

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе №8
Дисциплина: Телекоммуникационные технологии
Тема: Фильтрация и свёртка

Работу выполнил:
Ляшенко В.В.
Группа: 3530901/80201
Преподаватель:
Богач Н.В.

Санкт-Петербург
2021

Оглавление

1	Упражнение 8.1	4
2	Упражнение 8.2	7
3	Упражнение 8.3	10
4	Выводы	13

Список иллюстраций

1.1	Гауссово окно. $\text{std} = 2$	4
1.2	Гауссово окно. $\text{std} = 3.5$	5
1.3	Гауссово окно. $\text{std} = 6.5$	5
1.4	Гауссово окно. $\text{std} = 20$	6
2.1	Гауссово окно и его БПФ	8
2.2	Уменьшение std	8
2.3	Увеличение std	9
3.1	Графики окон	11
3.2	ДПФ окон	11
3.3	ДПФ окон в логарифмическом масштабе	12

Листинги

2.1	Функция <code>plot_gaussian</code>	7
3.1	Создание сигнала	10
3.2	Построение окон	10
3.3	Построение ДПФ	11
3.4	Построение ДПФ в логарифмическом масштабе	12

Глава 1

Упражнение 8.1

В начале мы запустим примеры из `chap08.ipynb`.

В данном коде есть интерактивный виджет, в котором можно поэкспериментировать с параметрами Гауссова окна и изучить их влияние на частоту среза. Воспользуемся им.

Установим значение $M = 20$, $\text{std} = 2$ (Рис.1.1).

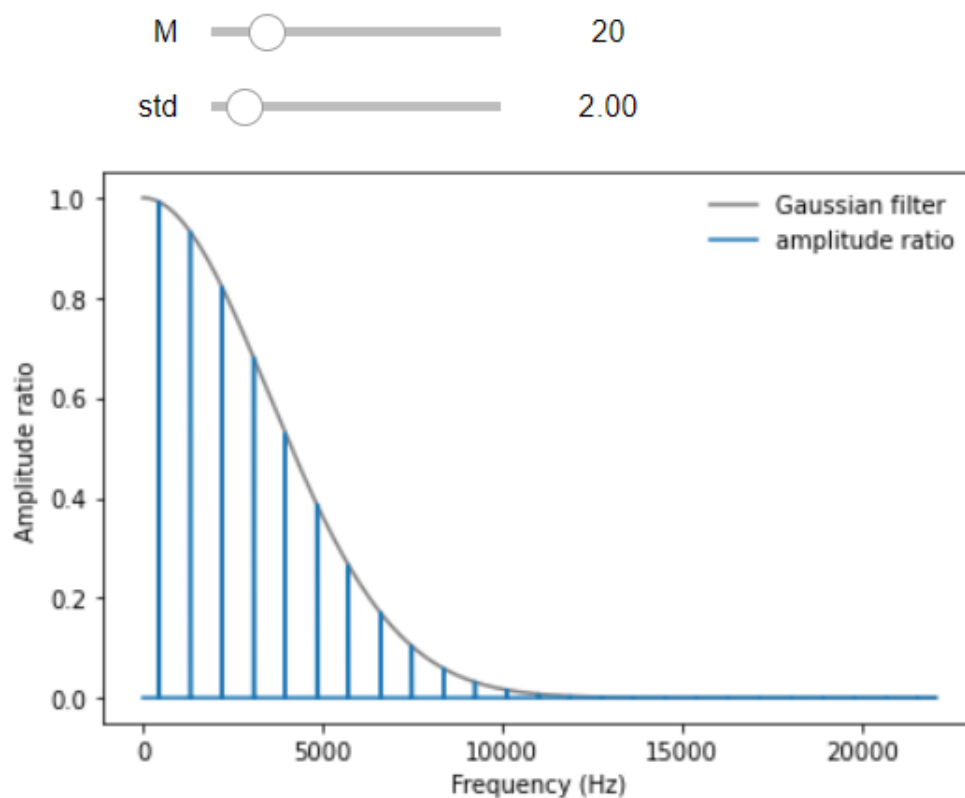


Рис. 1.1: Гауссово окно. $\text{std} = 2$

Теперь увеличим std до 3.5 (Рис.1.2).

