

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

Телекоммуникационные технологии

Отчёт по лабораторной работе №4

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

Санкт-Петербург
2021

Содержание

1. Теоретическая часть	3
2. Выполнение работы	3
2.1. Упражнение 1	3
2.2. Упражнение 2	4
2.3. Упражнение 3	5
2.4. Упражнение 4	5
2.5. Упражнение 5	6
3. Выводы	8

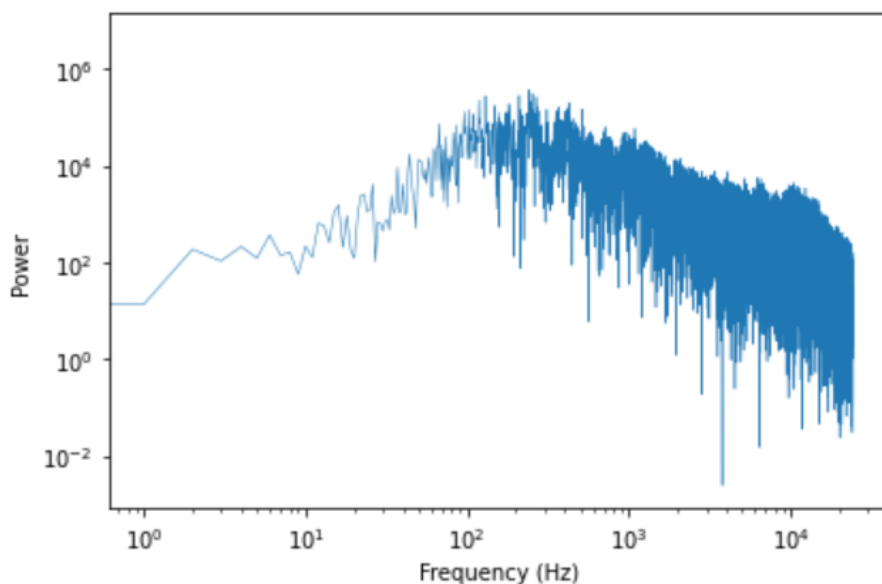
1. Теоретическая часть

В данной главе рассматривается понятие шумов — то есть сигналов, содержащих компоненты с разными частотами, однако не имеющих гармонической структуры периодических сигналов. Ниже кратко приведено описание каждого из рассмотренных видов шумов. Некоррелированный равномерный шум — сигнал, имеющий случайные значения из равномерного распределения, которые не зависят друг от друга. Его мощность равномерна распределена по всем частотам. Также называют белым шумом. Броуновский шум — отличается от некоррелированного тем, что его значения зависят от предыдущих, а именно от предыдущего и еще одного случайного. Спектр мощности броуновского шума имеет уклон равный -2 , из-за чего его называют красным шумом, то есть шумом, большая часть мощности которого приходится на низкие частоты. Розовый шум — середина между белым и красным шумами. То есть коэффициент β в соотношении $P = K/f^\beta$ принимает значения от 0 (белый шум) до 2 (красный шум). Соответственно меняется и наклон спектра мощности от горизонтального, до наклонного, который рассматривался в части с броуновским шумом. Гауссов шум - альтернатива некоррелированного равномерного шума, в которой значения имеют гауссово распределение. Так же является белым шумом, так как наклон спектра мощностей равен нулю.

