

Лабораторная работа №8
Фильтрация и свёртка

Смирнов Никита

26 апреля 2021 г.

Оглавление

1	Упражнение 8.1	4
2	Упражнение 8.2	5
3	Упражнение 8.3	9
4	Выводы	13

Список иллюстраций

2.1	1	5
2.2	Визуализация FFT	6
2.3	Визуализация гауссовского сигнала	6
2.4	Визуализация окна Гаусса и его FFT	7
2.5	Изменение <code>std</code>	8
3.1	Визуализация окон	10
3.2	Визуализация ДПФ	11
3.3	Визуализация ДПФ	11

Листинги

2.1	Функция <code>plot_gaussian</code>	6
2.2	Изменение <code>std</code>	7
3.1	Создание сигнала	9
3.2	Создание различных окон	9
3.3	Визуализация окон	9
3.4	Визуализация ДПФ	10
3.5	Визуализация ДПФ	11

Глава 1

Упражнение 8.1

В данном упражнении нас просят открыть `chap02.ipynb`, прочитать пояснения и запустить примеры. Поэтому я просто изучил все примеры и комментарии к ним. В итоге, на основе полученных результатов можно сделать вывод, что увеличение `std` приводит к "сплющиванию" гауссовой кривой.

Глава 2

Упражнение 8.2

Во втором пункте нам необходимо оаробывать ДПФ и понять, что происходит при изменении std.

Рассмотрим Гауссовский пример:

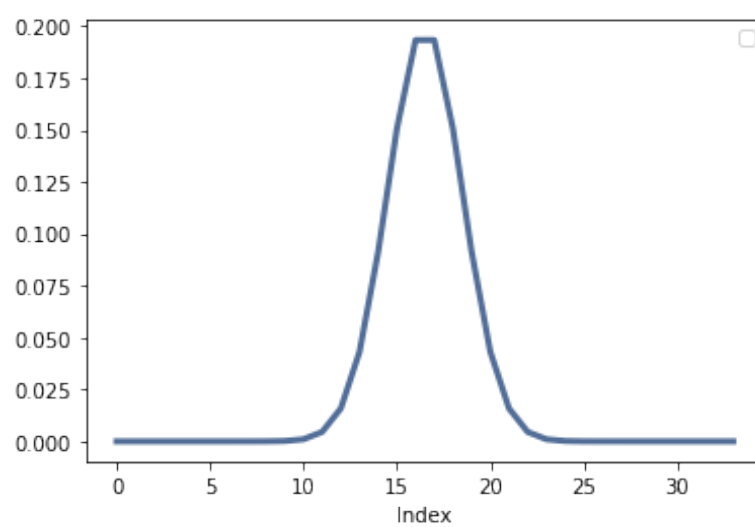


Рис. 2.1: 1

Отобразим FFT:

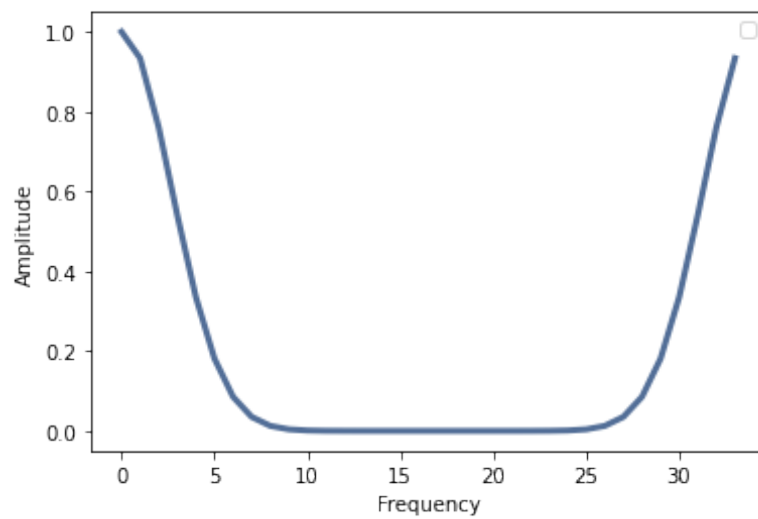


Рис. 2.2: Визуализация FFT

Если мы повернём отрицательные частоты влево, то сможем яснее увидеть, что это гауссово, по крайней мере приблизительно.

