

Санкт-Петербургский государственный политехнический  
университет Петра Великого

**Высшая школа интеллектуальных систем и  
суперкомпьютерных технологий**

Лабораторная работа №8

## Фильтрация и свертка

Выполнил студент 3-го курса  
группа 3530901/80201  
Матвеец Андрей Вадимович

Преподаватель:  
Богач Наталья Владимировна

Санкт-Петербург

2021

# Содержание

1	Часть №1: chap08.ipynb	5
2	Часть №2: Преобразование Фурье гауссовой кривой	10
3	Часть №3: NumPy	14
4	Выводы	16

## Список иллюстраций

1	Std = 0,5 . . . . .	6
2	Std = 2 . . . . .	7
3	Std = 5 . . . . .	8
4	Std = 20 . . . . .	9
5	Std = 0,5 . . . . .	11
6	Std = 2 . . . . .	11
7	Std = 5 . . . . .	12
8	Std = 10 . . . . .	12
9	Результат работы plot-window . . . . .	15

## Листинги

1	Функция <code>plot-filter</code> . . . . .	5
2	Интерактивный виджет для функции <code>plot-filter</code> . . . . .	5
3	Функция <code>plot-gaussian</code> . . . . .	10
4	Интерактивный виджет для функции <code>plot-gaussian</code> . . . . .	10
5	Функция <code>plot-window</code> . . . . .	14

# 1 Часть №1: chap08.ipynb

В первой части лабораторной работы нам необходимо запустить весь код из блокнота `chap08.ipynb` и проверить, что будет при увеличении ширины гауссова окна `std`, не увеличивая число элементов в окне `m`.

Для этого возьмем функцию `plot-filter`:

```
1 def plot_filter(M=11, std=2):
2     signal = SquareSignal(freq=440)
3     wave = signal.make_wave(duration=1, framerate=44100)
4     spectrum = wave.make_spectrum()
5
6     gaussian = scipy.signal.gaussian(M=M, std=std)
7     gaussian /= sum(gaussian)
8     high = gaussian.max()
9     thinkplot.preplot(cols=2)
10    thinkplot.plot(gaussian)
11    thinkplot.config(xlabel='Index', ylabel='Window',
12                    xlim=[0, len(gaussian)-1], ylim=[0, 1.1*high])
13
14    ys = np.convolve(wave.ys, gaussian, mode='same')
15    smooth = Wave(ys, framerate=wave.framerate)
16    spectrum2 = smooth.make_spectrum()
17
18    amps = spectrum.amps
19    amps2 = spectrum2.amps
20    ratio = amps2 / amps
21    ratio[amps<560] = 0
22
23    padded = zero_pad(gaussian, len(wave))
24    dft_gaussian = np.fft.rfft(padded)
25
26    thinkplot.subplot(2)
27    thinkplot.plot(abs(dft_gaussian), color='0.7', label='Gaussian filter')
28    thinkplot.plot(ratio, label='amplitude ratio')
29
30    thinkplot.show(xlabel='Frequency (Hz)', ylabel='Amplitude ratio')
```

