

Лабораторная работа 8

Чернышев Ярослав

1 июня 2021 г.

Оглавление

1	Задание 8.1	2
2	Задание 8.2	3
3	Задание 8.3	7

Глава 1

Задание 8.1

В данном задании требуется ознакомиться и запустить примеры из `chap08.ipynb`. Как показало тестирование, увеличение параметра `std` при неизменном значении `m` приводит к появлению боковых лепестков, что говорит об ухудшении подавления высоких частот.

Глава 2

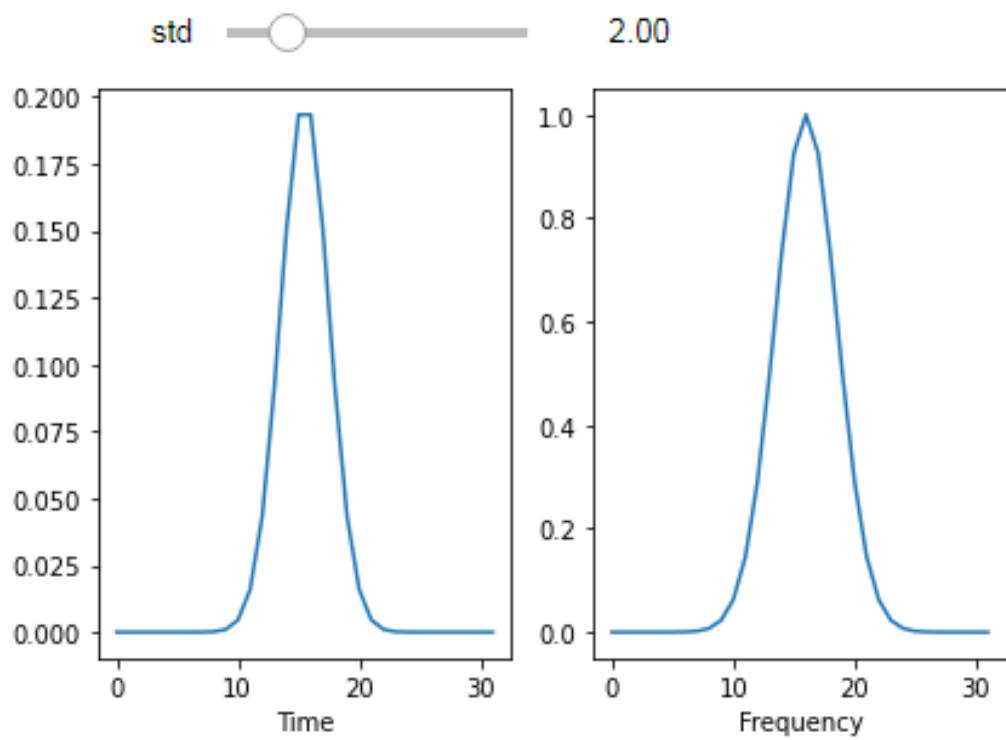
Задание 8.2

В этом задании требуется исследовать утверждение, что преобразование Фурье, примененное к Гауссовой кривой - тоже Гауссова кривая.

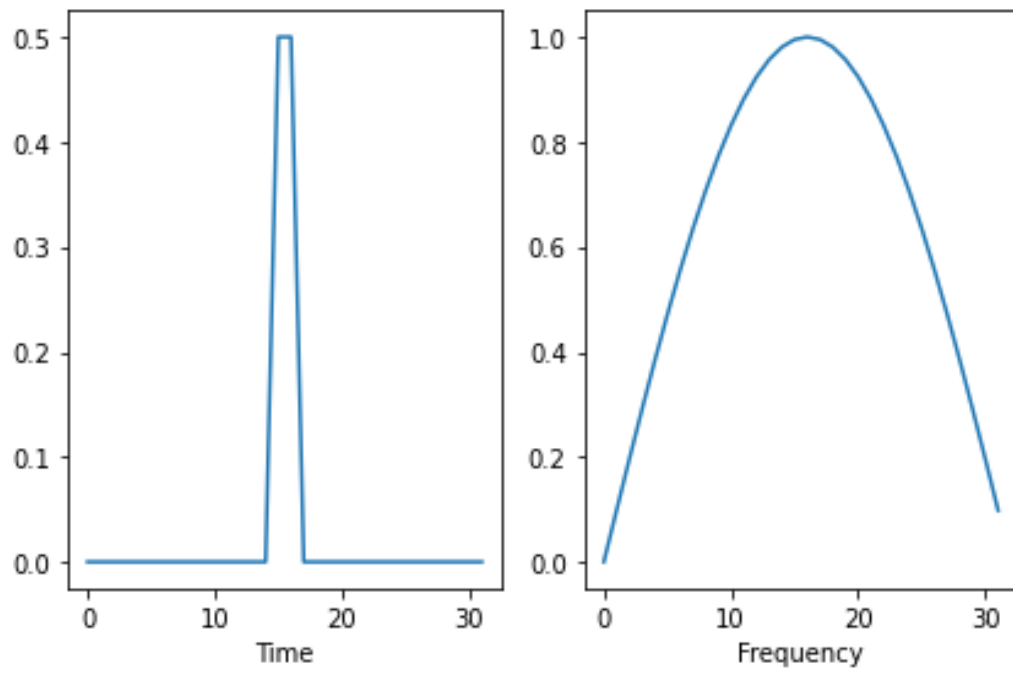
Для решения поставленной задачи напомним функцию и виджет, позволяющие изобразить рядом Гауссово окно и быстрое преобразование Фурье, а также динамично изменять `std`.

