниже).

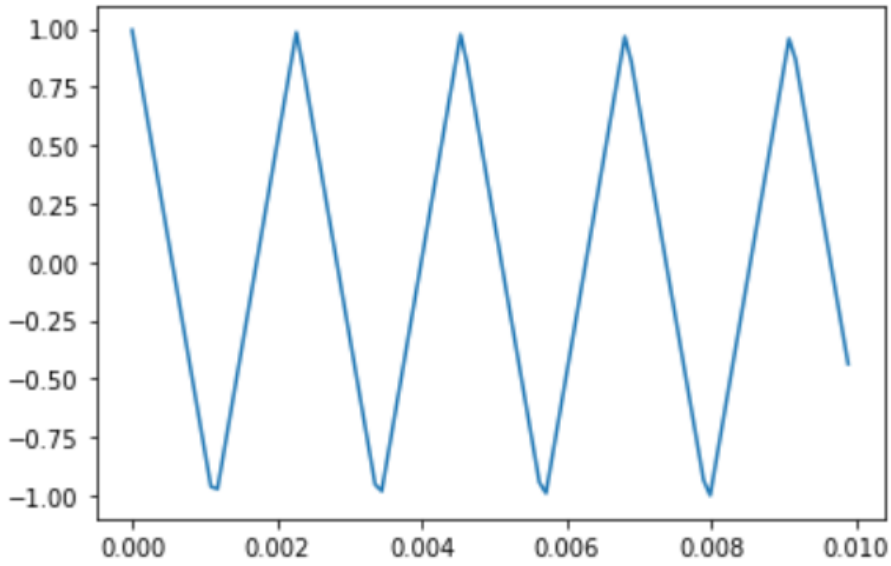
```
sq_signal = SquareSignal(1100)
sq_wave = sq_signal.make_wave(duration=1, framerate=10000)
sq_spectrum = sq_wave.make_spectrum()
sq_spectrum.plot()
```



Как можно видеть, все гармоники, начиная с четвертой "завернулись", из-за чего звук, получившийся в результате воспроизведения волны, звучит "грязно", как будто 2 звука с вызоким и низким тембрами наложили друг на друга.

2.4. Упражнение 4

Как указано в учебнике, я создал треугольный сигнал с частотой в 440 Гц, волну на его основе продолжительностью 0.01 сек и вывел ее.

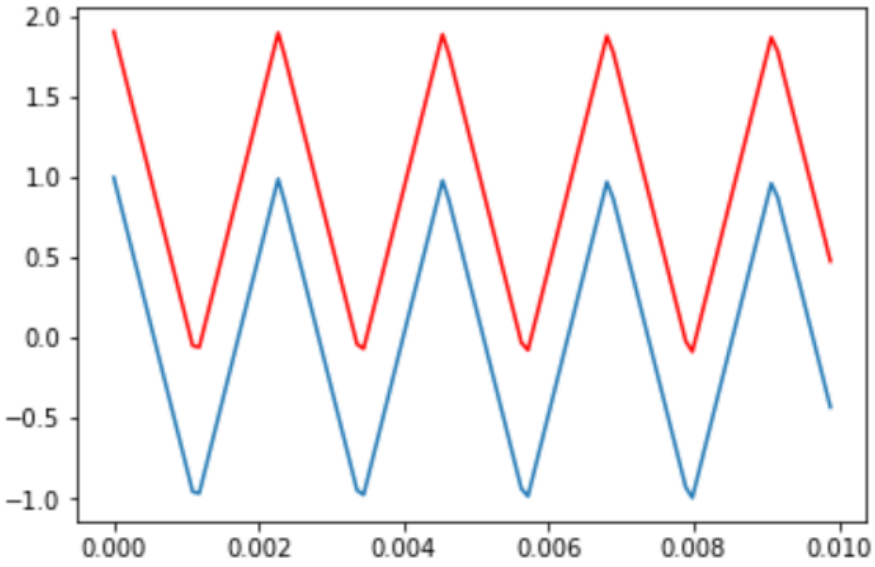


Затем получившуюся волну разложил в спектр и вывел первый элемент массива с аплитудами и фазами частотных компонент сигнала. Получено число 1.0436096431476471e-14+0j.

Изменение этого значение на 100, как можно видеть по Рисунку **??** (новая волна выделена красным цветом), привело к смещению всех значений волны примерно на 1 пункт вверх.