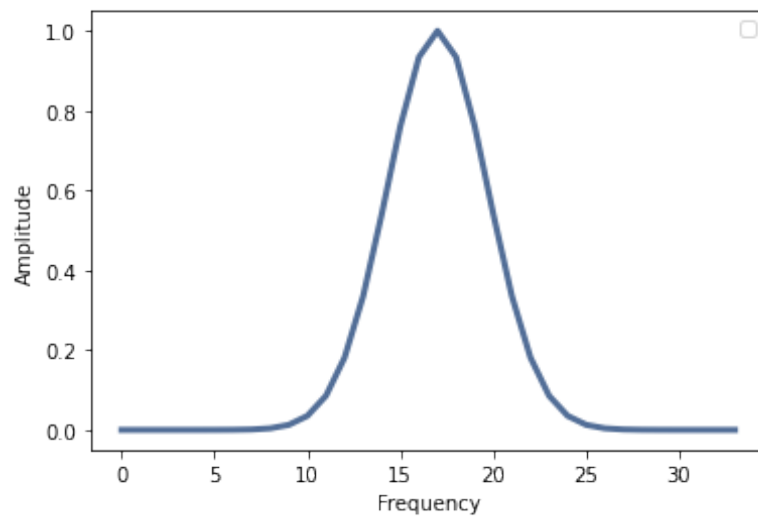


Рис. 2.3: Визуализация гауссовского сигнала

Эта функция отображает окно Гаусса и его FFT друг с другом.

```

1 def plot_gaussian(std):
2     M = 64
3     gaussian = scipy.signal.gaussian(M=M, std=std)

```

```

4     gaussian /= sum(gaussian)
5
6     thinkplot.preplot(num=2, cols=2)
7     thinkplot.plot(gaussian)
8     thinkplot.config(xlabel='Time', legend=False)
9
10    fft_gaussian = np.fft.fft(gaussian)
11    fft_rolled = np.roll(fft_gaussian, M//2)
12
13    thinkplot.subplot(2)
14    thinkplot.plot(abs(fft_rolled))
15    thinkplot.config(xlabel='Frequency')
16
17
18    plot_gaussian(2)

```

Листинг 2.1: Функция `plot_gaussian`

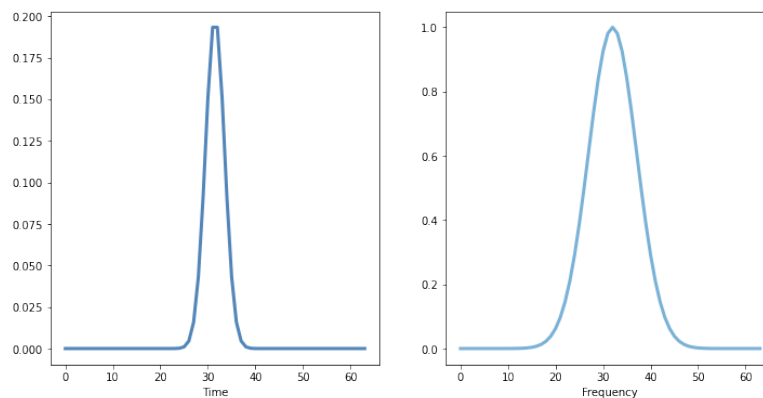


Рис. 2.4: Визуализация окна Гаусса и его FFT

Теперь мы можем сделать манипуляции, которые покажут, что происходит при изменении `std`.

```

1 from ipywidgets import interact, interactive, fixed
2 import ipywidgets as widgets
3
4 slider = widgets.FloatSlider(min=0.1, max=10, value=2)
5 interact(plot_gaussian, std=slider);