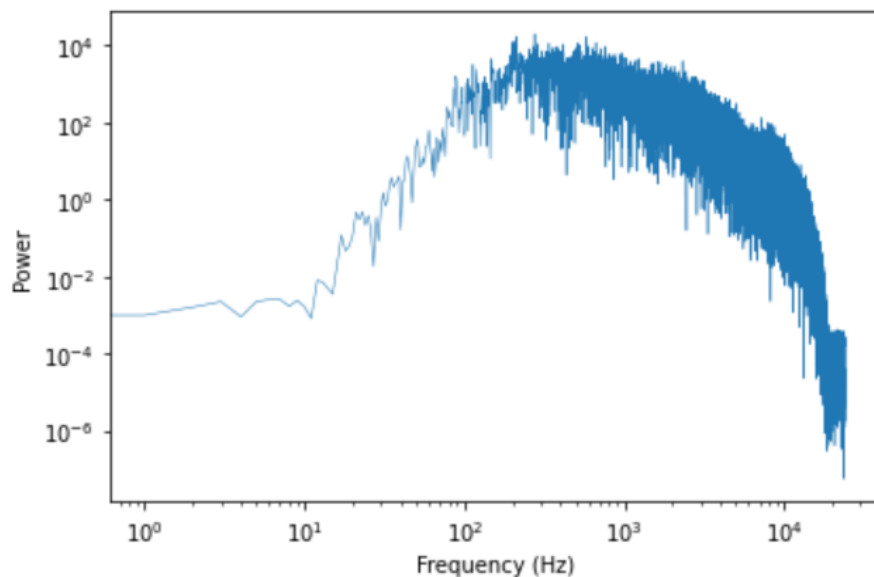
2. Выполнение работы

2.1. Упражнение 1

Для этого упражнения я скачал 2 записи звуков волн и водопада, извлек из них по секундному фрагменту и получил следующие изображения спектров (первое для водопада, второе для волн):



Полученные изображения не похожи ни на одно из рассмотренных ранее, так как первая часть спектра примерно до 100 Герц возрастает по мощности, а вторая начинает резко убывать, причем во втором случае (для волн) нелинейно. Взятие других сегментов аудио дорожек не изменило общей картины, поэтому могу лишь сделать вывод, что похож спектр больше всего на красный шум.



Полный код для упражнения:

```
from thinkdsp import read_wave

fall = read_wave('waterfall.wav')
fall_seg = fall.segment(start = 10.0, duration = 1.0)
fall_seg.make_audio()
fall_spectrum = fall_seg.make_spectrum()
fall_spectrum.plot_power(linewidth=0.5)
decorate(xlabel='Frequency (Hz)', ylabel='Power')
loglog = dict(xscale='log', yscale='log')
decorate(xlabel='Frequency (Hz)', ylabel='Power', **loglog)

waves = read_wave('waves.wav')
waves_seg = waves.segment(duration = 1.0)
waves_seg.make_audio()
waves_spectrum = waves_seg.make_spectrum()
waves_spectrum.plot_power(linewidth=0.5)
decorate(xlabel='Frequency (Hz)', ylabel='Power')
loglog = dict(xscale='log', yscale='log')
decorate(xlabel='Frequency (Hz)', ylabel='Power', **loglog)
```

2.2. Упражнение 2

Для начала я написал необходимую функцию, реализующую метод Барлетта:

```
def bartlettts_method(wave, length=512, flag=True):

    spectro = wave.make_spectrogram(length, flag)
    spectrums = spectro.spec_map.values()

    powers = [spectrum.power for spectrum in spectrums]

    hs = np.sqrt(sum(powers) / len(powers))
```

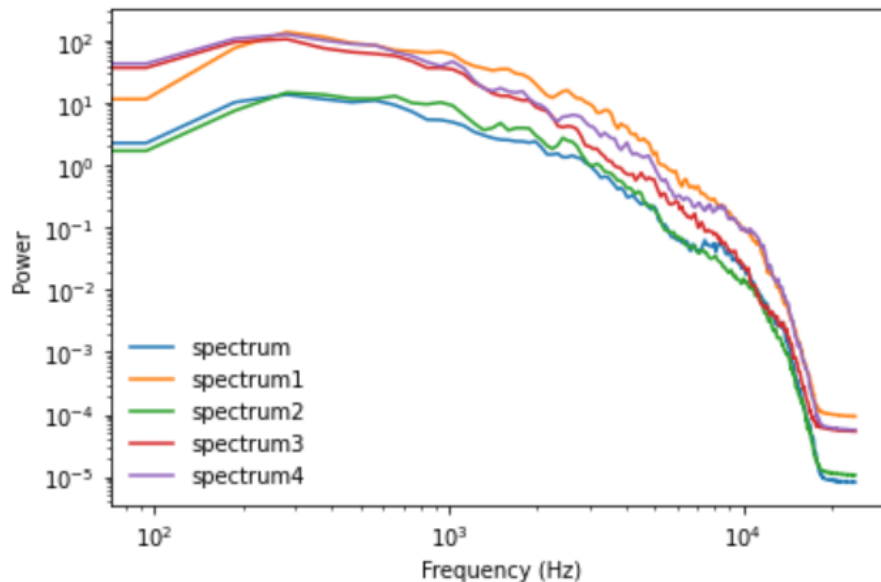
```

fs = next(iter(spectrums)).fs

spectrum = Spectrum(hs, fs, wave. framerate)
return spectrum

```

После этого я создал к одному уже имеющемуся фрагменту звуков волны еще 4 и вывел их спектры мощностей. Полученное изображение позволяет провести сравнение зависимостей мощностей от частот на разных промежутках волны.