Листинг 1: Функция `plot-filter`

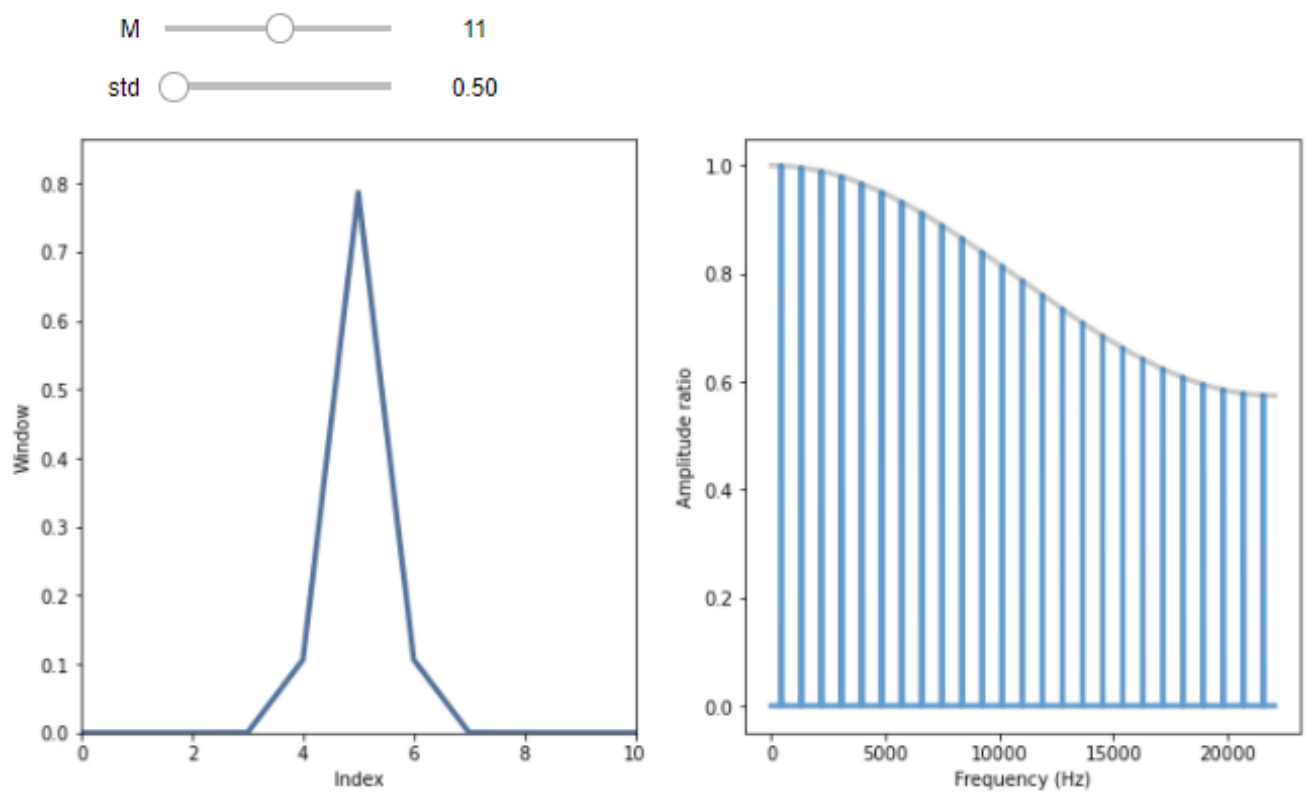
Напишем для нашей функции интерактивный виджет:

```
1 slider = widgets.IntSlider(min=2, max=100, value=11)
2 slider2 = widgets.FloatSlider(min=0, max=20, value=2)
3 interact(plot_filter, M=slider, std=slider2);
```

Листинг 2: Интерактивный виджет для функции `plot-filter`

После этого начнем менять значения `std`, не трогая `m`:

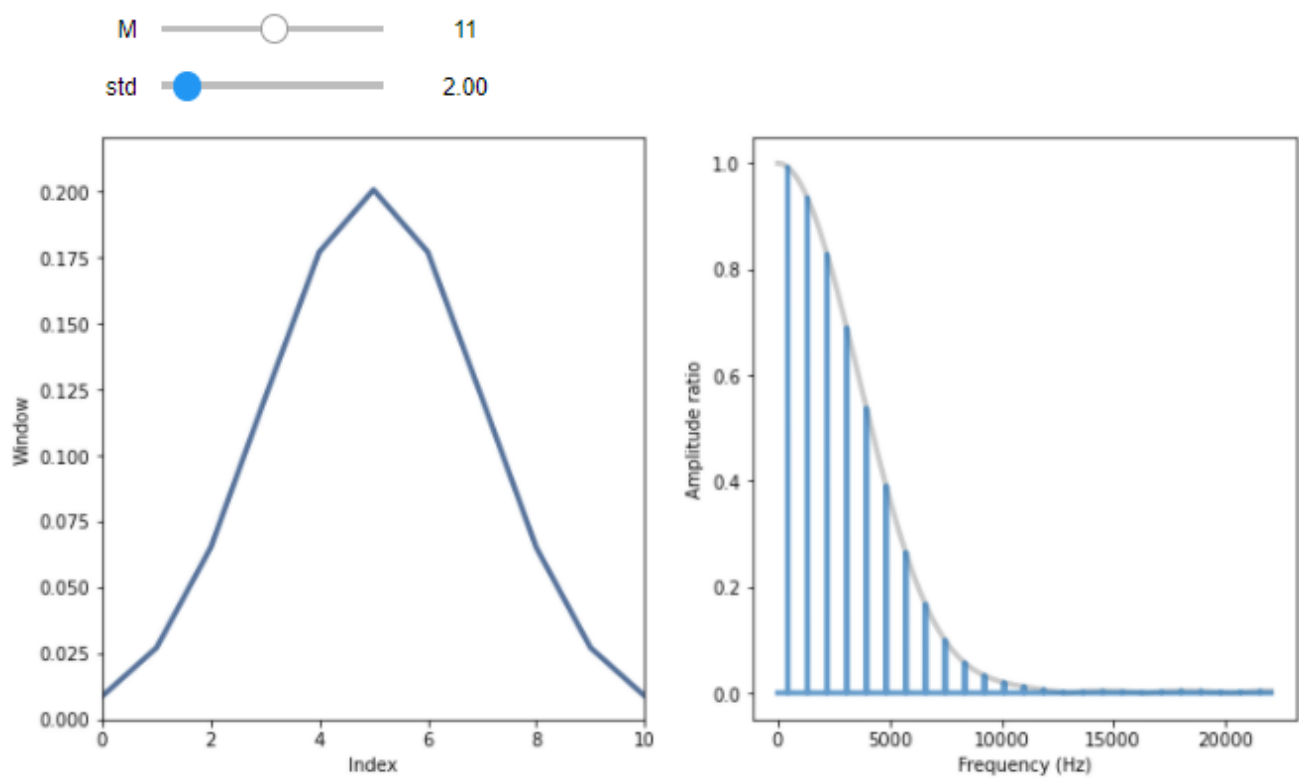
Сначала поставим `std = 0,5` и посмотрим на полученные результаты:



<Figure size 576x432 with 0 Axes>

Рис. 1: Std = 0,5

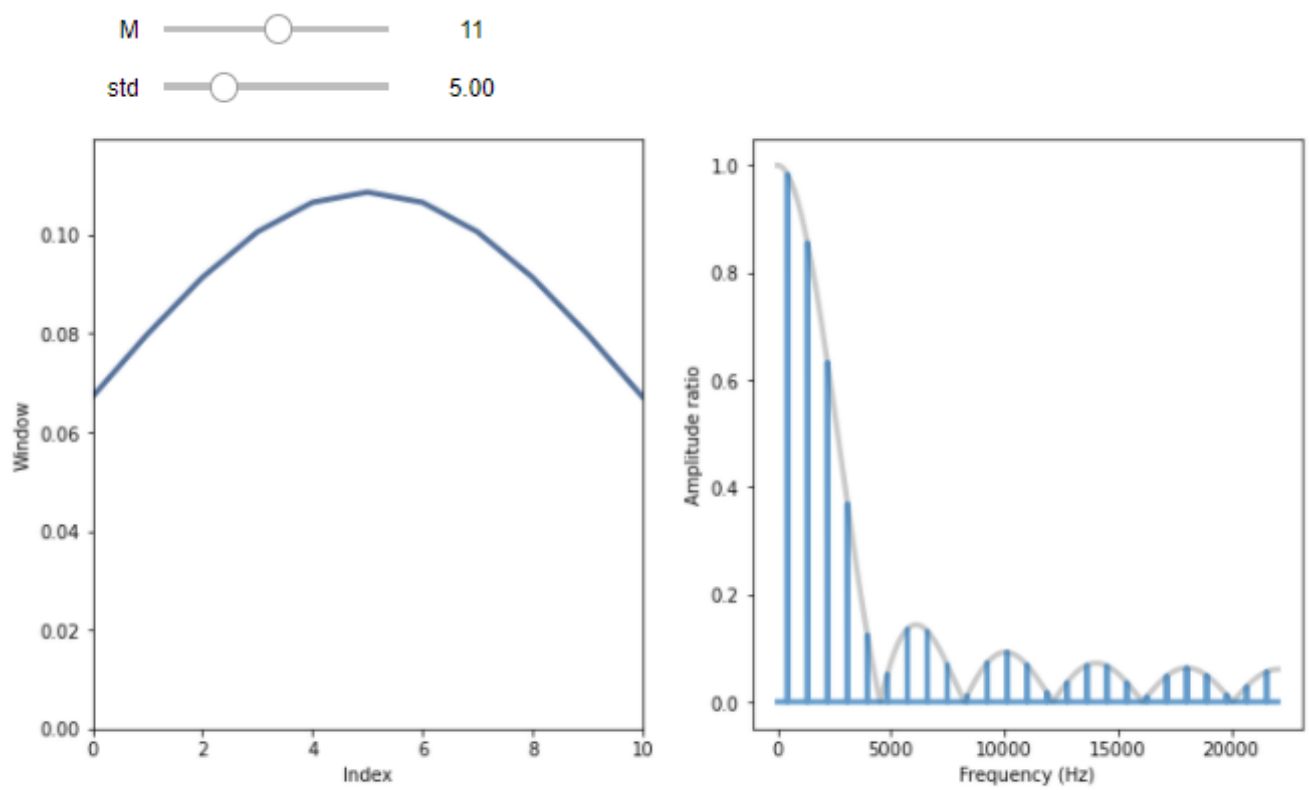
Теперь поставим `std = 2`:



<Figure size 576x432 with 0 Axes>

Рис. 2: Std = 2

После этого поставим `std = 5`:

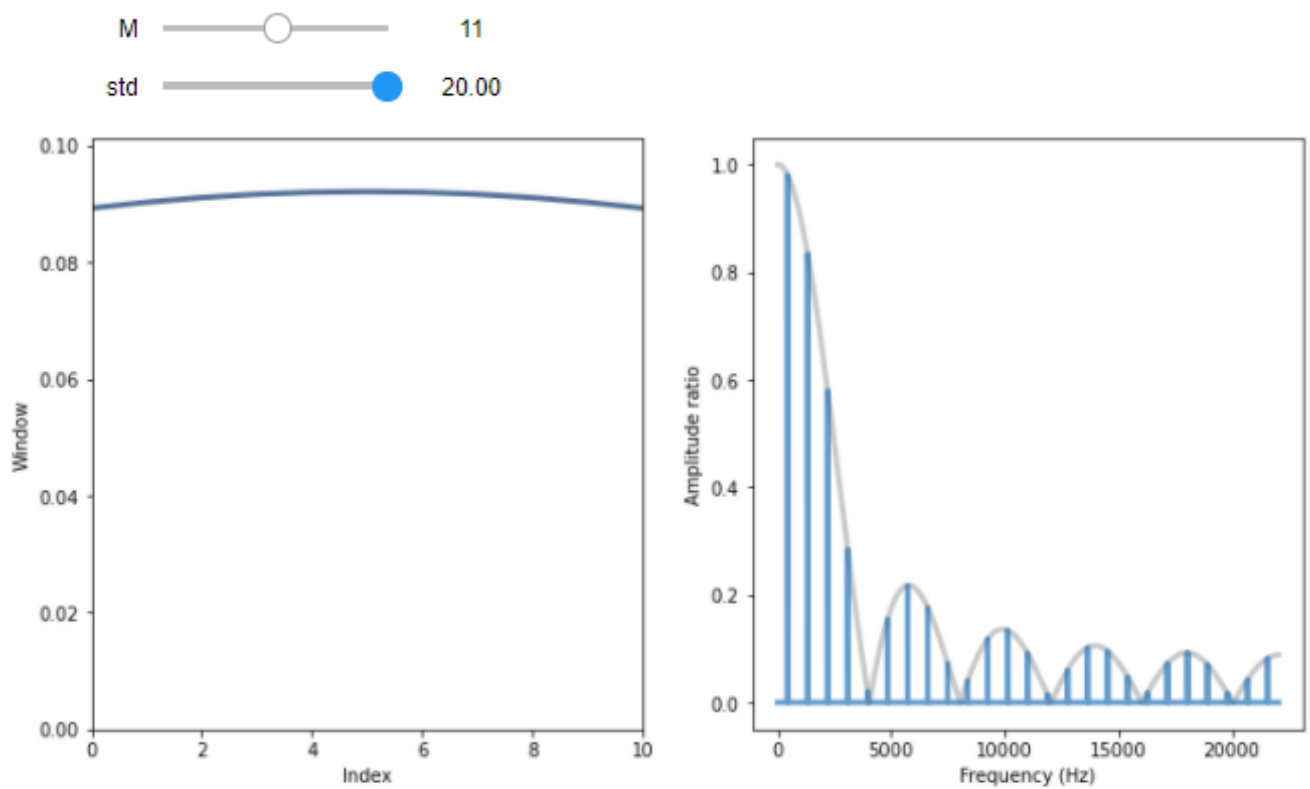


<Figure size 576x432 with 0 Axes>

Рис. 3: Std = 5

И, наконец, ради интереса поставим  $\text{std} = 20$ :





<Figure size 576x432 with 0 Axes>

Рис. 4: Std = 20

В итоге, на основе полученных результатов можно сделать вывод, что увеличение **std** приводит к "сплющиванию" гауссовой кривой.

## 2 Часть №2: Преобразование Фурье гауссовой кривой

Во втором пункте лабораторной работы нам необходимо попробовать ДПФ на нескольких примерах и понять, что происходит при изменении `std`.

Для этого создадим функцию `plot-gaussian`, которая будет отображать окно Гаусса и для него же БПФ:

```
1 def plot_gaussian(std):
2     gaussian = scipy.signal.gaussian(M=32, std=std)
3     gaussian /= sum(gaussian)
4
5     thinkplot.preplot(num=2, cols=2)
6     thinkplot.plot(gaussian)
7     thinkplot.config(xlabel='Time', legend=False)
8
9     fft_gaussian = np.fft.fft(gaussian)
10    fft_rolled = np.roll(fft_gaussian, 32//2)
11
12    thinkplot.subplot(2)
13    thinkplot.plot(abs(fft_rolled))
14    thinkplot.config(xlabel='Frequency')
```