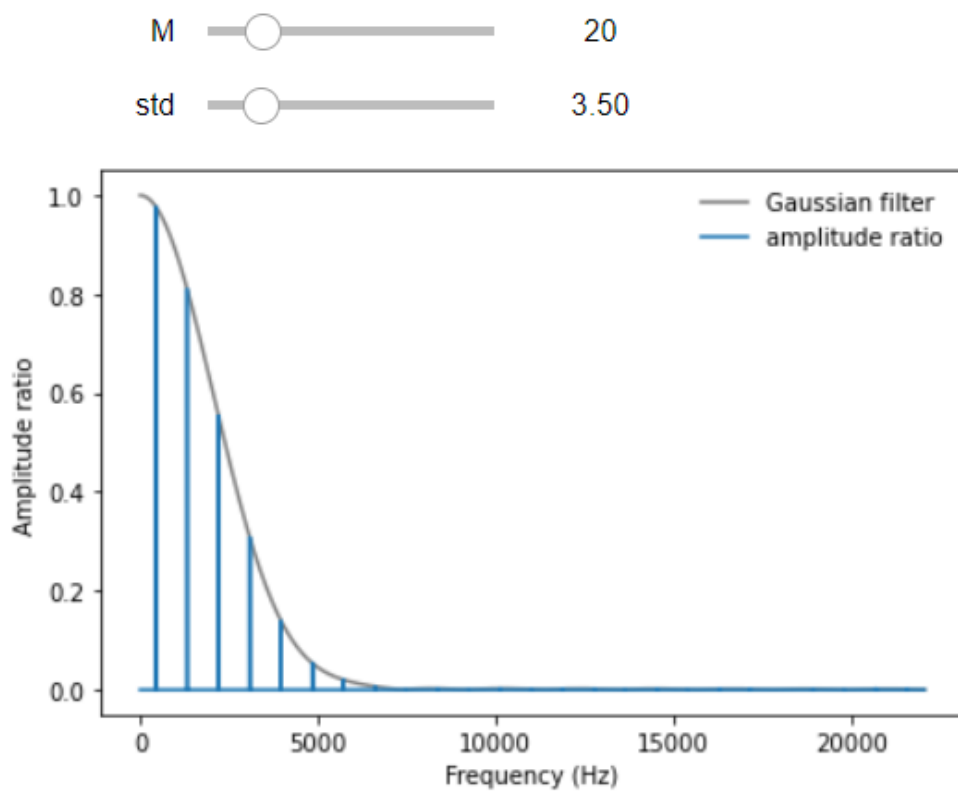


Рис. 1.2: Гауссово окно. $\text{std} = 3.5$

Продолжим увеличивать значение std до 6.5 (Рис.1.3).

