

Лабораторная работа 4

Чернышев Ярослав

31 мая 2021 г.

Оглавление

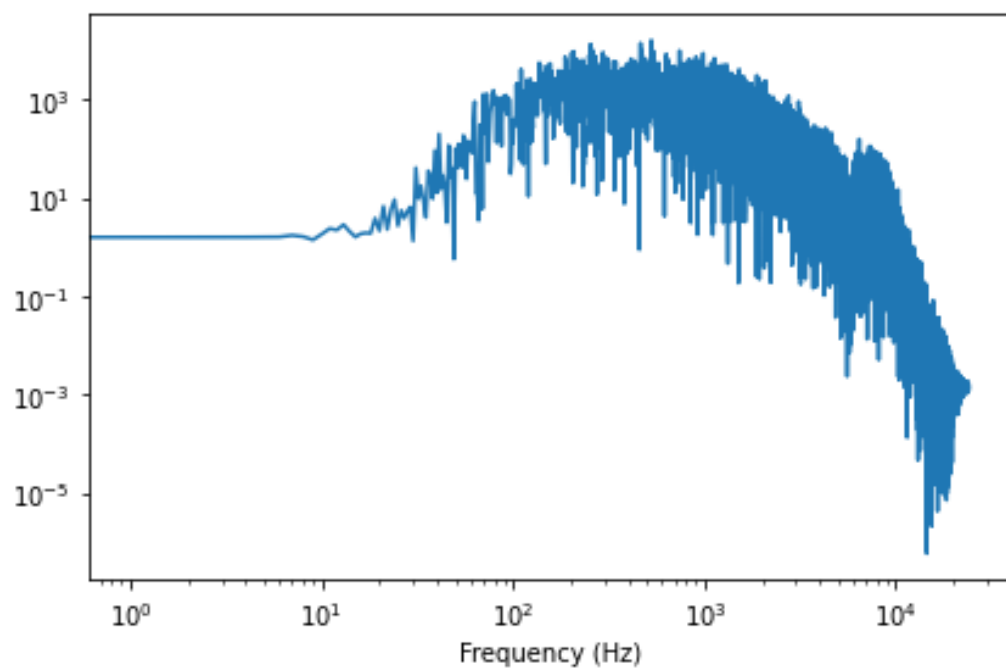
1	Задание 4.1	2
2	Задание 4.2	4
3	Задание 4.3	6
4	Задание 4.4	7
5	Задание 4.5	9

Глава 1

Задание 4.1

В данном задании требуется скачать звуки волн, извлечь секунднй фрагмент и проанализировать его.

```
1 from thinkdsp import read_wave
2
3 wave = read_wave('132736__ciccarelli__ocean-waves.wav')
4
5 segment = wave.segment(start=1.5, duration=1.0)
6
7 spectrum = segment.make_spectrum()
8
9 spectrum.plot_power()
10 loglog = dict(xscale='log', yscale='log')
11 decorate(xlabel='Frequency (Hz)', **loglog)
12
```



Взятие других сегментов из аудиозаписи не изменило общий вид графика. Можно предположить, что спектр больше всего похож на красный шторм.