2.3. Упражнение 3

Для данного упражнения я написал следующий код:

```

import pandas as pd
from thinkdsp import Wave

df = pd.read_csv('BTC_USD_2013-10-01_2021-04-19-CoinDesk.csv',
                 parse_dates=[0])
ys = df['Closing Price (USD)']
ts = df.index
wave = Wave(ys, ts, framerate=1)
spectrum = wave.make_spectrum()
spectrum.plot_power()
decorate(xlabel='Frequency (1/days)', **loglog)

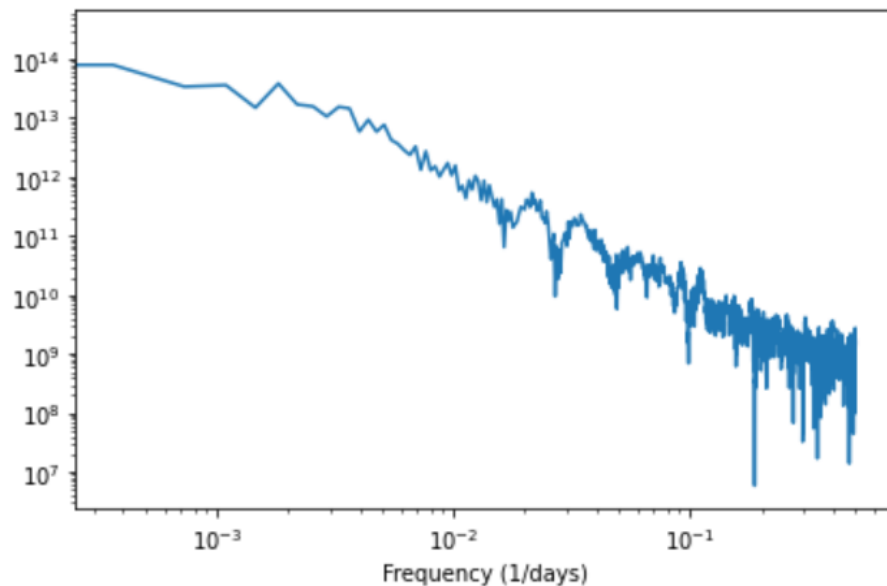
```

Ниже приведен спектр мощности на основе графика цены Биткоина за период с 13 года по сегодняшний день:

Исходя из значения наклона полученного спектра, равного -1.88, можно сказать, что это розовый спектр, который довольно близок к красному.

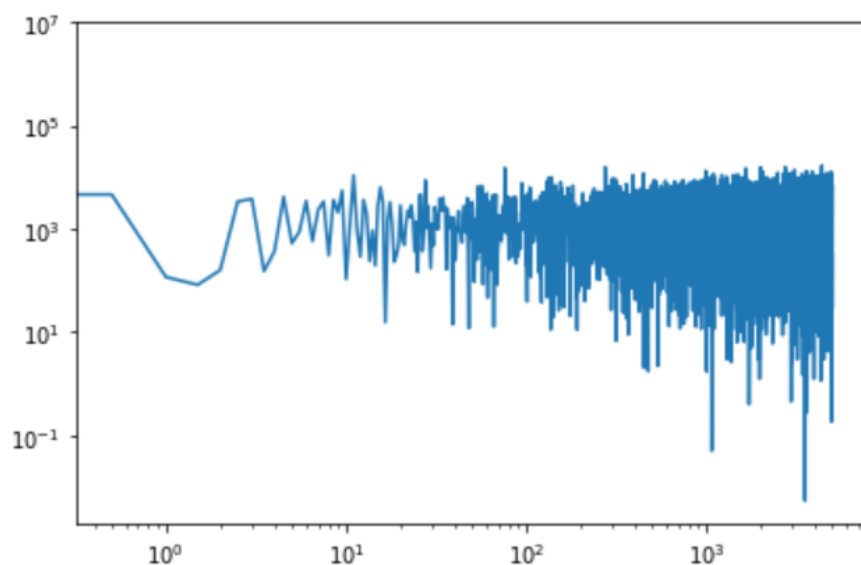
2.4. Упражнение 4

Вот собственно само описание класса, создающего пуассоновский шум:



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

С использованием этого класса я создал 2 сигнала с амплитудой 0.001 и 0.1 и прослушал их. Первый состоит из отдельных коротких "тресков", которые с увеличением амплитуды сливаются и уже во втором сигнале слышатся как белый шум, что легко подтвердилось выводом спектра мощности второго сигнала, который приведен ниже.