Полный код для этого упражнения:

```
tri_sig = TriangleSignal(440)
```



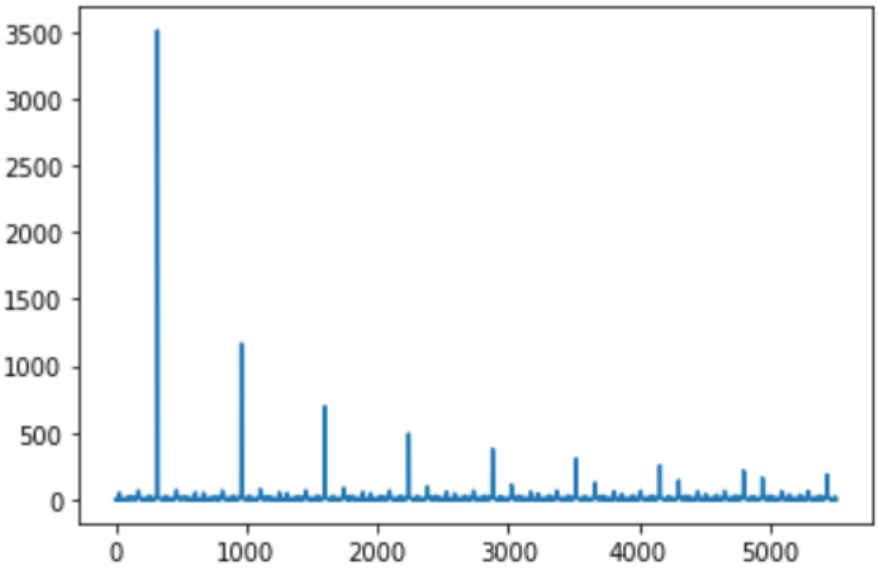

```
tri_wave = tri_sig.make_wave(duration=0.01)
tri_wave.plot()

tri_spectrum = tri_wave.make_spectrum()
tri_spectrum.hs[0]

tri_spectrum.hs[0] = 100
tri_wave_1 = tri_spectrum.make_wave()
tri_wave.plot()
tri_wave_1.plot(color='red')
```

2.5. Упражнение 5

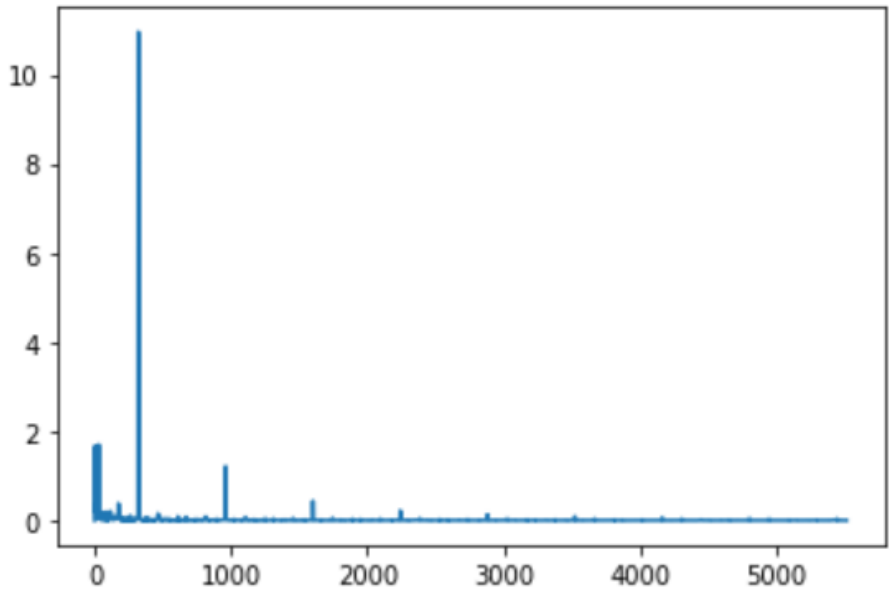
Для начала я написал необходимую функцию-делитель:

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

Для исследования ее поведения я создал прямоугольный сигнал. Его разложение в спектр до использования функции выглядит следующим образом:



После того, как я применил к спектру свою функцию, он стал выглять вот так:

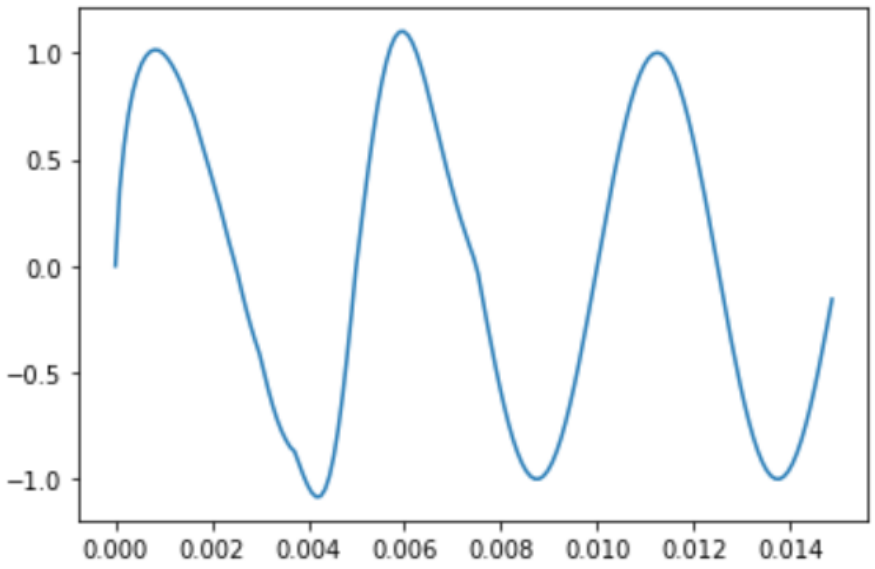


Как можно видеть, применение функции уменьшило все гармоники (как пример, для основной гармоники с *примерно* 3500 до 11).

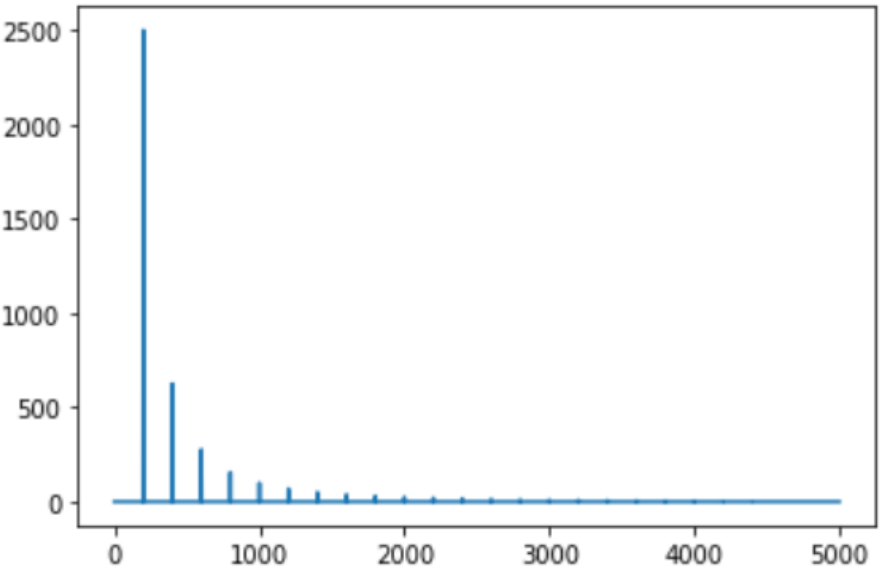
Волна, сделанная на основе получившегося спектра при воспроизведении звучит глуше и с меньшим дребезгом.

2.6. Упражнение 6

Я реши пойти первой из двух предложенных дорог и создать нужный сигнал путем сложения синусоид, общая формула которых $sin(2 * i * \pi * x)/i^2$. Получился сигнал следующего вида:



Ниже представлено разложение этого сигнала в спектр. Как можно видеть, убывание гармоник похоже на квадратичное, и присутствуют как четные, так и нечетные основной гармоники.



Полный код для этого упражнения приведен ниже.

```
sum_wave = 0
for i in range(1, 101):
    sin_sig = SinSignal(freq=200*i, amp=1.0, offset=0)
    sin_wave = sin_sig.make_wave(duration=0.5, framerate=10000)
    sin_wave.scale(1/(i*i))
    sum_wave += sin_wave
sum_wave.plot()

sum_wave.make_spectrum().plot()
```

3. Выводы

В ходе выполнения этой лабораторной работы я ознакомился с треугольными, прямоугольными и пилообразными сигналами и изучил взаимодействие со спектрами. Упражнения помогли лучше усвоить пройденный материал и закрепить навыки работы с синусоидальными сигналами.