Санкт-Петербургский государственный политехнический
университет Петра Великого

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

Лабораторная работа

Фильтрация и свертка

Работу выполнил студент
3-го курса, группа 3530901/80201
Солянкин Илья Андреевич

Преподаватель:
Богач Наталья Владимировна

Санкт-Петербург 2021

Содержание

Список иллюстраций

Листинги

1 Часть №1: chap08.ipynb

В первой части восьмой 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 возмем функцию plot-filter:

```
def plot_filter(M=11, std=2):
    signal = SquareSignal(freq=440)
    wave = signal.make_wave(duration=1, framerate=44100)
    spectrum = wave.make_spectrum()

    gaussian = scipy.signal.gaussian(M=M, std=std)
    gaussian /= sum(gaussian)
    high = gaussian.max()
    thinkplot.preplot(cols=2)
    thinkplot.plot(gaussian)
    thinkplot.config(xlabel='Index', ylabel='Window',
    xlim=[0, len(gaussian)-1], ylim=[0, 1.1*high])

    ys = np.convolve(wave.ys, gaussian, mode='same')
    smooth =  Wave(ys, framerate=wave.framerate)
    spectrum2 = smooth.make_spectrum()

    amps = spectrum.amps
    amps2 = spectrum2.amps
    ratio = amps2 / amps
    ratio[amps<560] = 0

    padded = zero_pad(gaussian, len(wave))
    dft_gaussian = np.fft.rfft(padded)

    thinkplot.subplot(2)
    thinkplot.plot(abs(dft_gaussian), color='0.7', label='Gaussian filter')
    thinkplot.plot(ratio, label='amplitude ratio')

    thinkplot.show(xlabel='Frequency (Hz)', ylabel='Amplitude ratio')
```

Листинг 1: Функция plot-filter

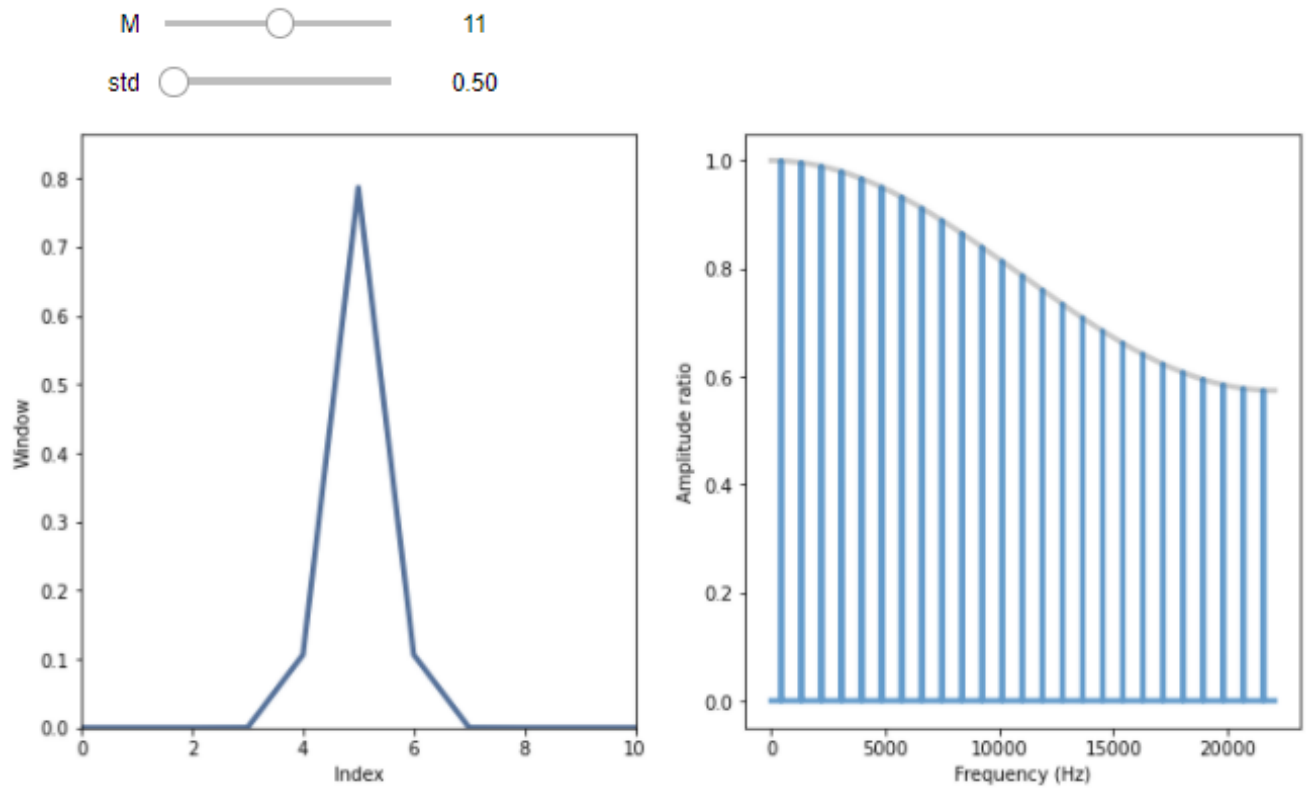
Напишем для нашей функции интерактивный виджет:

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

Листинг 2: Интерактивный виджет для функции plot-filter

После этого начнем менять значения std, не трогая m:

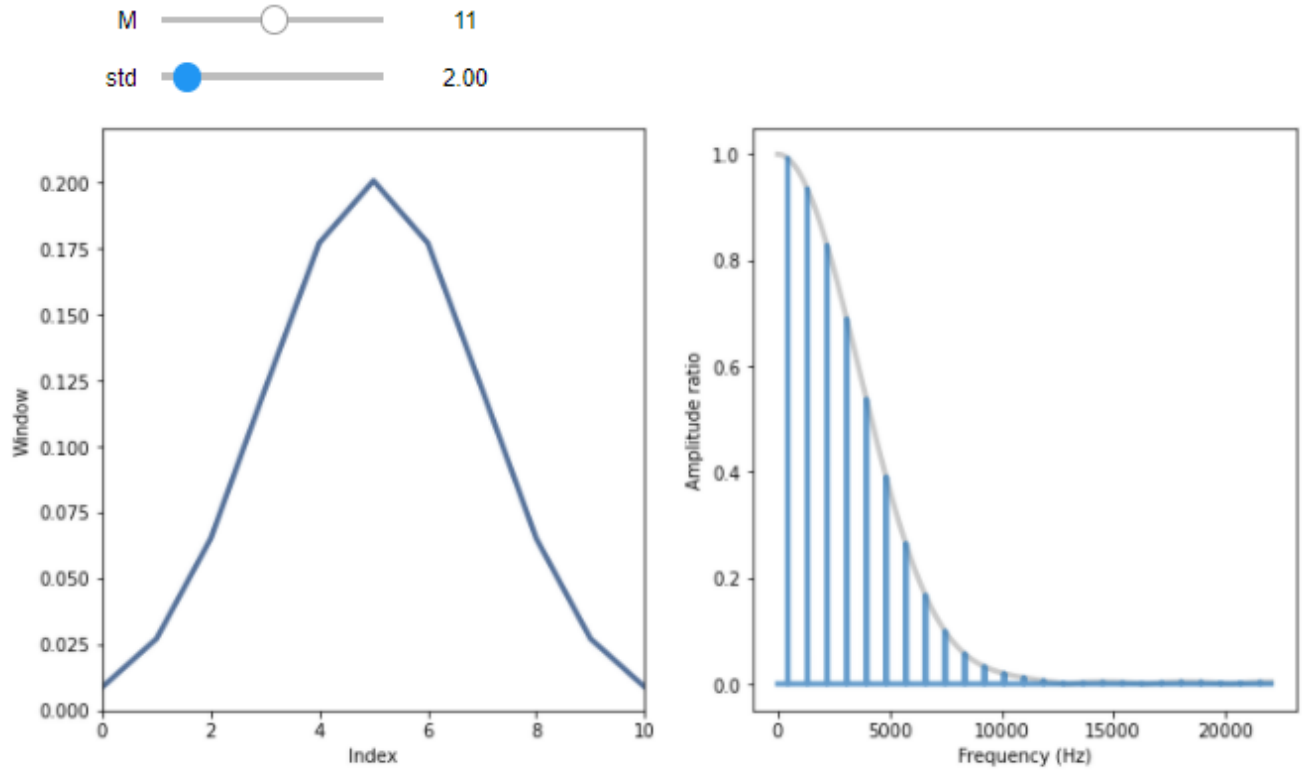
Сначала поставим std = 0,5 и посмотрим на полученные результаты:




```
<Figure size 576x432 with 0 Axes>
```