Листинг 3: Функция `plot-gaussian`

Также напишем интерактивный виджет для взаимодействия с данной функцией:

```
1 slider = widgets.FloatSlider(min=0.1, max=10, value=2)
2 interact(plot_gaussian, std=slider);
```

Листинг 4: Интерактивный виджет для функции `plot-gaussian`

Теперь с помощью данного виджета посмотрим, как будут меняться окно Гаусса и БПФ при изменении `std`:

Сначала сделаем `std = 0,5`:

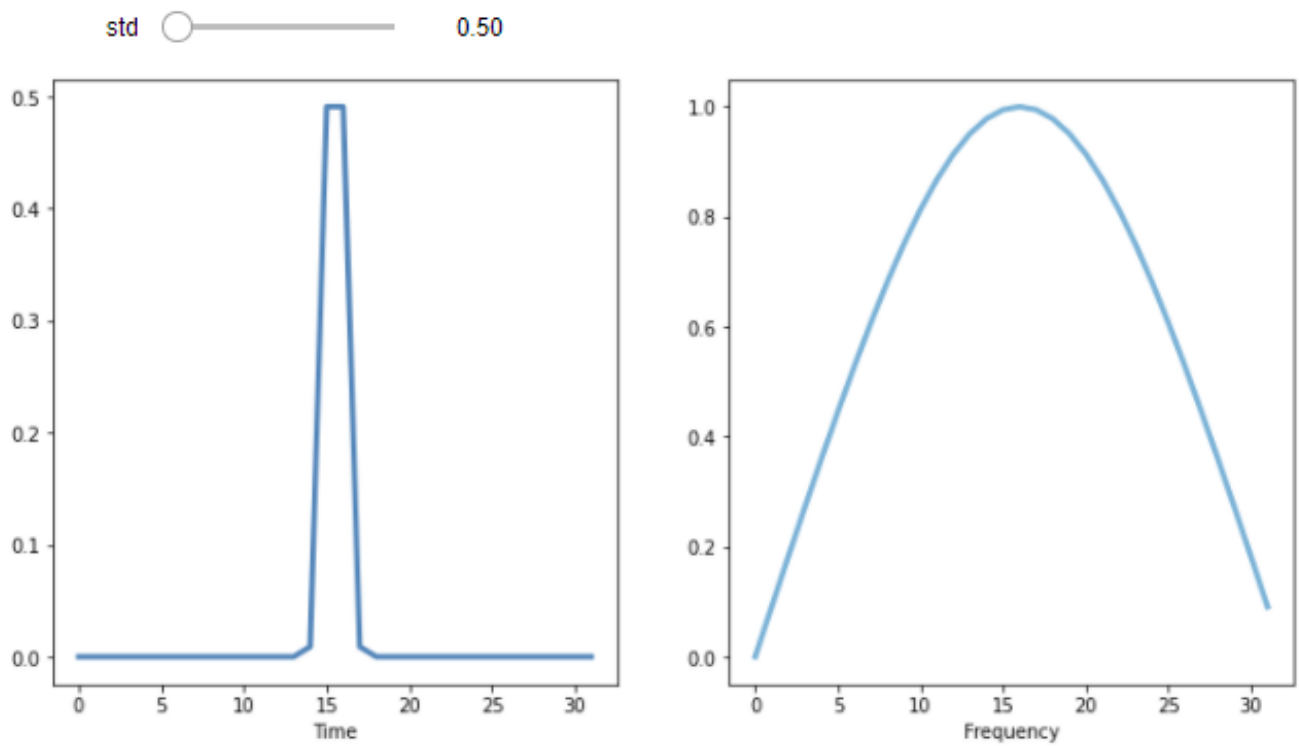


Рис. 5: Std = 0,5

После чего сделаем `std = 2`:

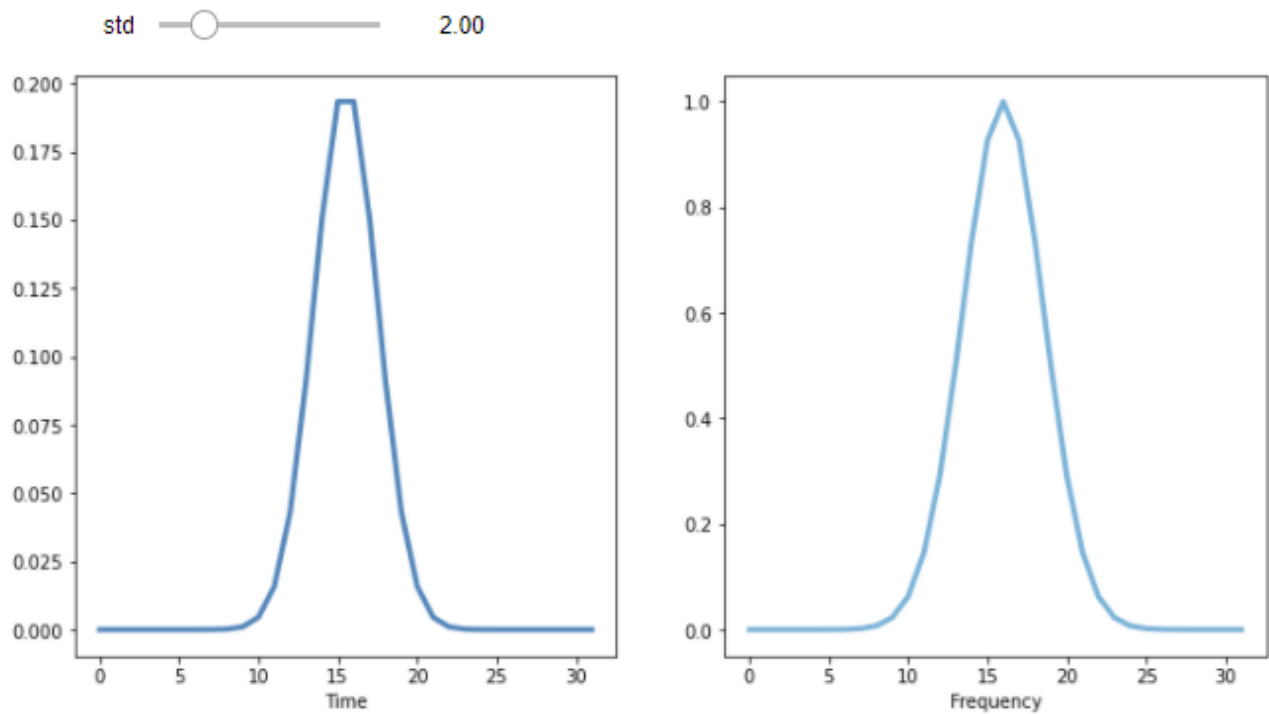


Рис. 6: Std = 2

Затем сделаем `std = 5`:

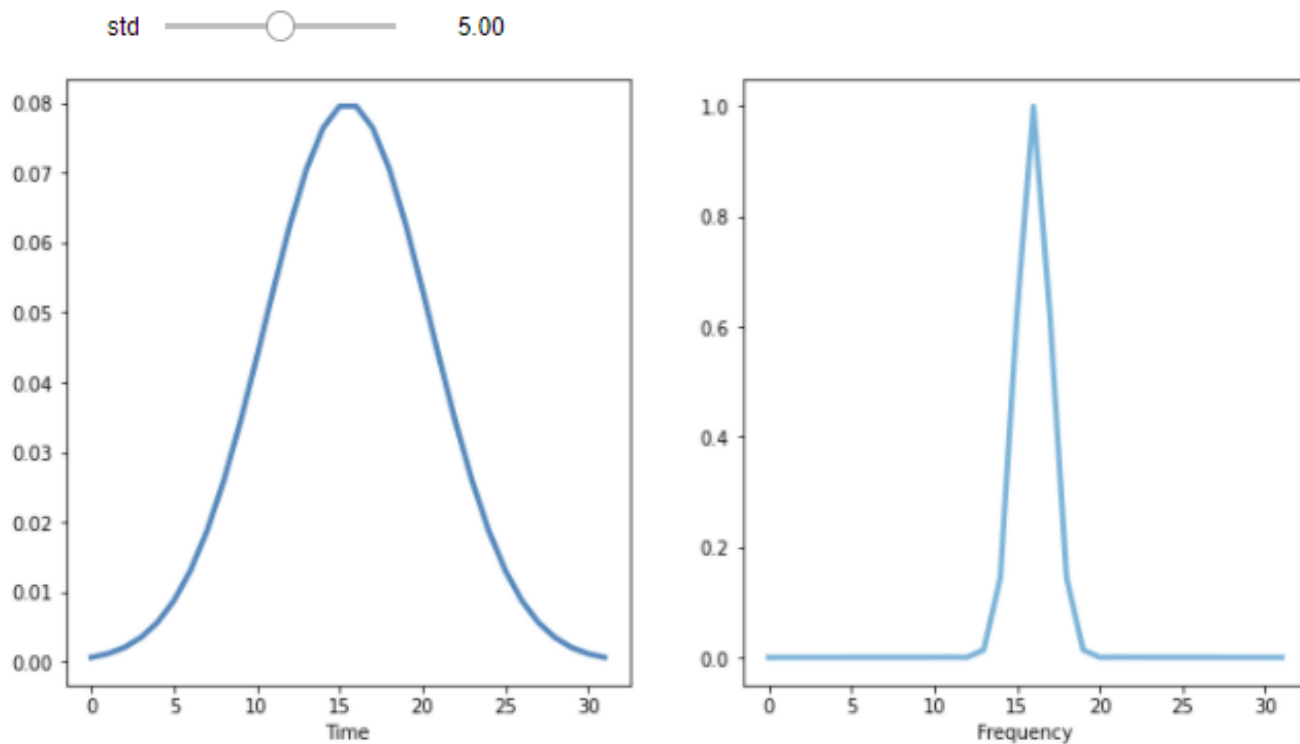


Рис. 7: Std = 5

И, наконец, сделаем максимальное  $\text{std} = 10$ :

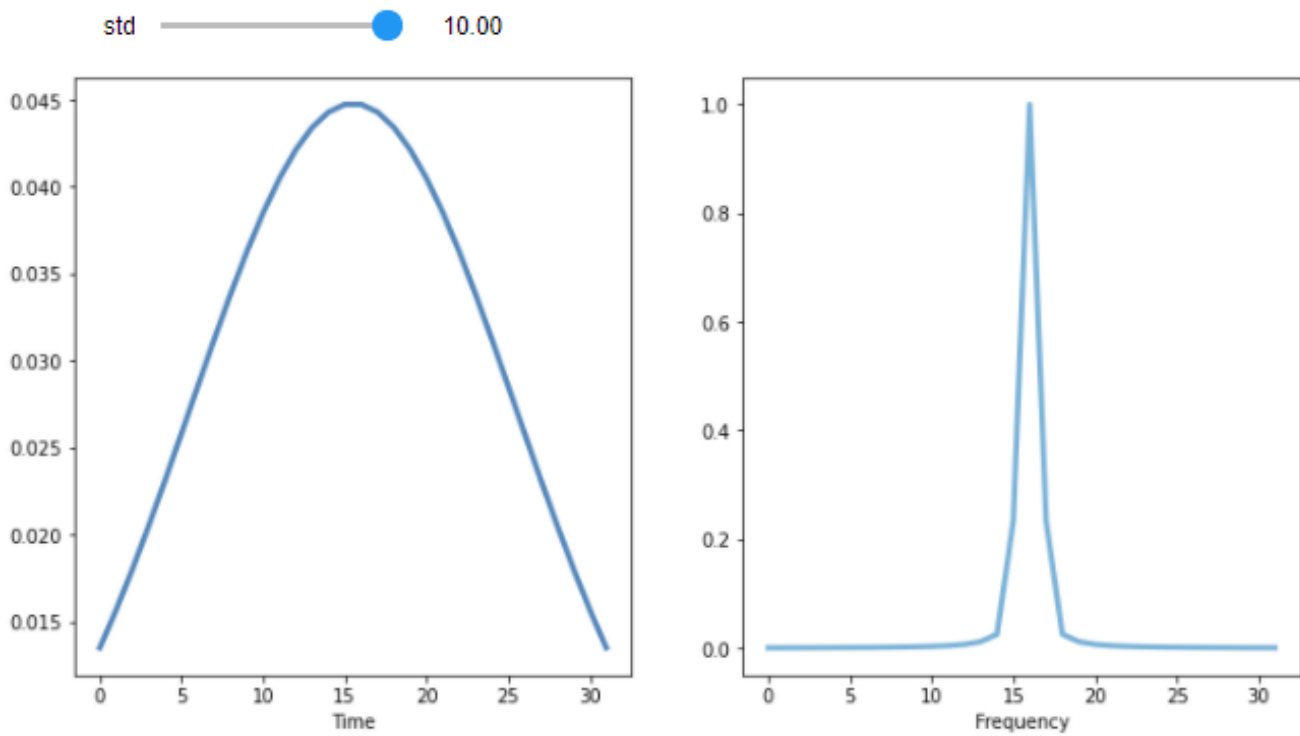


Рис. 8: Std = 10

По итогу, при просмотре полученных результатов становится понятно, что

при увеличении ширины сигнала, идет уменьшение ширины преобразования Фурье. Это работает так же и наоборот.

### 3 Часть №3: NumPy

В третьем пункте лабораторной работы нам необходимо в дополнение к Гауссову окну, использованному ранее создать окно Хэмминга тех же размеров. Также нужно дополнить окно нулями и напечатать его ДПФ. Определить, какое окно больше подходит для фильтра НЧ.