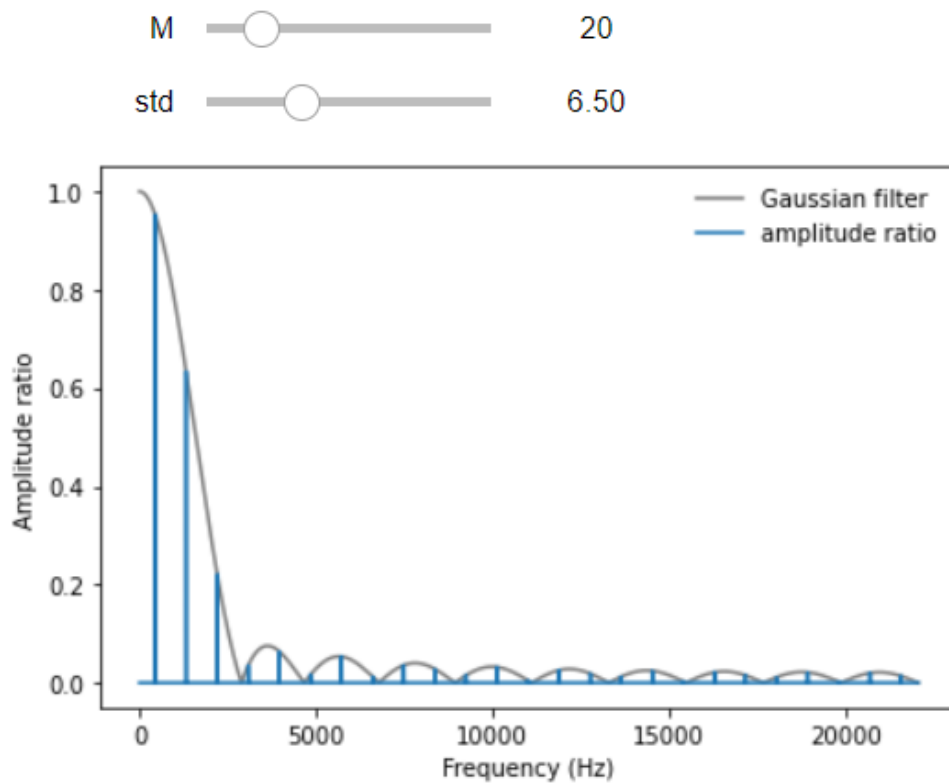


Рис. 1.3: Гауссово окно. $\text{std} = 6.5$

И, наконец, поставим максимальное значение $\text{std} = 20$ (Рис.1.4).

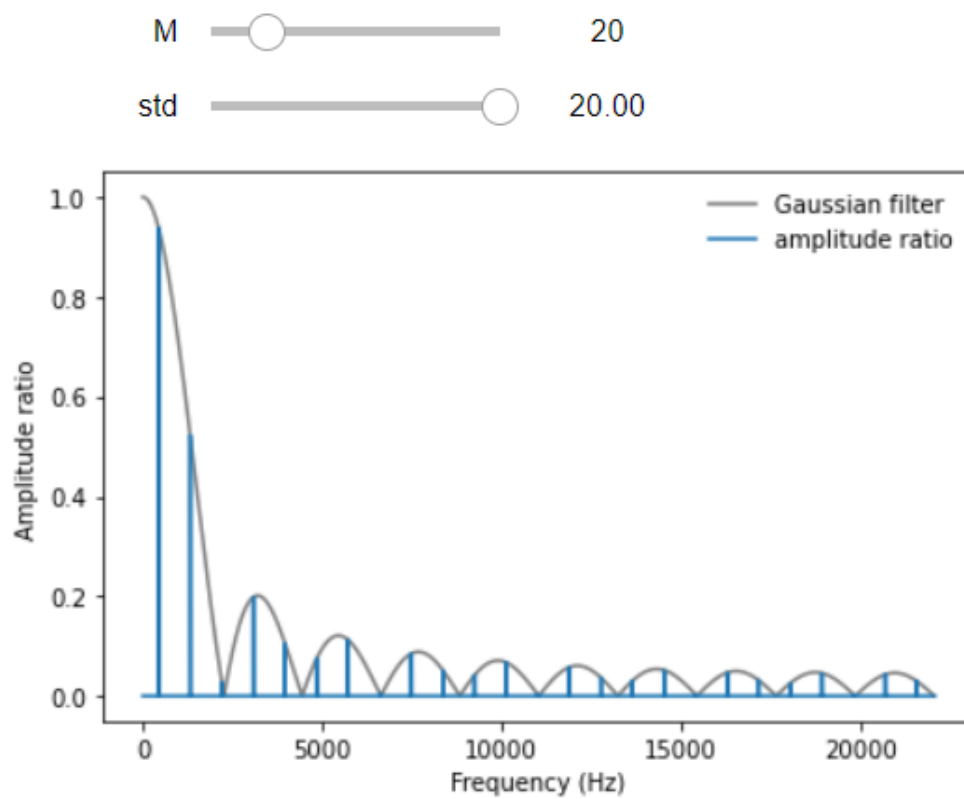


Рис. 1.4: Гауссово окно. $\text{std} = 20$

Как мы можем видеть, без увеличения значения M , окно постепенно сжимается, высокочастотные гармоники в спектре спадают медленно и из-за этого появляются боковые лепестки.

Глава 2

Упражнение 8.2

Используем преобразование Фурье гауссовой кривой на нескольких примерах.

Создадим функцию `plot_gaussian`, которая создает окно Гаусса и его БПФ и отображает их графики.

```
def plot_gaussian(std):
    M = 32
    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()

plot_gaussian(2)
```

Листинг 2.1: Функция `plot_gaussian`

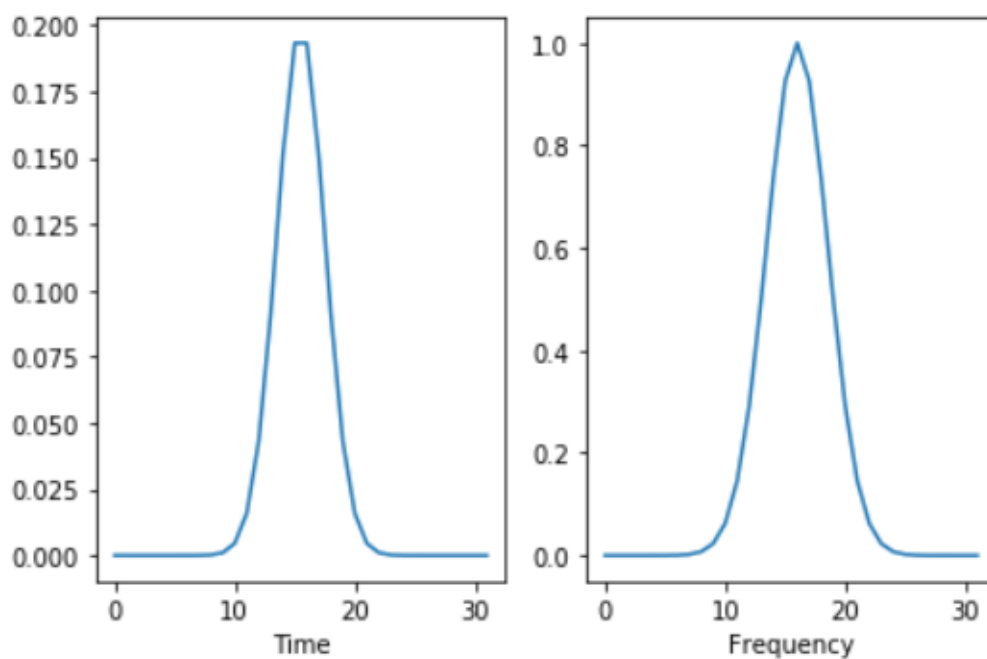


Рис. 2.1: Гауссово окно и его БПФ

Теперь добавим слайдер, с помощью которого можно будет изменять std , и посмотрим, что происходит с ПФ.