Рис. 1: Std = 0,5

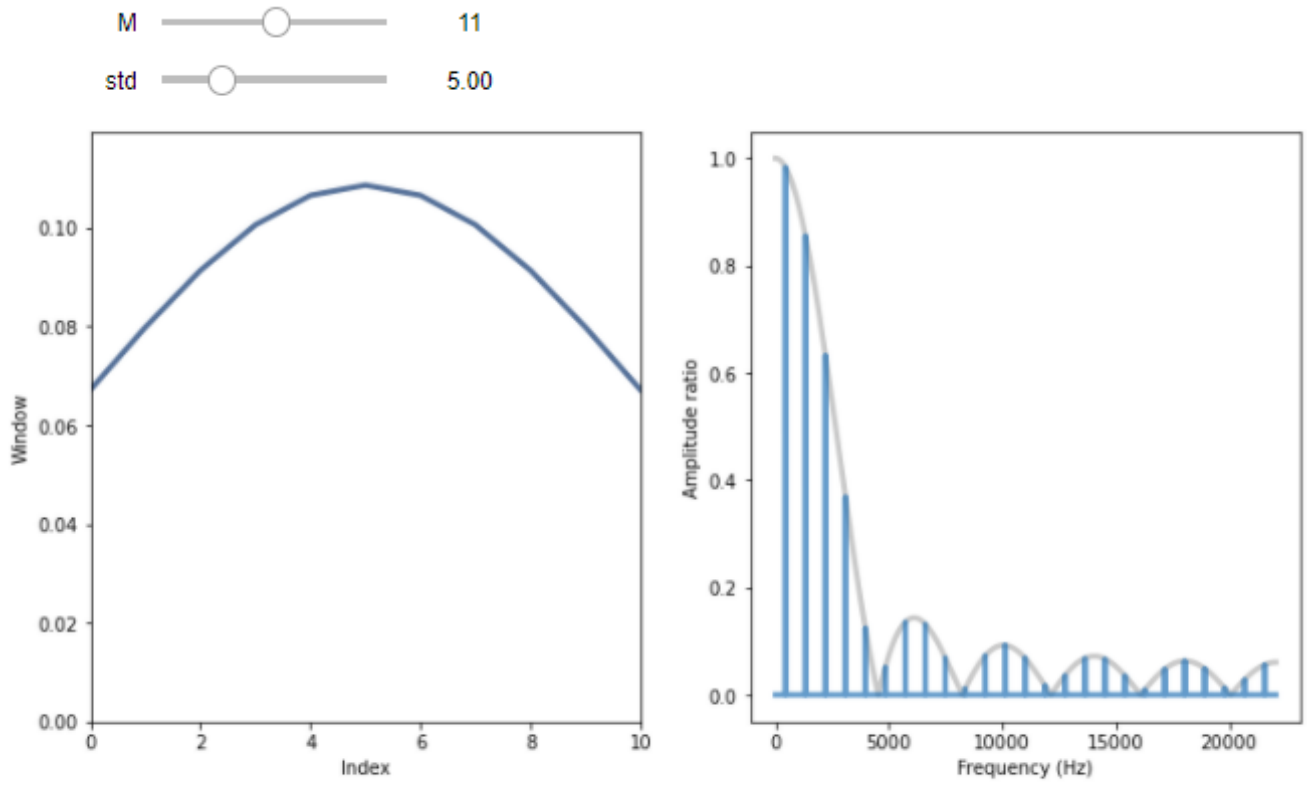
Теперь поставим std = 2:




```
<Figure size 576x432 with 0 Axes>
```