Глава 2

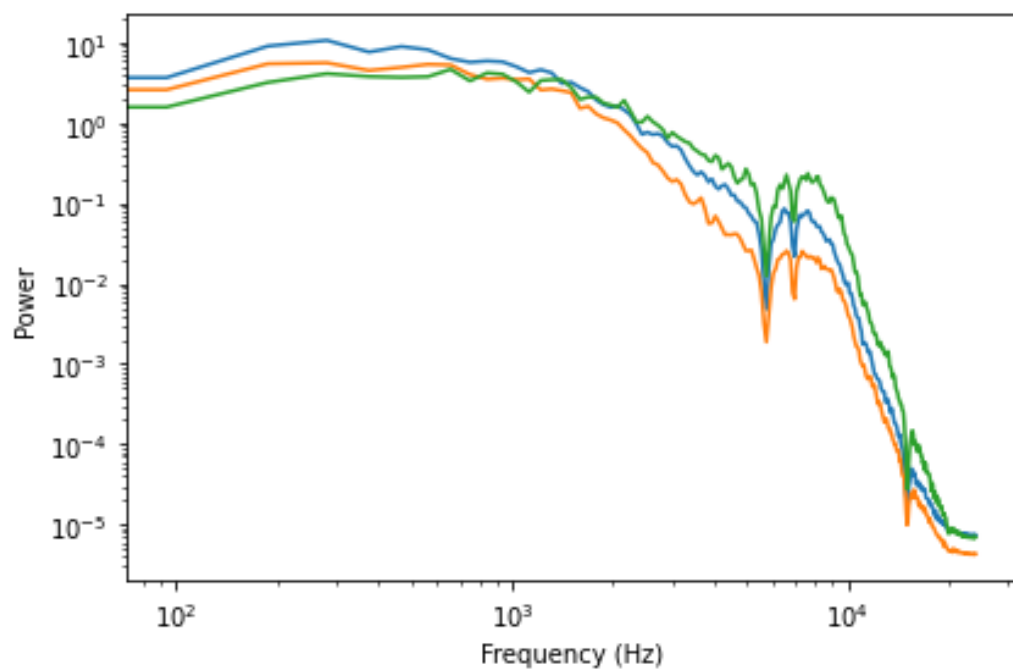
Задание 4.2

В этом задании требуется создать и протестировать функцию, реализующую метод Барлета.

Первым делом, напомним соответствующий класс:

```
1 def bartlett_method(wave, seg_length=512, win_flag=True):
2
3     spectro = wave.make_spectrogram(seg_length, win_flag)
4     spectrums = spectro.spec_map.values()
5
6     psds = [spectrum.power for spectrum in spectrums]
7
8     hs = np.sqrt(sum(psds) / len(psds))
9     fs = next(iter(spectrums)).fs
10
11     spectrum = Spectrum(hs, fs, wave.framerate)
12     return spectrum
13
```

Были взяты три сегмента из звуков волны и получены следующие спектры мощностей:



Полученное изображение позволяет провести сравнение зависимостей мощностей от частот на разных промежутках волны.

Глава 3

Задание 4.3

В этом задании требуется проанализировать динамику стоимости биткоина как шум.

```
1 import pandas as pd
2
3 df = pd.read_csv('BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv',
4                  , parse_dates=[0])
5
6 ys = df['Closing Price (USD)']
7 ts = df.index
8
9 from thinkdsp import Wave
10
11 wave = Wave(ys, ts, framerate=1)
12 spectrum = wave.make_spectrum()
13 spectrum.estimate_slope()[0]
```

Получили следующий вывод:

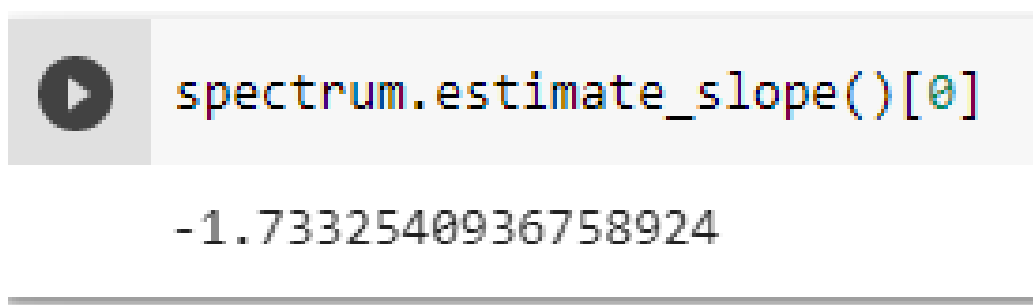


Рис. 3.1: Искомая волна

Из этого можно решить, что полученный шум близок к розовому.