Для решения данной задачи напомним функцию `plot-window`:

```
1 def plot_window(ax, window_fun, M=30):
2     signal = SquareSignal(freq=440)
3     wave = signal.make_wave(duration=1.0, framerate=44100)
4
5     window = window_fun(M)
6     window /= sum(window)
7
8     padded = zero_pad(window, len(wave))
9     fft = np.fft.rfft(padded)
10
11     ax[0].plot(window, label=window_fun.__name__)
12     ax[1].plot(abs(fft)[:8000], label=window_fun.__name__)
13     plt.legend()
14
15 _, ax = plt.subplots(1, 2)
16 for w in [np.blackman, np.bartlett, np.hamming, np.hanning]:
17     plot_window(ax, w)
```

Листинг 5: Функция `plot-window`

Запустим данную функцию и посмотрим на полученный результат:

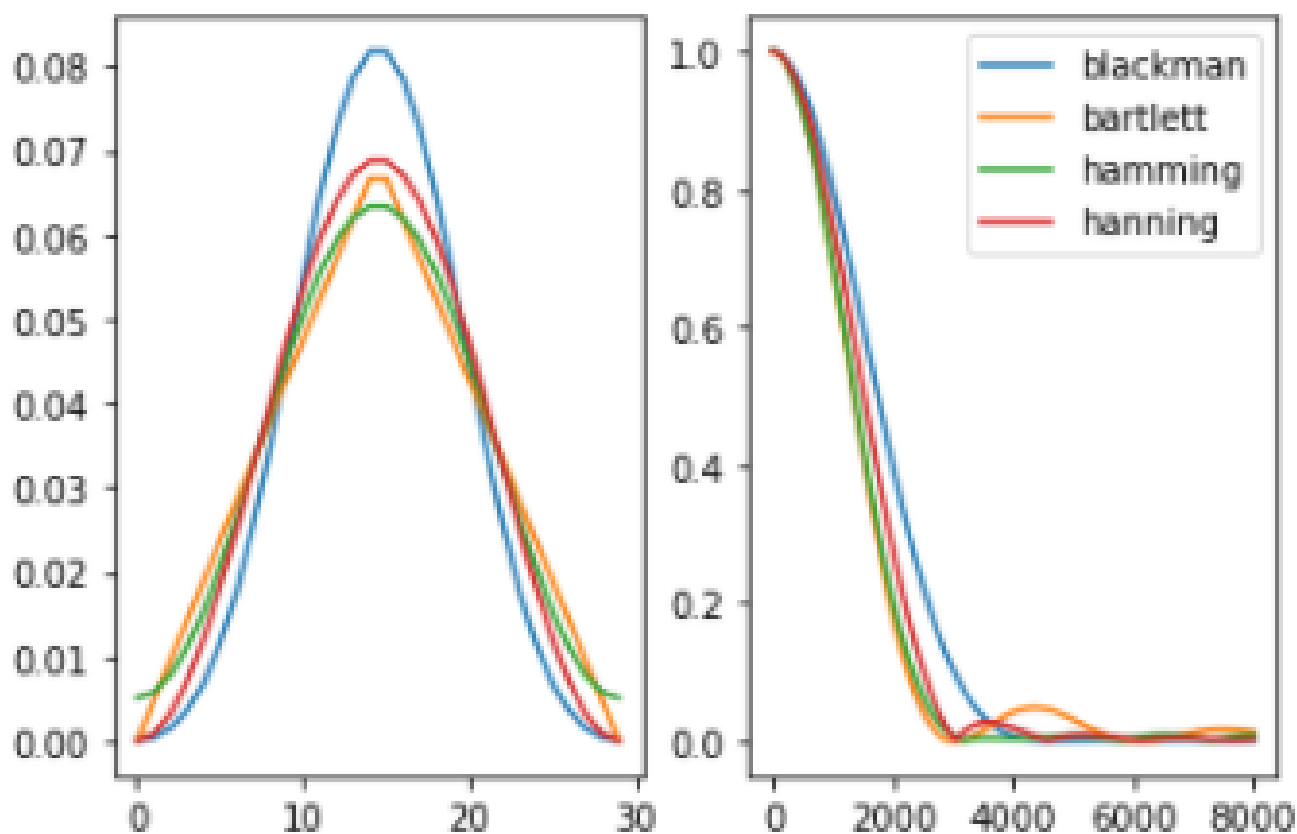


Рис. 9: Результат работы `plot-window`

Как можно увидеть из результата работы функции, для фильтрации НЧ лучше всего использовать окно Хэмминга, т.к. оно дает меньше "выпуклостей".

## 4 Выводы

В результате выполнения лабораторной работы мы разобрались с тем, как ширина гауссова окна `std` влияет на гауссову кривую. Кроме того мы написали функции `plot-gaussian` и `plot-window` для отображения окна Гаусса с выводом БПФ для него и для отображения окна Хэмминга соответственно. В результате было установлено, что для фильтрации НЧ лучше всего использовать окно Хэмминга.