Рис. 2: Std = 2

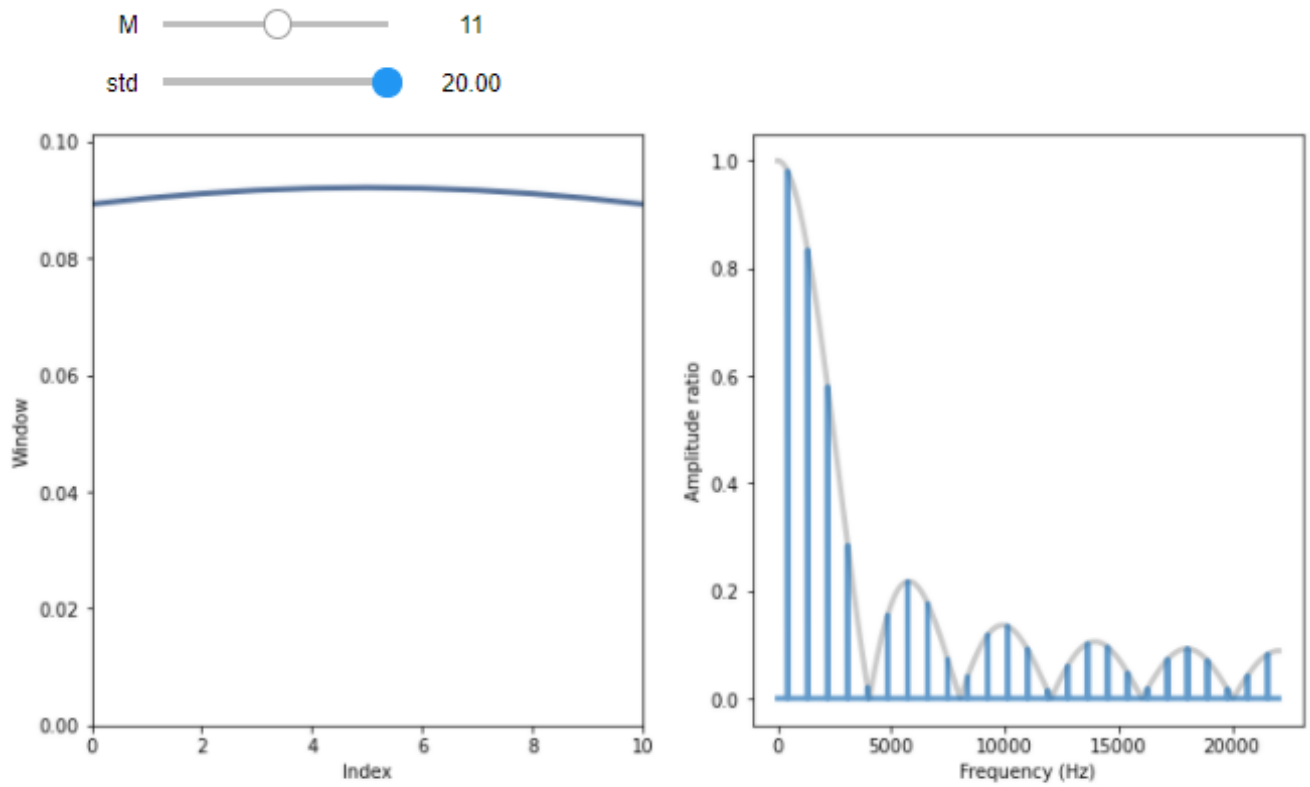
После этого поставим std = 5:




```
<Figure size 576x432 with 0 Axes>
```

Рис. 3: `Std` = 5

И, наконец, ради интереса поставим `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 восьмой лабораторной работы нам необходимо попробовать ДПФ на несколких примерах и понять, что происходит при изменении `std`.

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

```
    def plot_gaussian(std):
        gaussian = scipy.signal.gaussian(M=32, std=std)
        gaussian /= sum(gaussian)

        thinkplot.preplot(num=2, cols=2)
        thinkplot.plot(gaussian)
        thinkplot.config(xlabel='Time', legend=False)

        fft_gaussian = np.fft.fft(gaussian)
        fft_rolled = np.roll(fft_gaussian, 32//2)

        thinkplot.subplot(2)
        thinkplot.plot(abs(fft_rolled))
        thinkplot.config(xlabel='Frequency')
```

Листинг 3: Функция `plot-gaussian`

Также напигем интерактивный виджет для взаимодействия с данной функцией:

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

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

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

Сначала сделаем `std` = 0,5:

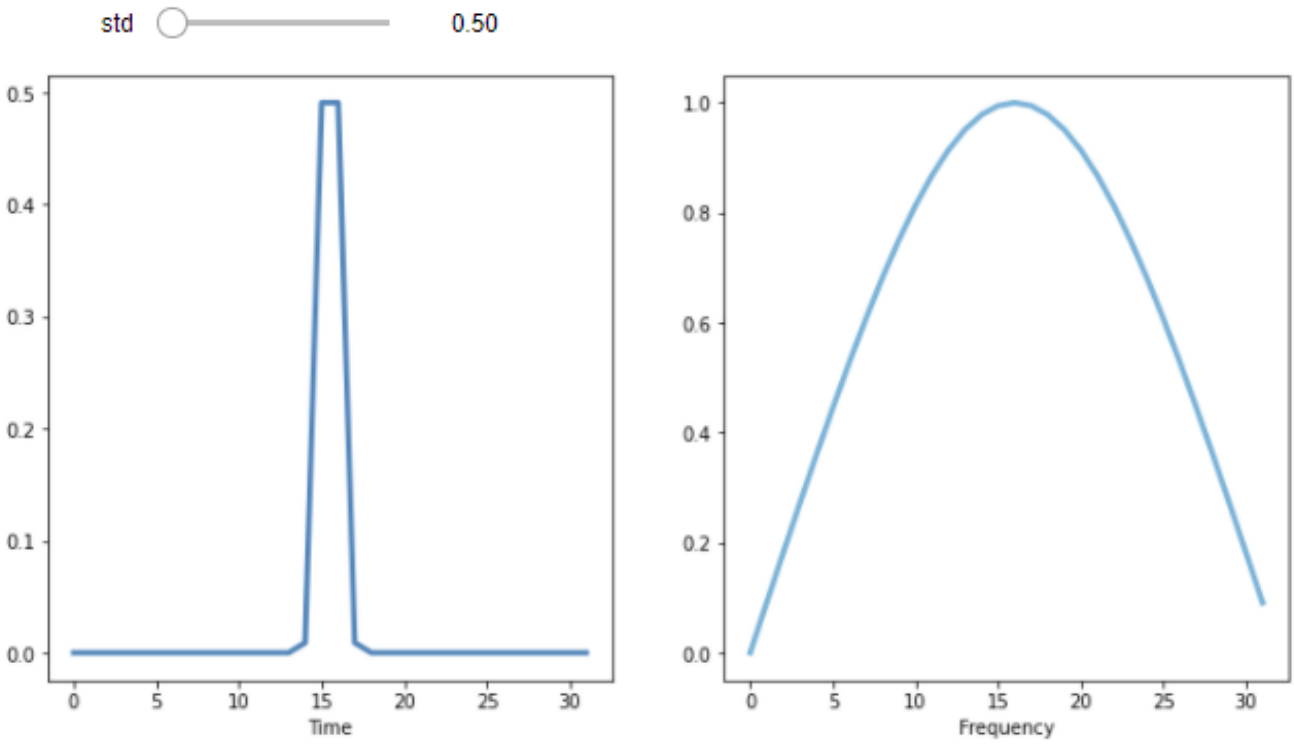


Рис. 5: Std = 0,5

После чего сделаем std = 2:

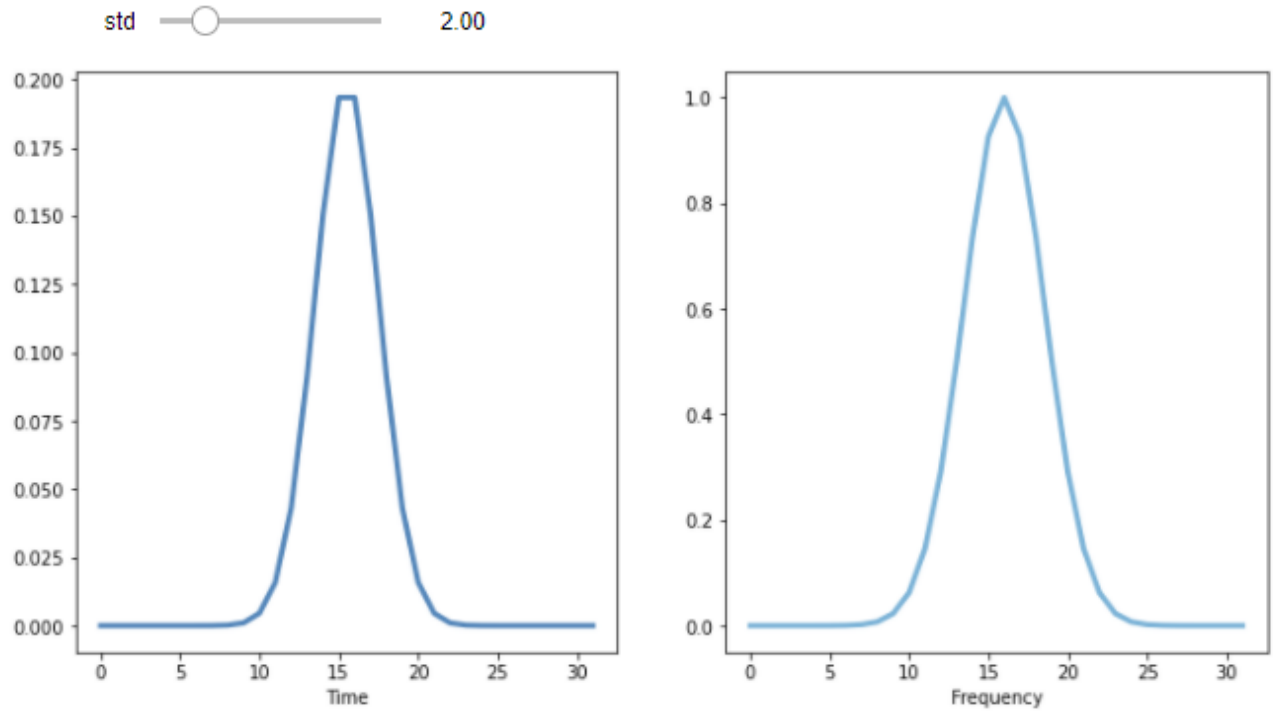


Рис. 6: Std = 2

Затем сделаем std = 5:

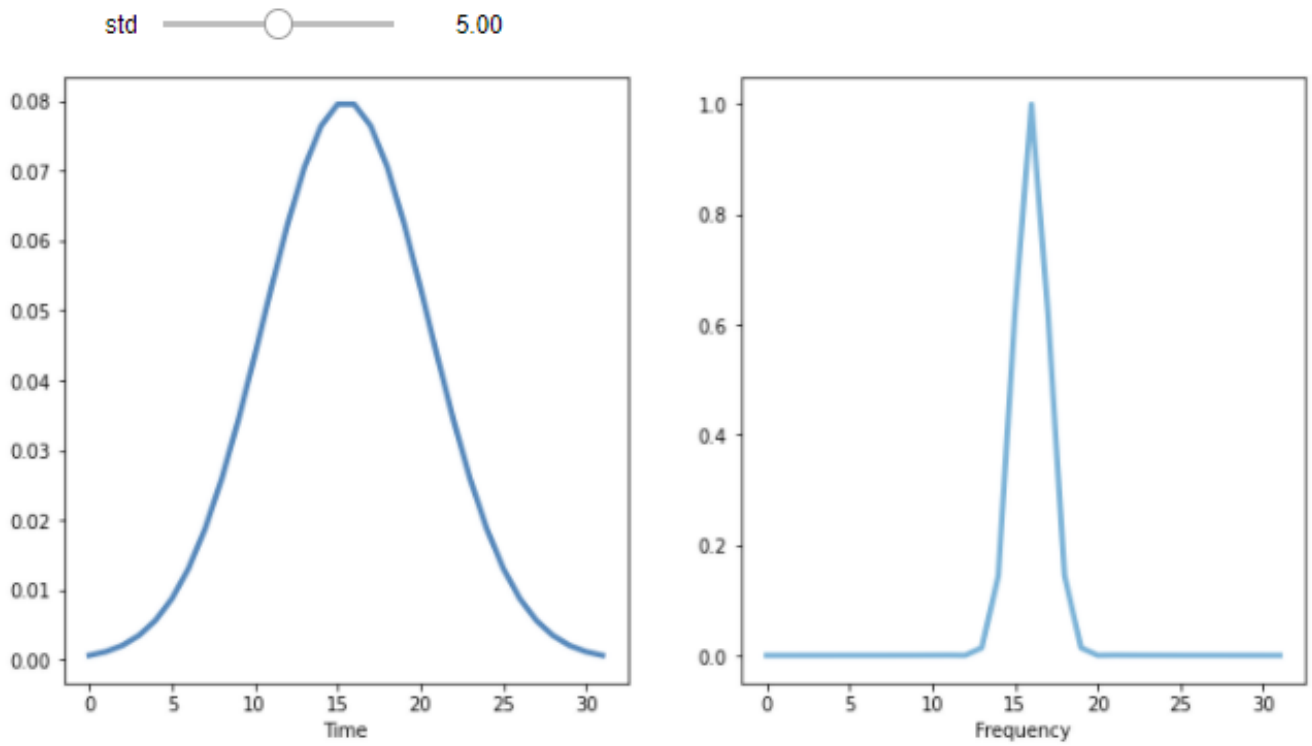


Рис. 7: `Std` = 5

И, наконец, сделаем максимальное `std` = 10:

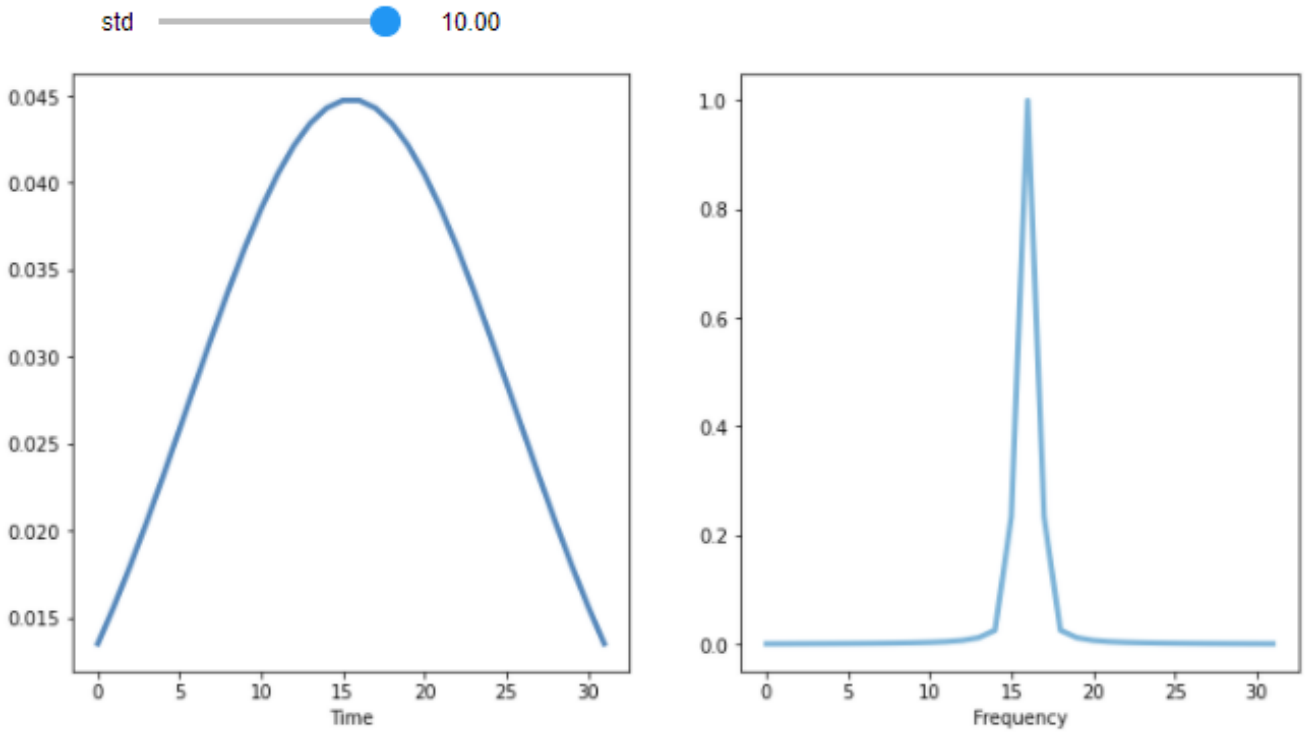


Рис. 8: `Std` = 10

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 восьмой 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 напишем функцию `plot-window`:

```
def plot_window(ax, window_fun, M=30):
    signal = SquareSignal(freq=440)
    wave = signal.make_wave(duration=1.0, framerate=44100)

    window = window_fun(M)
    window /= sum(window)

    padded = zero_pad(window, len(wave))
    fft = np.fft.rfft(padded)

    ax[0].plot(window, label=window_fun.__name__)
    ax[1].plot(abs(fft)[:8000], label=window_fun.__name__)
    plt.legend()

_, ax = plt.subplots(1, 2)
for w in [np.blackman, np.bartlett, np.hamming, np.hanning]:
    plot_window(ax, w)
```