2.5. Упражнение 5

Для начала я написал саму функцию, которая реализует алгоритм Восс-МакКартни:

```
def voss(nrows, ncols=16):
    array = np.empty((nrows, ncols))
    array.fill(np.nan)
```

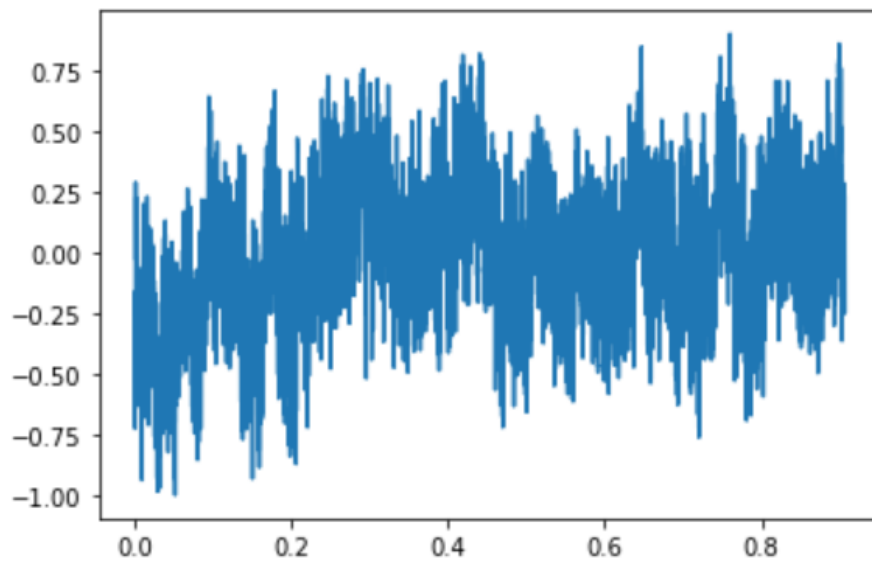
```

array[0, :] = np.random.random(ncols)
array[:, 0] = np.random.random(nrows)
cols = np.random.geometric(0.5, nrows)
cols[cols ≥ ncols] = 0
rows = np.random.randint(nrows, size = nrows)
array[rows, cols] = np.random.random(nrows)
df = pd.DataFrame(array)
df.fillna(method='ffill', axis=0, inplace=True)
total = df.sum(axis=1)

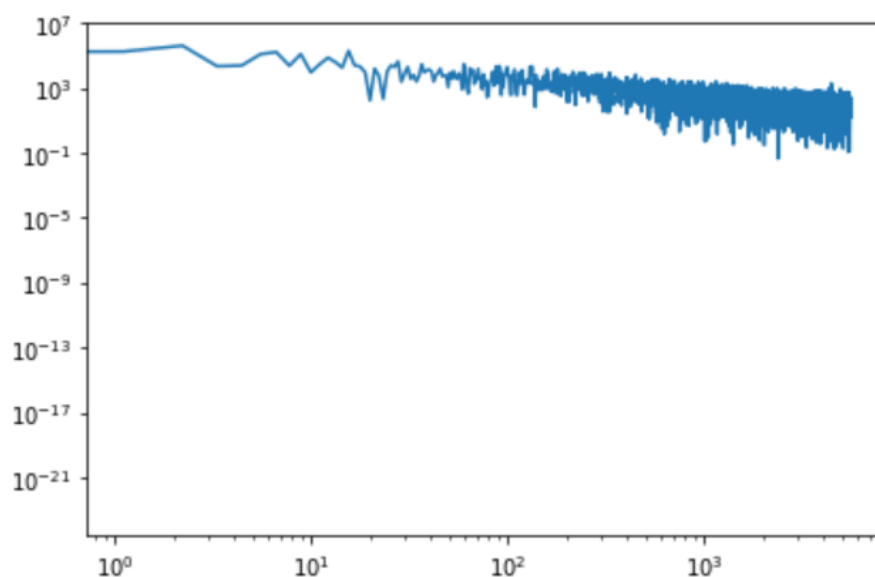
return total.values

```

На основе этой функции я создал волну и изобразил её:



Затем я вывел ее спектр мощности.



Как можно видеть по его изображению и значению его наклона, равному -0.98, это розовый шум.

3. Выводы

В ходе выполнения данной лабораторной работы я изучил несколько видов сигналов, а так же разные способы их генерации. Кроме того я познакомился с понятиями интегрального спектра и спектра мощности, и закрепил знания на практике.