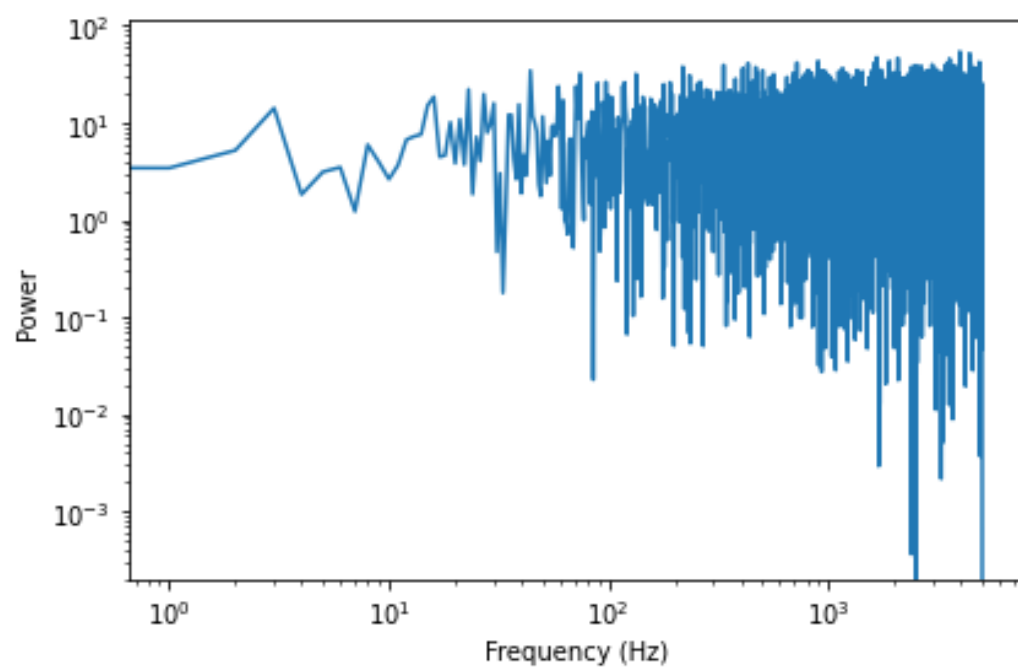
Глава 4

Задание 4.4

Необходимо реализовать сигнал в соответствии с распределением Пуассона. Ниже привожу класс, создающий этот шум:

```
1 class UncorrelatedPoissonNoise(Noise):  
2  
3     def evaluate(self, ts):  
4         ys = np.random.poisson(self.amp, len(ts))  
5         return ys
```

Я поэкспериментировал с этим классом, создав несколько экземпляров с разными амплитудами. Тестирование через прослушивание показало, получившийся звук - серия тресков, с увеличением амплитуды сливающихся в белый шум. Ниже приведён спектр одного из сигналов с большой амплитудой:



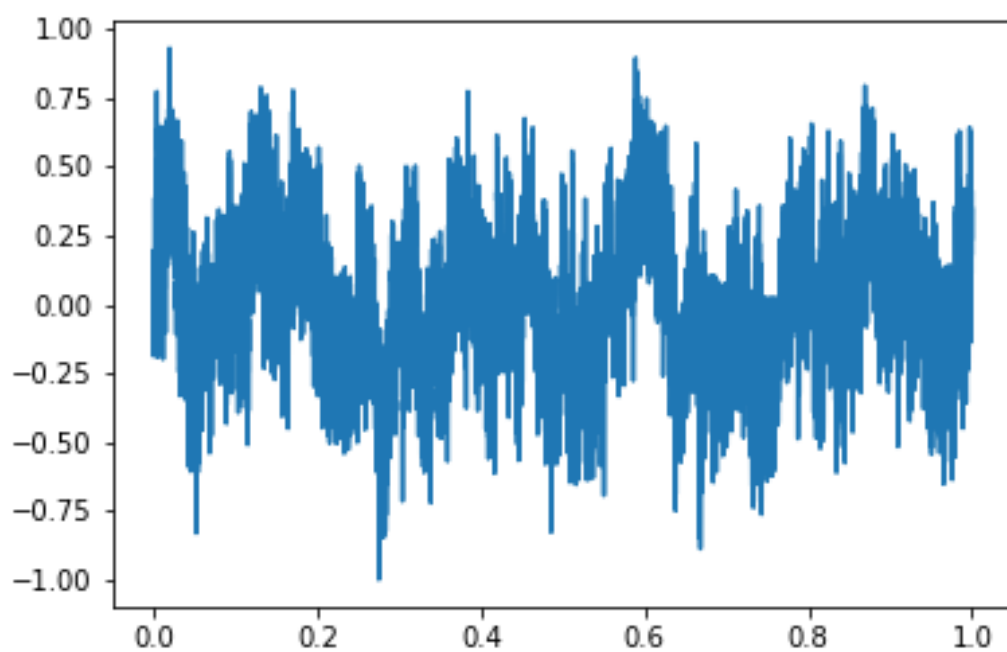
Глава 5

Задание 4.5

Требуется реализовать алгоритм Восса-МакКартни:

```
1 def voss(nrows, ncols=16):
2     array = np.empty((nrows, ncols))
3     array.fill(np.nan)
4     array[0, :] = np.random.random(ncols)
5     array[:, 0] = np.random.random(nrows)
6     n = nrows
7     cols = np.random.geometric(0.5, n)
8     cols[cols >= ncols] = 0
9     rows = np.random.randint(nrows, size=n)
10    array[rows, cols] = np.random.random(n)
11
12    df = pd.DataFrame(array)
13    df.fillna(method='ffill', axis=0, inplace=True)
14    total = df.sum(axis=1)
15
16    return total.values
```

На основе этой функции была создана волна:



А также её спектр:

