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
Сахибгареев Рамис Ринатович

Преподаватель:
Богач Наталья Владимировна

Санкт-Петербург 2021

Contents

List of Figures

Listings

1 Part 1: Research and execution of chap08

In this part we need to research and execute existing chap08.ipynb file, that contains information windows and auto-correlation.

File was successfully executed. Window - is a wa to smooth signal, make it less ”noisy”. Convolution window can be showed as amplitude to frequency ratio.

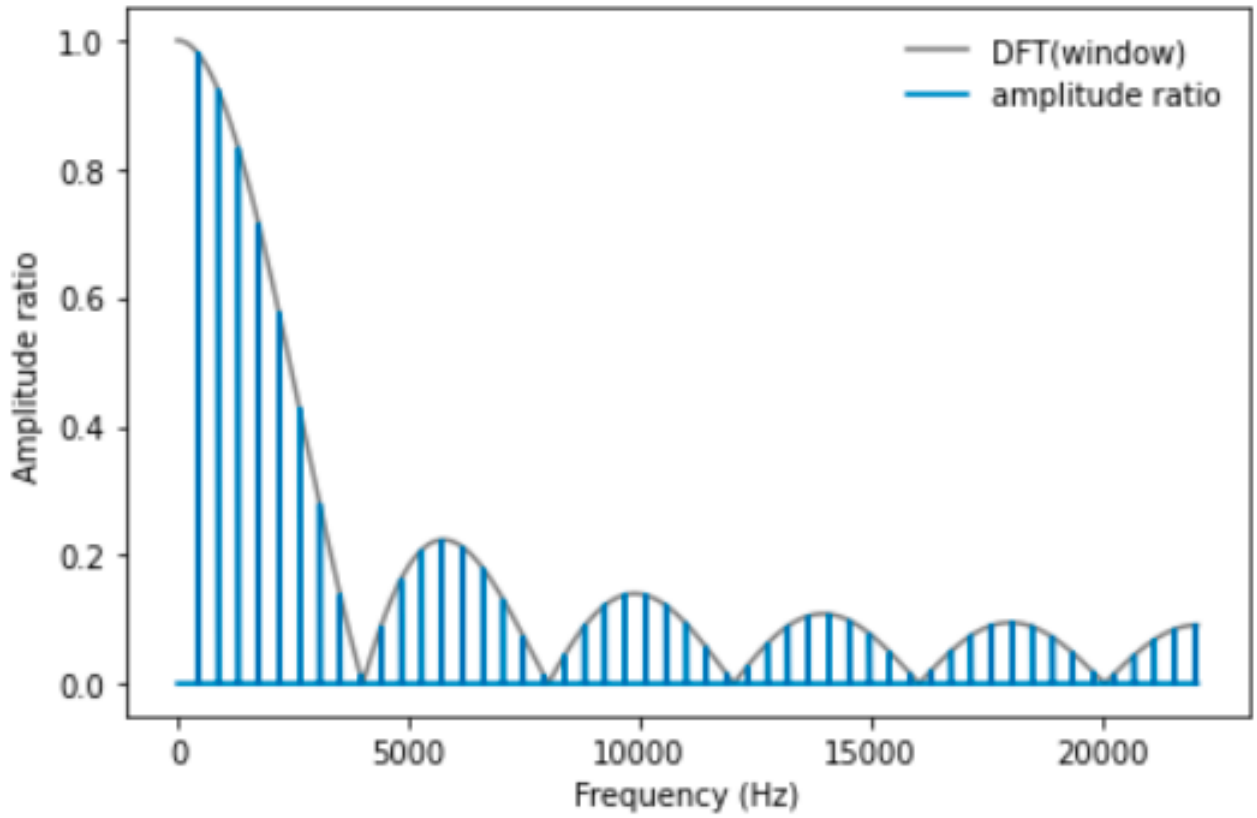

Figure 1: Ampltude to frequency ratio

We can test Gaussian Window by increasing standard deviation. We can see, as we increasing the standard deviation, ”боковые лепестки” increasing too. It happens, because Gaussian Window becomes more and more flat.

```
plot_filter(11, 2)
```

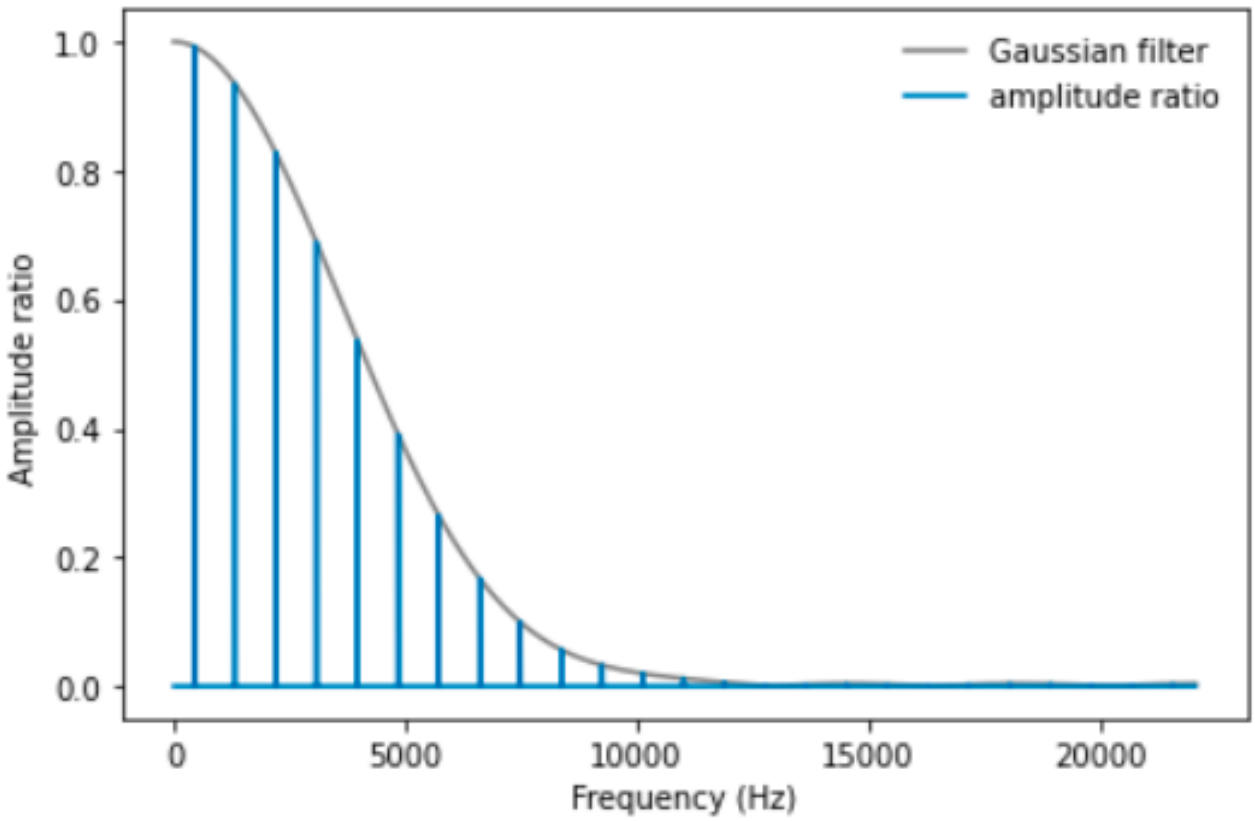

Figure 2: Gaussian window result (std = 2)