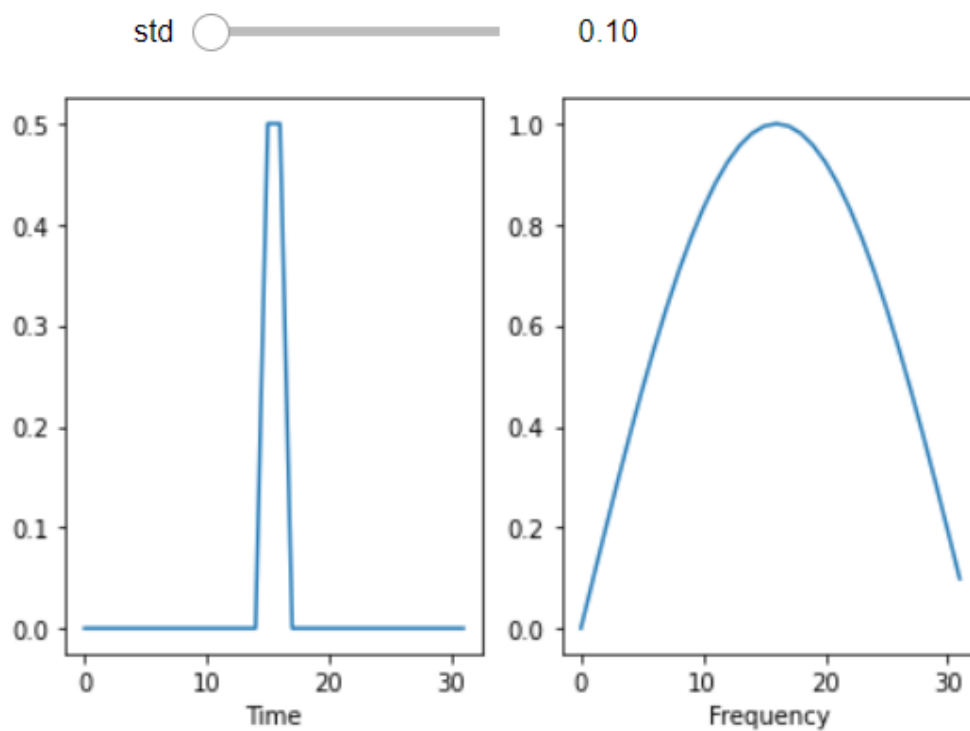


Рис. 2.2: Уменьшение std

При уменьшении std ПФ сжимается, а его БПФ расширяется (Рис.2.2).

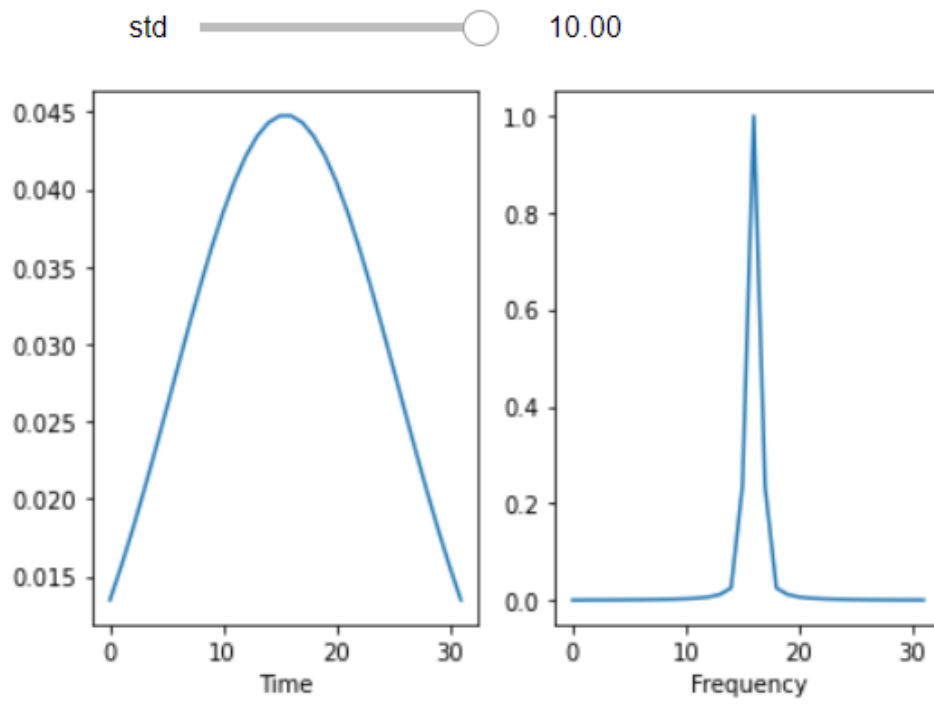


Рис. 2.3: Увеличение std

При увеличении мы видим обратную ситуацию, ПФ расширяется, а БПФ сужается. Таким образом, между ними есть обратная зависимость.

Глава 3

Упражнение 8.3

Ранее мы изучали влияние на утечки спектра окна Хэмминга и некоторых других окон. Чтобы глубже понять эти окна, изучим их ДПФ.