```

Листинг 2.2: Изменение `std`

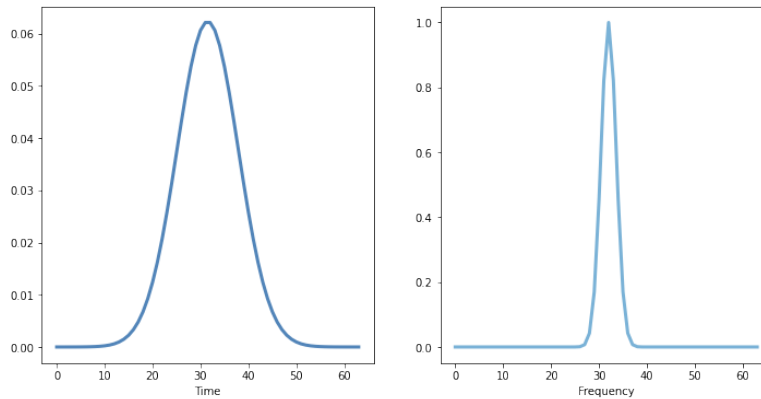


Рис. 2.5: Изменение **std**

По мере увеличения **std** Гауссовский становится шире, а его БПФ сужается.

С точки зрения непрерывной математики, если

$$f(x) = e^{-ax^2}$$

который является гауссовским со средним 0 и стандартным отклонением $1/a$, его преобразование Фурье имеет вид

$$F(k) = \sqrt{\frac{\pi}{a}} e^{-\pi^2 k^2 / a}$$

который является гауссовским со стандартным отклонением a/π^2 . Таким образом, существует обратная зависимость между стандартными отклонениями f и F .

Глава 3

Упражнение 8.3

Тут нам нужно создать окно Хемминга тех размеров, что и Гаусса. Распечатать его ДПФ. Определить какое окно больше для фильтрации НЧ. Создадим 1-секундную волну с частотой дискретизации 44 кГц.

```
1 signal = thinkdsp.SquareSignal(freq=440)
2 wave = signal.make_wave(duration=1.0, framerate=44000)
```

Листинг 3.1: Создание сигнала

Затем создадим несколько окон. Выберем стандартное отклонение окна Гаусса, чтобы сделать его похожим на другие.

```
1 M = 17
2 std = 2.5
3
4 gaussian = scipy.signal.gaussian(M=M, std=std)
5 bartlett = np.bartlett(M)
6 blackman = np.blackman(M)
7 hamming = np.hamming(M)
8 hanning = np.hanning(M)
9
10 windows = [gaussian, blackman, hamming, hanning]
11 names = ['gaussian', 'blackman', 'hamming', 'hanning']
12
13 for window in windows:
14     window /= sum(window)
```

Листинг 3.2: Создание различных окон

Теперь посмотрим, как выглядят эти окна.

```
1 thinkplot.preplot(4)
```

```

2 for window, name in zip(windows, names):
3     thinkplot.plot(window, label=name)
4
5 thinkplot.config(xlabel='Index', legend=True, loc='center bottom')

```

Листинг 3.3: Визуализация окон

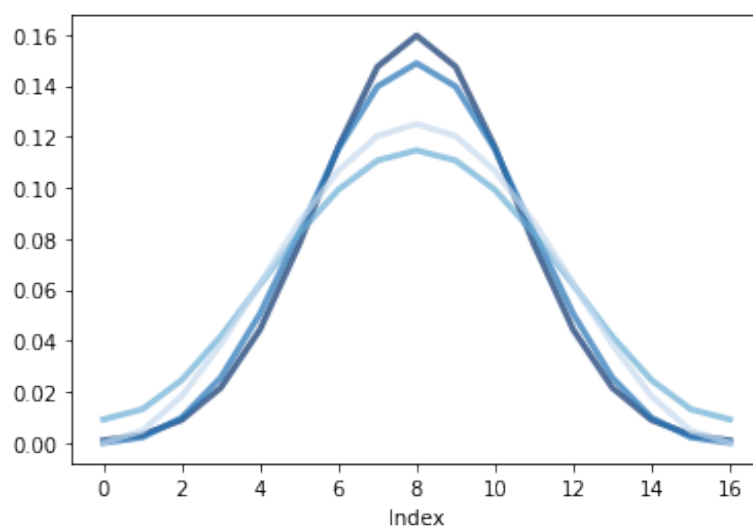


Рис. 3.1: Визуализация окон

Они выглядят довольно похоже. Посмотрим, как выглядят их ДПФ:

```

1 plot_window_dfts(windows, names)
2 thinkplot.config(xlabel='Frequency (Hz)', loc='upper right')