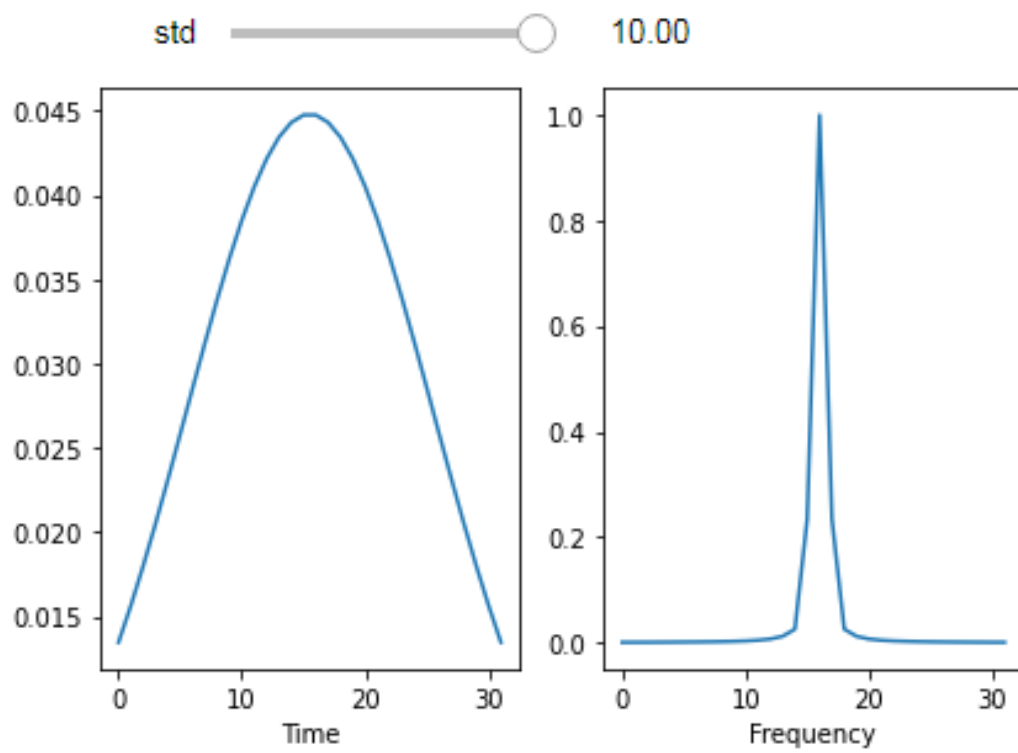
```
1 def plot_gaussian(std):
2     M = 32
3     gaussian = scipy.signal.gaussian(M=M, std=std)
4     gaussian /= sum(gaussian)
5     plt.subplot(1, 2, 1)
6     plt.plot(gaussian)
7     decorate(xlabel='Time')
8     fft_gaussian = np.fft.fft(gaussian)
9     fft_rolled = np.roll(fft_gaussian, M//2)
10    plt.subplot(1, 2, 2)
11    plt.plot(np.abs(fft_rolled))
12    decorate(xlabel='Frequency')
13    plt.show()
14
15 slider = widgets.FloatSlider(min=0.1, max=10, value=2)
16 interact(plot_gaussian, std=slider);
```

Результат его выполнения:



std





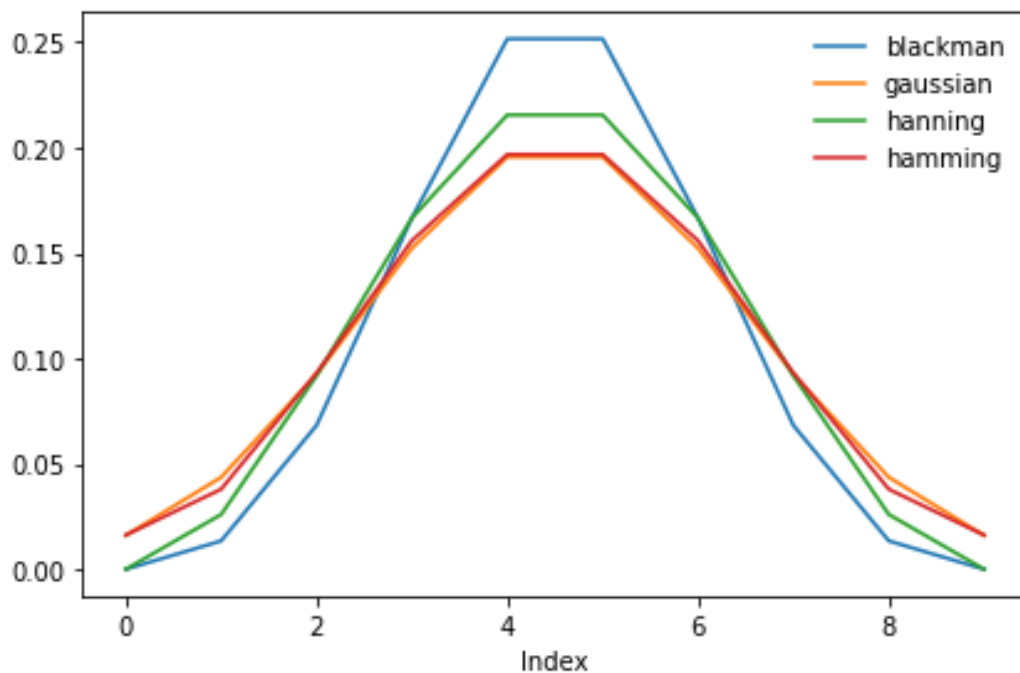
Глава 3

Задание 8.3

В данном задании требуется провести сравнение разных видов окон, речь о которых шла в предыдущей главе. Создадим их:

```
1 M = 10
2 std = 2.0
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 = [blackman, gaussian, hanning, hamming]
11 names = ['blackman', 'gaussian', 'hanning', 'hamming']
12
13 for window in windows:
14     window /= sum(window)
15
16 for window, name in zip(windows, names):
17     plt.plot(window, label=name)
18
19 decorate(xlabel='Index')
```

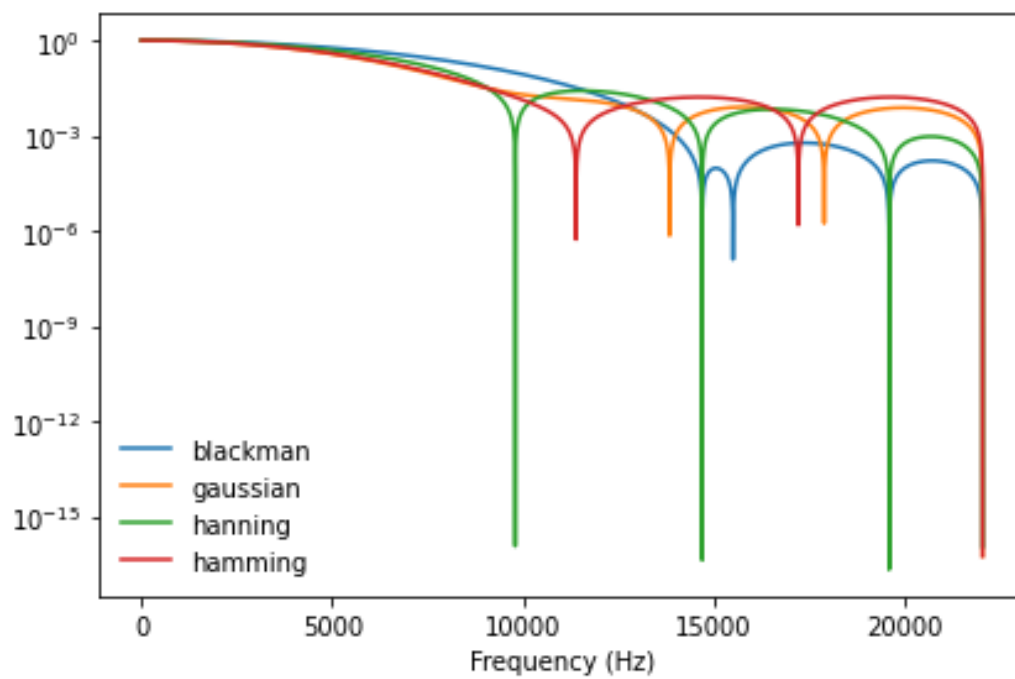
В результате работы кода выше можно увидеть их вид:



Далее, требуется вывести их дискретные преобразования Фурье в логарифмическом масштабе:

```

1 def zero_pad(array, n):
2     res = np.zeros(n)
3     res[:len(array)] = array
4     return res
5
6 def plot_window_dfts(windows, names):
7     for window, name in zip(windows, names):
8         padded = zero_pad(window, len(wave))
9         dft_window = np.fft.rfft(padded)
10        plt.plot(abs(dft_window), label=name)
11
12 plot_window_dfts(windows, names)
13 decorate(xlabel='Frequency (Hz)', ylabel='log')
```



На основе этого рисунка можно сделать вывод, что окна Хэннинга и Хэмминга уменьшаются быстрее остальных, причем картина оставалась примерно такой же и при изменении исходных параметров (M и std).