Создадим сигнал для дальнейшей работы с ним.

```
from thinkdsp import SquareSignal

signal = SquareSignal(freq=440)
wave = signal.make_wave(duration=1.0, framerate=44100)
```

Листинг 3.1: Создание сигнала

Затем построим все нужные нам окна и напечатаем их графики.

```
M = 15
std = 2.5

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

windows = [blackman, gaussian, hanning, hamming]
names = ['blackman', 'gaussian', 'hanning', 'hamming']

for window in windows:
    window /= sum(window)

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

decorate(xlabel='Index')
```

Листинг 3.2: Построение окон

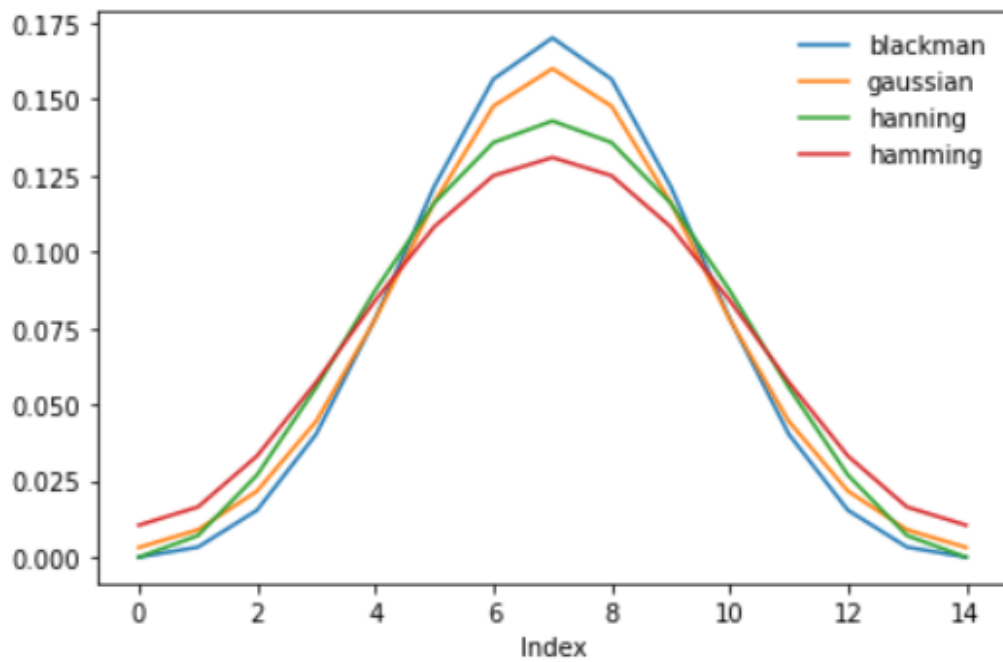


Рис. 3.1: Графики окон

Как мы можем видеть на рис.3.1, их графики довольно похожи. Теперь получим их ДПФ.

```
plot_window_dfts(windows, names)
decorate(xlabel='Frequency (Hz)')
```

Листинг 3.3: Построение ДПФ

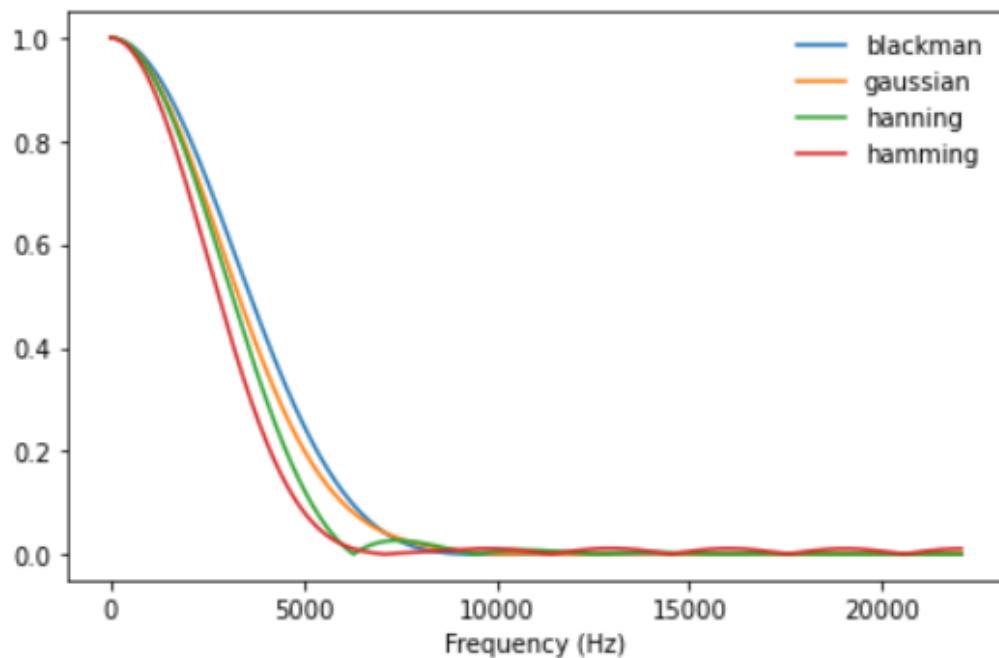


Рис. 3.2: ДПФ окон

Из рис.3.2 видно, что ДПФ окон тоже похожи. Однако Хэмминг спадает быстрее всех, Блэкман - медленнее всех, а у Ханнинга самые заметные боковые лепестки.

Построим эти же графики в логарифмическом масштабе.

```
plot_window_dfts(windows, names)
decorate(xlabel='Frequency (Hz)', yscale='log')
```

Листинг 3.4: Построение ДПФ в логарифмическом масштабе

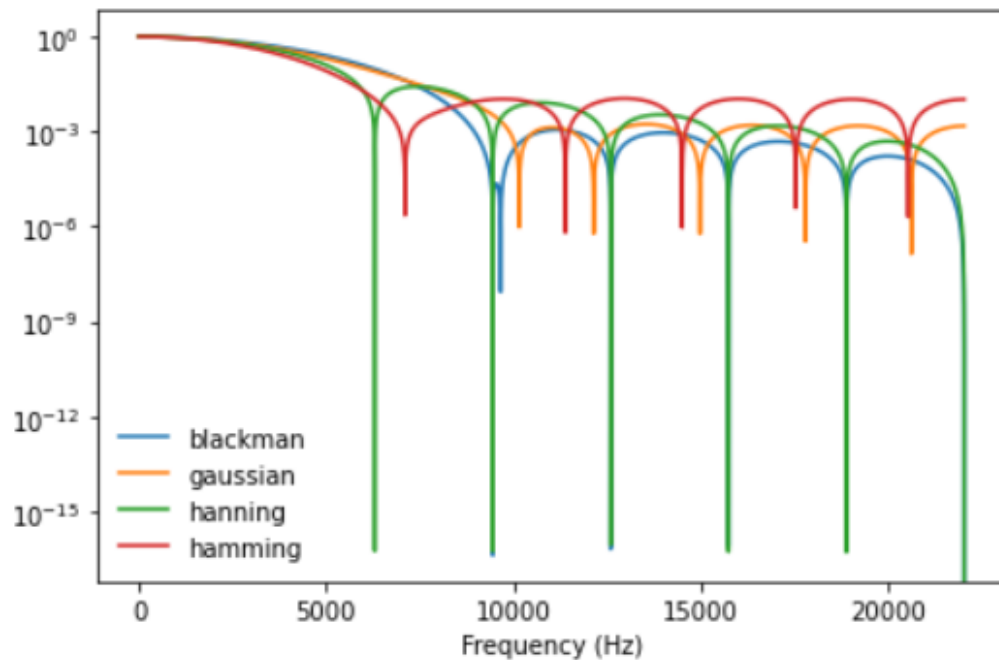


Рис. 3.3: ДПФ окон в логарифмическом масштабе

Из рис.3.3 мы можем увидеть, что сначала значения Хэмминга и Хеннинга падают быстрее, чем два других. Окна Гаусса и Хэмминга имеют самые стойкие боковые лепестки. Окно Ханнинга может иметь наилучшее сочетание быстрого падения и минимальных боковых лепестков.

Глава 4

Выводы

В результате выполнения данной работы мы изучили понятия фильтрации и свёртки. Также мы получили навыки работы с Гауссовым окном и лучше изучили разные окна, через их ДПФ.