```

Листинг 3.4: Визуализация ДПФ

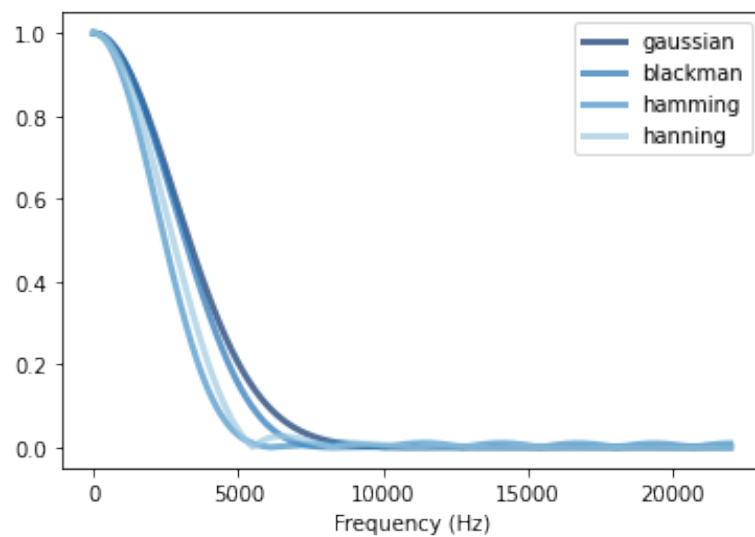


Рис. 3.2: Визуализация ДПФ

Стоит отметить, что Гауссово падает быстрее всех, Блэкман - самый медленный.

```

1 plot_window_dfts(windows, names)
2 thinkplot.config(xlabel='Frequency (Hz)', yscale='log',
3                 loc='lower left')
```

Листинг 3.5: Визуализация ДПФ

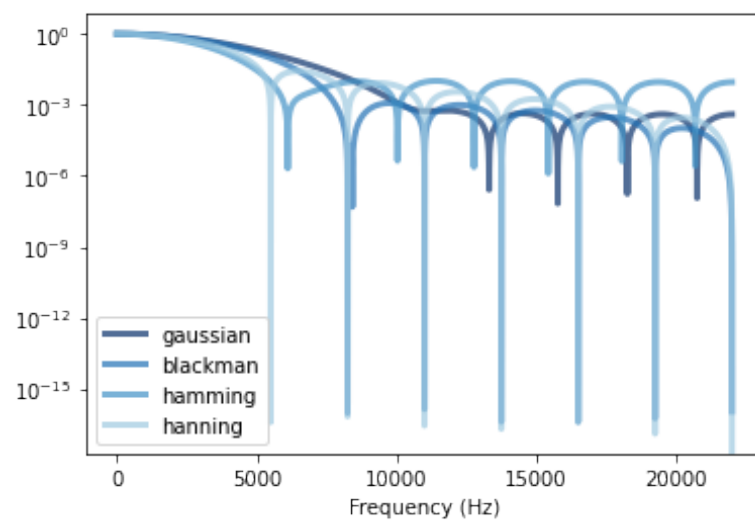


Рис. 3.3: Визуализация ДПФ

В логарифмической шкале мы видим, что сначала значения Хэмминга и Хеннинга падают быстрее, чем два других. И окна Хэмминга и Гаусса, кажется, имеют самые стойкие боковые лепестки. Окно Ханнинга, кажется, имеет наилучшее сочетание быстрого спада и минимальных боковых лепестков.

Глава 4

Выводы

Во время выполнения лабораторной работы получены навыки работы с концепцией свёртки и теоремой свёртки, а также научился применять эти знания на практике.