

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

# Телекоммуникационные технологии

Отчёт по лабораторной работе №8

**Работу**  
**выполнил:**  
Смирнов Л. Д.  
**Группа:**  
3530901/80202  
**Преподаватель:**  
Богач Н. В.

Санкт-Петербург  
2021

# Содержание

<b>1. Выполнение работы</b>	<b>3</b>
1.1. Упражнение 1 . . . . .	3
1.2. Упражнение 2 . . . . .	3
1.3. Упражнение 3 . . . . .	3
<b>2. Выводы</b>	<b>4</b>

# 1. Выполнение работы

## 1.1. Упражнение 1

Согласно указаниям в книге я изучил и запустил пример из файла chap08. Как показал небольшой "опыт", увеличение параметра `std` при неизменном значении `m` приводит к появлению боковых лепестков, что говорит об ухудшении подавления высоких частот.

## 1.2. Упражнение 2

Для исследования изменения параметров сигнала используем следующую функцию, позволяющую изобразить рядом Гауссово окно и быстрое преобразование Фурье.

```
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()
```

Для исследования зависимости от значений `std` добавим соответствующий виджет:

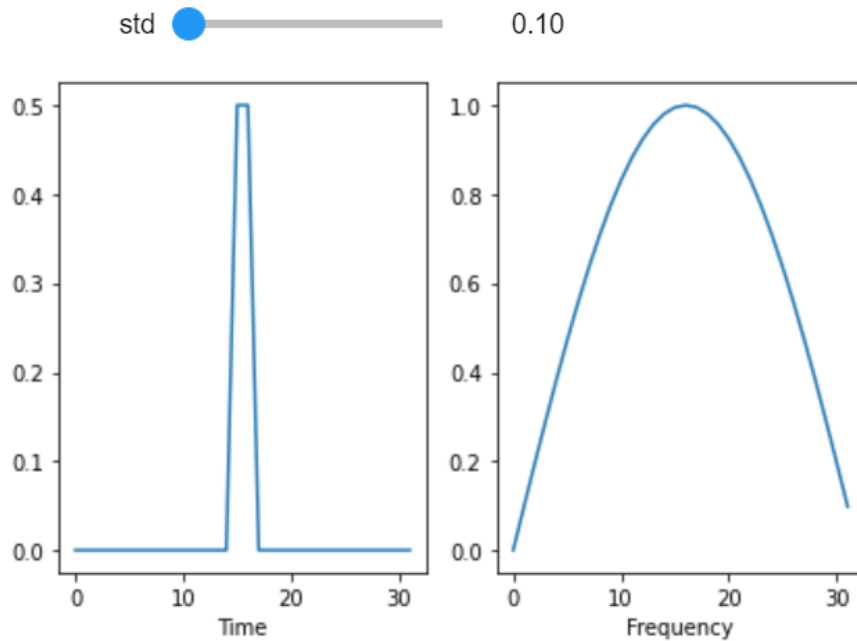
```
slider = widgets.FloatSlider(min=0.1, max=10, value=2)  
interact(plot_gaussian, std=slider);
```

Ниже приведу пару скриншотов, в подкрепление наблюдений о том, что при изменении параметра `std` один график "расширяется", второй "сужается" и наоборот, при обратных изменениях.

## 1.3. Упражнение 3

В этом упражнении нужно провести сравнение разных видов окон, речь о которых шла в предыдущей главе. Для это сначала создадим их.

```
M = 12  
std = 2.0  
  
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)
```

И сравним их вид для заданных параметров.

Далее выведем их дискретные преобразования Фурье в логарифмическом масштабе, как указано в книге.

```
for window, name in zip(windows, names):
    padded = zero_pad(window, len(wave))
    dft_window = np.fft.rfft(padded)
    plt.plot(abs(dft_window), label=name)
```

```
decorate(xlabel='Frequency (Hz)', ylabel='log')
```

Получим следующий результат:

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

## 2. Выводы

В ходе выполнения данной лабораторной работы я познакомился с понятием сглаживания и свертки, а так же изучил теорему о свертке.