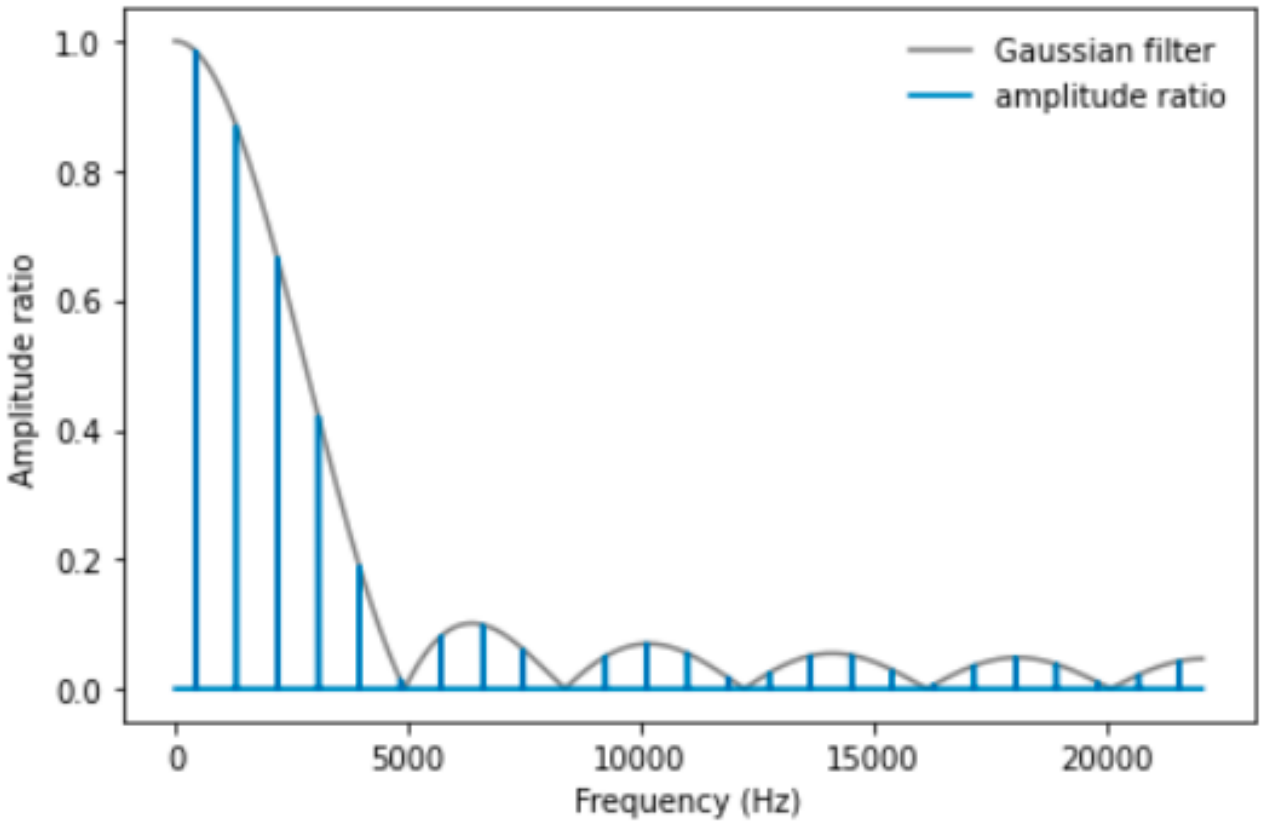

Figure 3: Gaussian window result (std = 4

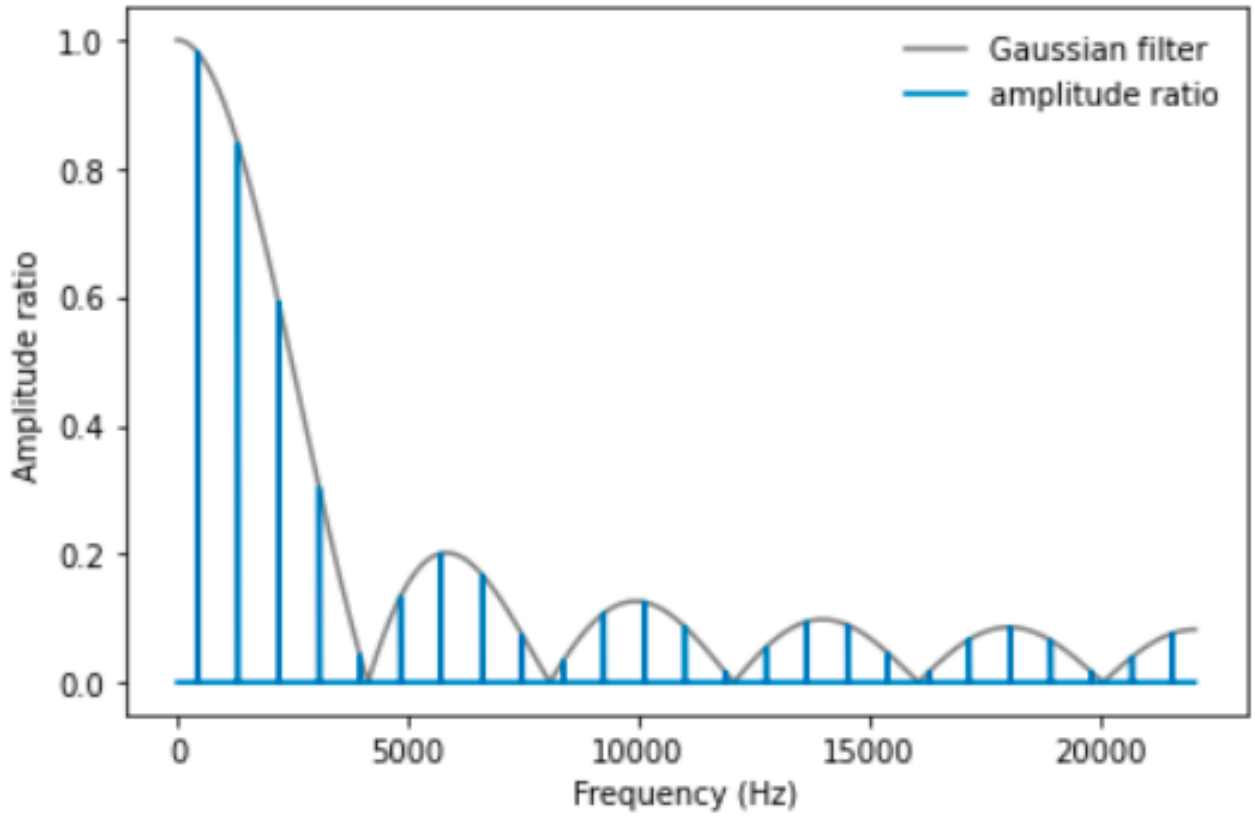

Figure 4: Gaussian window result (std = 10

2 Part 2: FFT on Gaussian

In this part we need to perform FFT on Gaussian curve with different standard deviation. Take into account, that plotting function rolls the FFT result, so we can clearly see gaussian. Actual zero frequency = half of plot frequency.

```
def plot_gaussian(std):
    M = 64
    gaussian = scipy.signal.gaussian(M=M, std=std)
    gaussian /= sum(gaussian)

    plt.subplot(1, 2, 1)
    plt.plot(gaussian)
    decorate(xlabel='Time')

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

    plt.subplot(1, 2, 2)
    plt.plot(np.abs(fft_rolled))
    decorate(xlabel='Frequency')
    plt.show()

```

Listing 1: Plotting function

```
plot_gaussian(0.5)
```

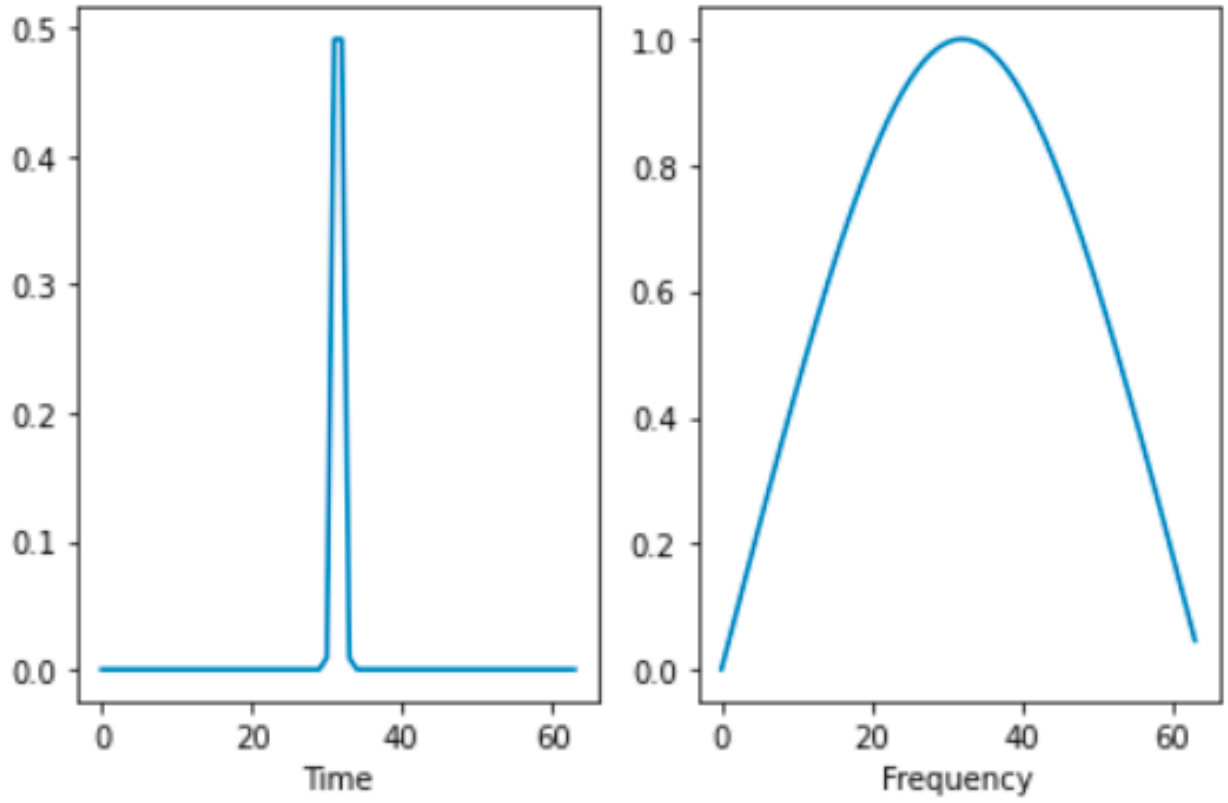

Figure 5: std = 0.5

```
plot_gaussian(2)
```

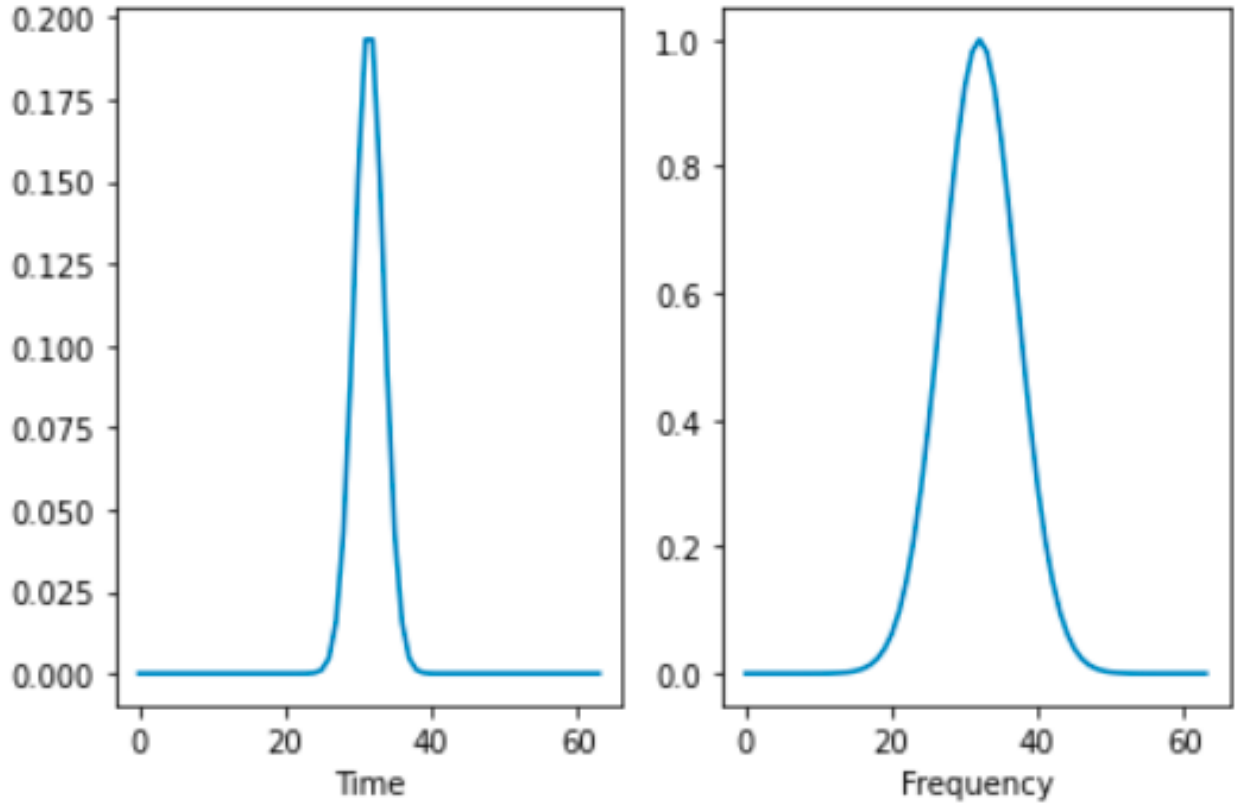

Figure 6: std = 2

```
plot_gaussian(5)
```

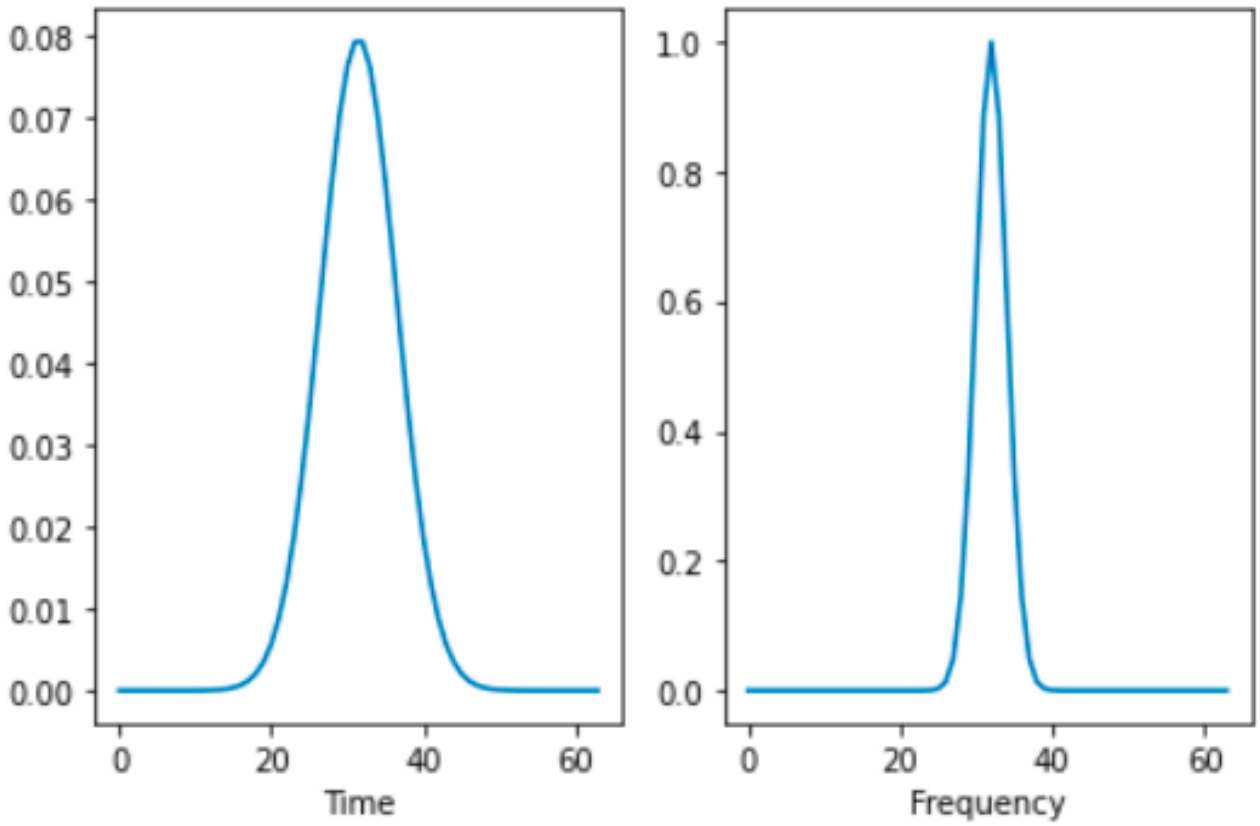

Figure 7: std = 5

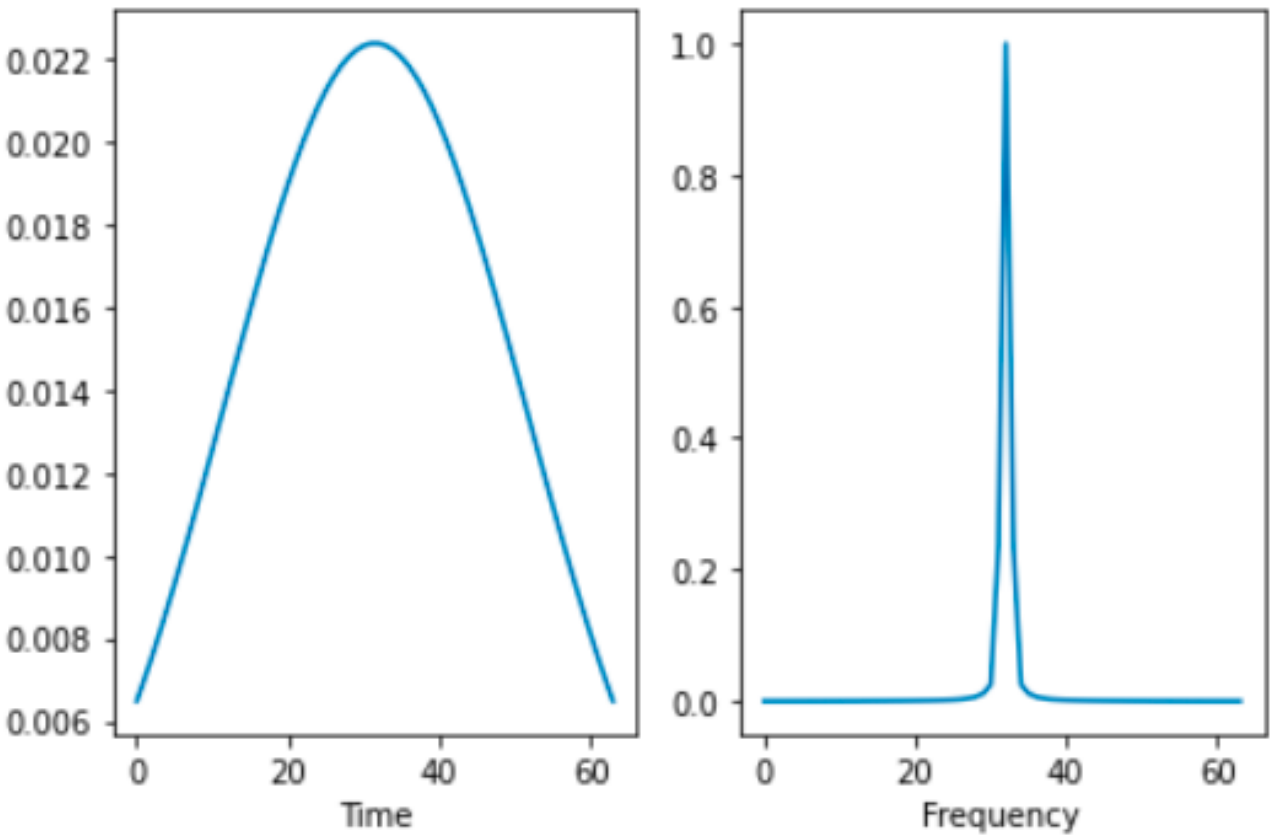

Figure 8: std = 20

We can clearly see, that by increasing standard deviation Gaussian curve becomes more flat and more looks like part of sin function. That's why frequency becomes more "clear" and concentrates around zero.

3 Different windows comparison

In this part, we need to compare different windows as they are good as a low-pass filter. Blackman, Gaussian, Hanning, Hamming, Bartlett filters was used.

```
    signal = SquareSignal(freq=440)
    wave = signal.make_wave(duration=1.0, framerate=441000)

    M = 30
    gaussian = scipy.signal.gaussian(M=M, std=2.5)
    bartlett = np.bartlett(M)
    blackman = np.blackman(M)
    hamming = np.hamming(M)
    hanning = np.hanning(M)

    windows = [blackman, gaussian, hanning, hamming, bartlett]
    names = ['blackman', 'gaussian', 'hanning', 'hamming', 'bartlett']
    for window in windows:
        window /= sum(window)

    for window, name in zip(windows, names):
        plt.plot(window, label=name)
        decorate(xlabel='Index')

    for window, name in zip(windows, names):
        padded =  zero_pad(window, len(wave))
        plt.plot(abs(np.fft.rfft(padded)), label=name)
        decorate(xlabel='Frequency')
        plt.ylim(top=0.1)
        plt.xlim(right = 20000)

```

Listing 2: Plotting code

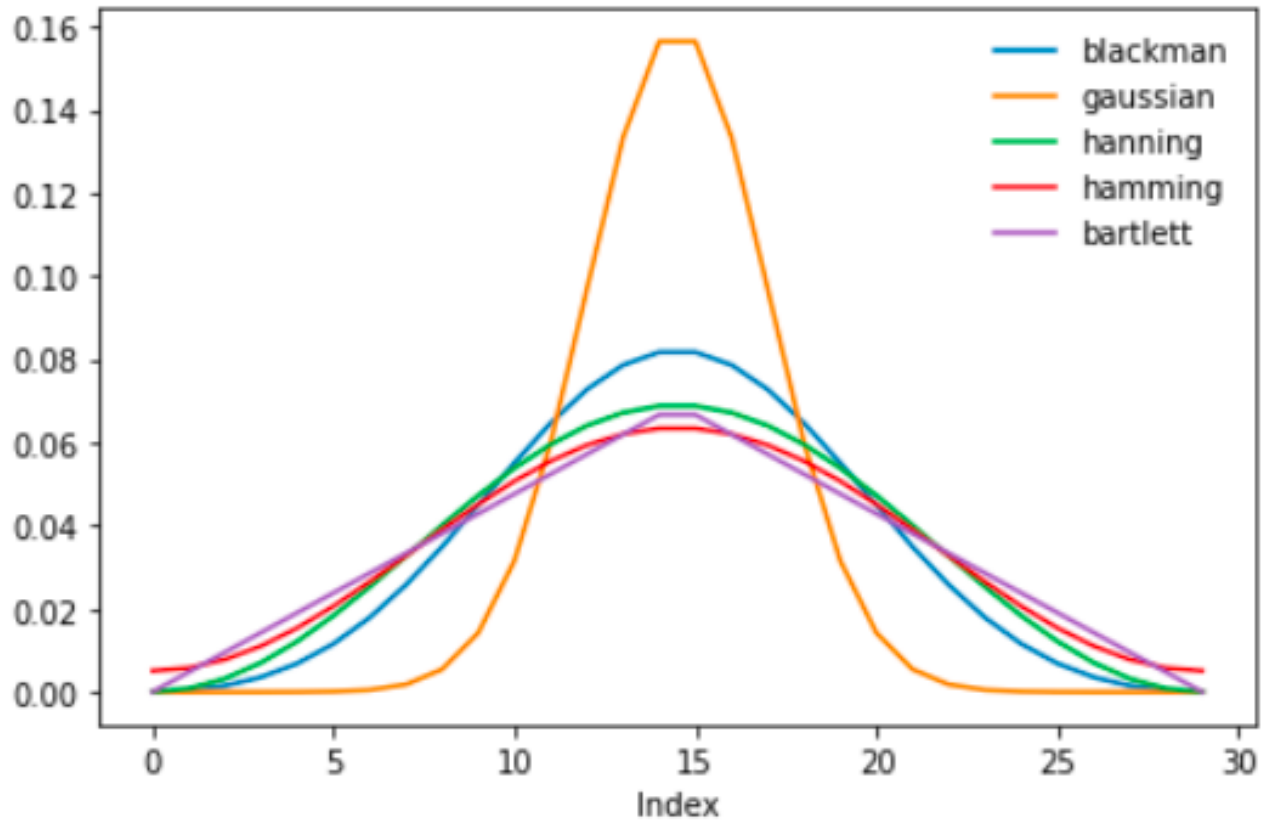

Figure 9: Resulting windows

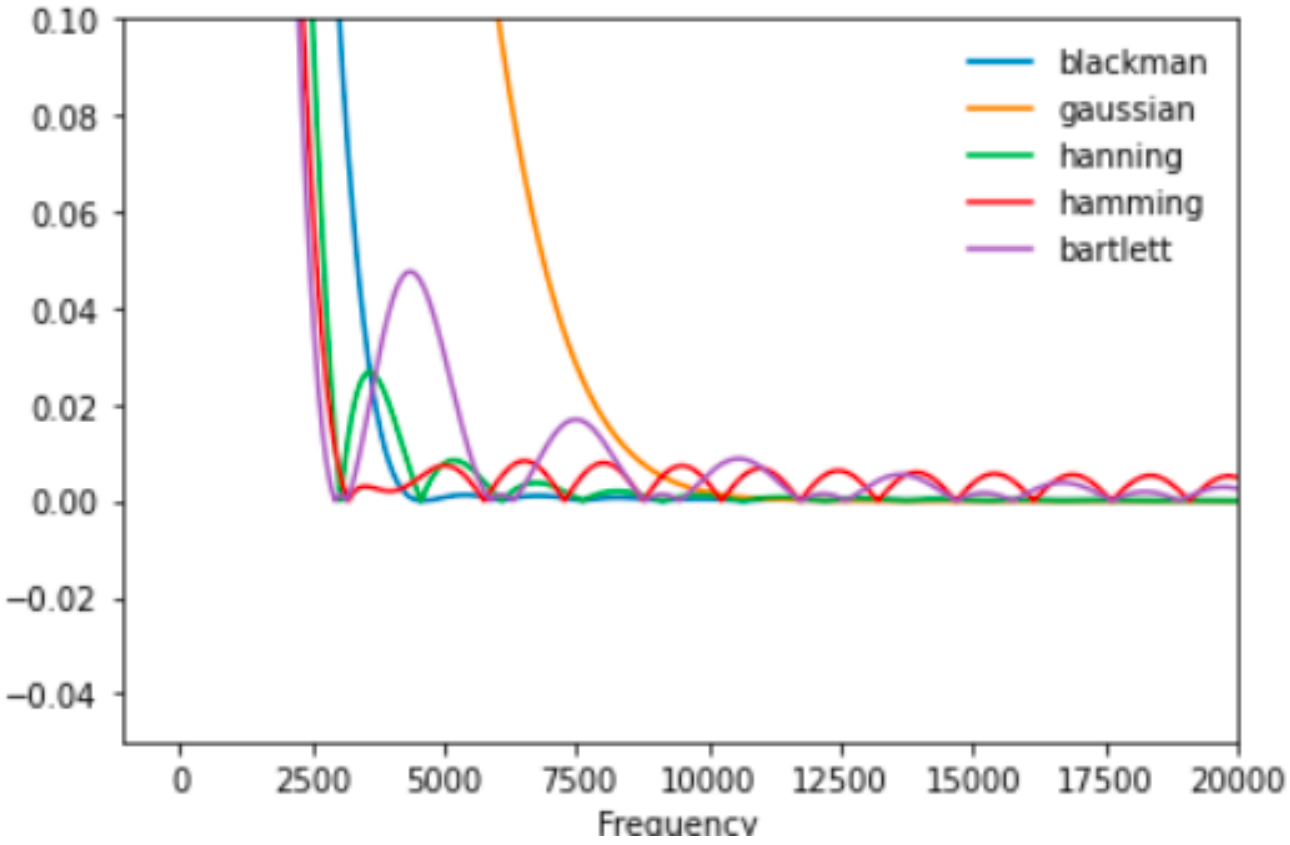

Figure 10: Resulting DFT plots

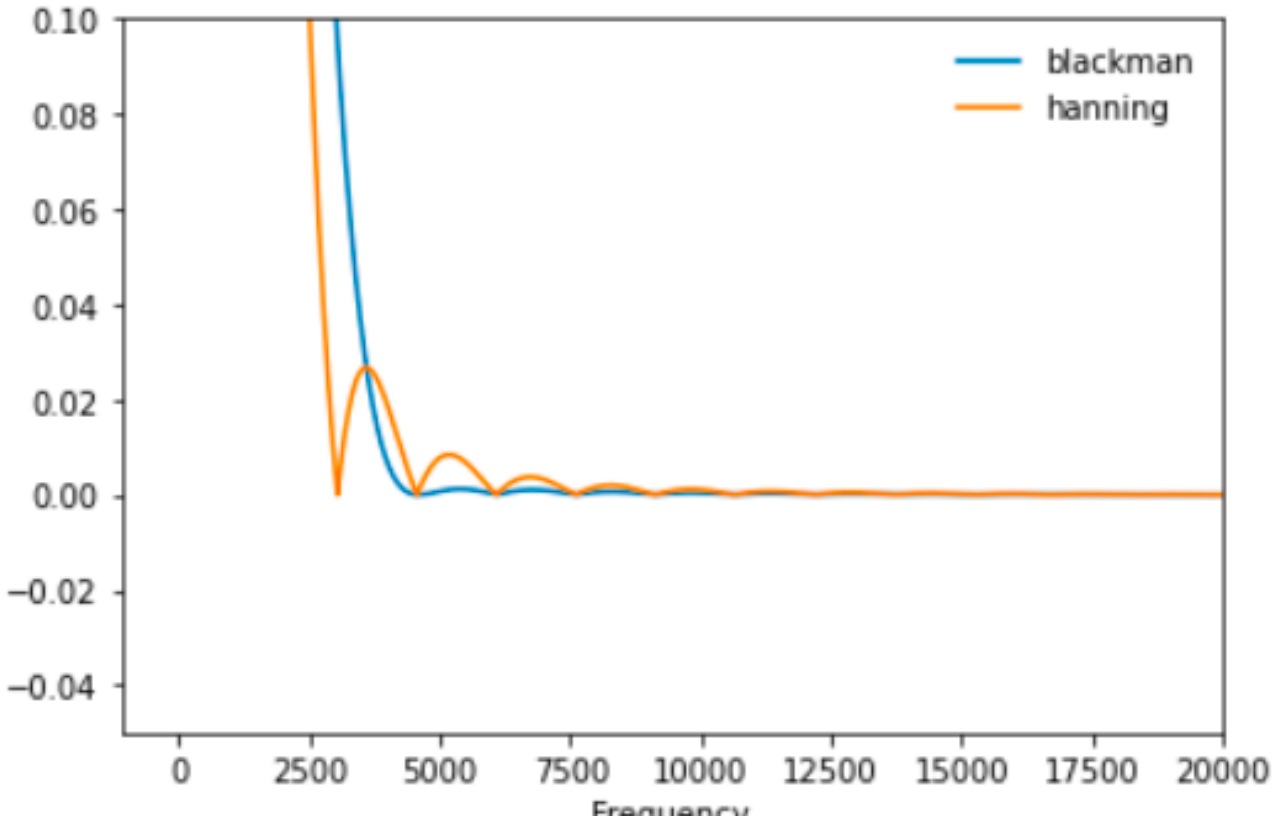

Figure 11: Hanning and Blackman side by side

We can see, that a Gaussian window drops off slow, but it most consistent and ”smooth”. However, there are good alternatives: a Hanning and a Blackman windows. They both drop off faster, than a Gaussian window, but also pretty smooth. Blackman window drops a bit slower, than a Hanning one, but it has much less ”bumps”.

4 Conclusion

We’ve learned, what is filtering windows and autocorrelation. We’ve made some test, to find out how different standard deviation affects the Gaussian window: with big standard deviation values Gaussian filter is very ”bumpy” and ”unsmooth”. We’ve discovered, that DFT of a Gaussian curve is a Gaussian curve also, but with reversed standard deviation. We’ve compared different windows as they are good as a low-pass filter. The result is that answer is complex: it can be a Blackman window, if we need to have a very smooth filter with pretty fast drop off, or it can be a Hanning window if we need very fast drop off, but we can have some unsmootheness in out filter.