Листинг 5: Функция `plot-window`

Запустим данную функцию и посмотрим на полученный результат:

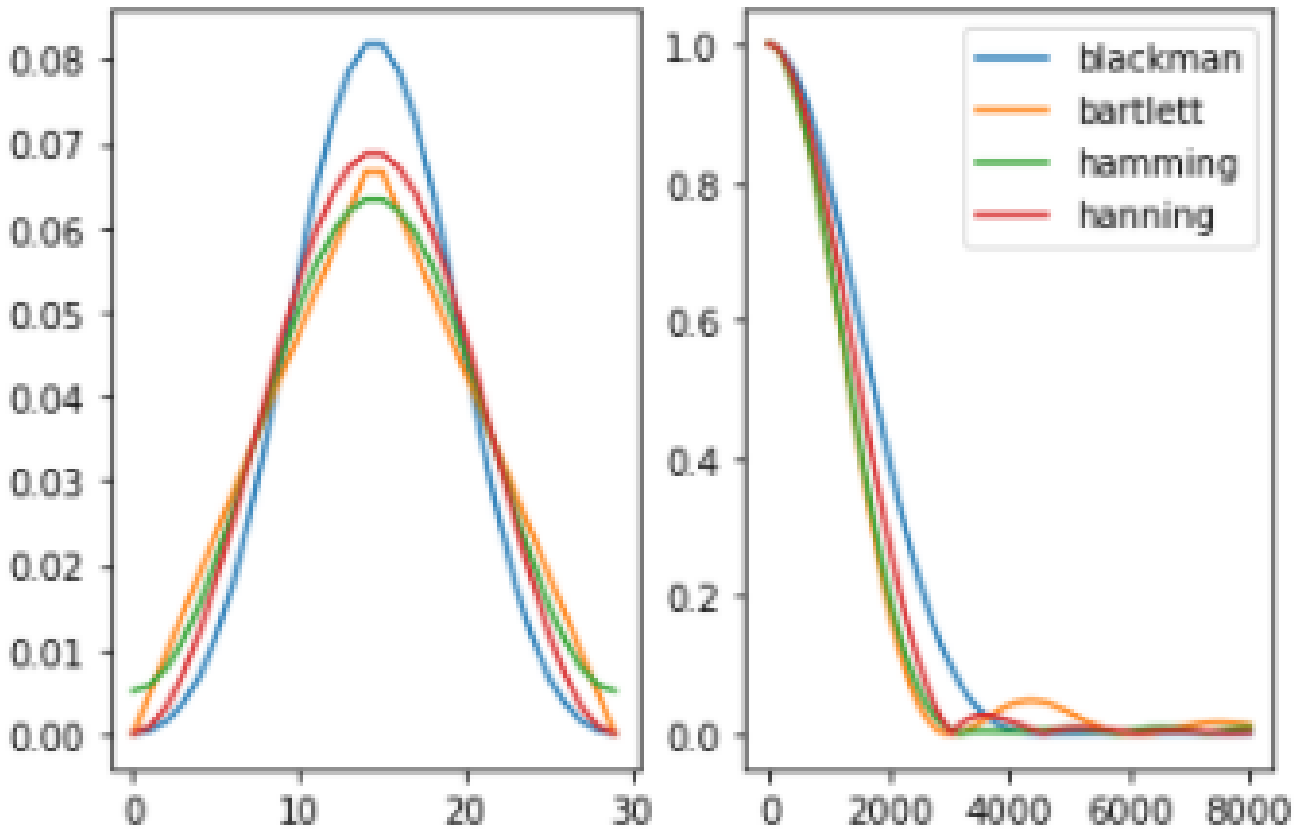


Рис. 9: Результат работы `plot-window`

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 данной лабораторной работы мы разобрались с тем, как гирина 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.