ThinkDSP. Лабораторная 8. Фильтрация и свертка.

Шерепа Никита

13 мая 2021 г.

Содержание

Список иллюстраций

Листинги

1 Упражнение 8.1

1. Задание

   Что случится, если при увеличении ширины гауссова окна `std` не увеличивать число элементов в окне `M`?

2. Ход работы

   При увеличении ширины гауссова окна `std` без увеличения числа элементов в окне `M` прямоугольное окно превращается в "скачкообразное"

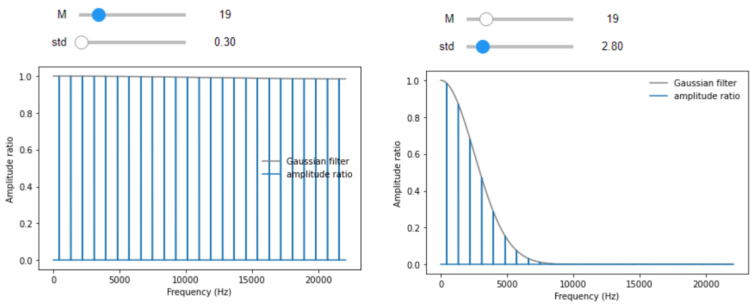

Рис. 1: 1

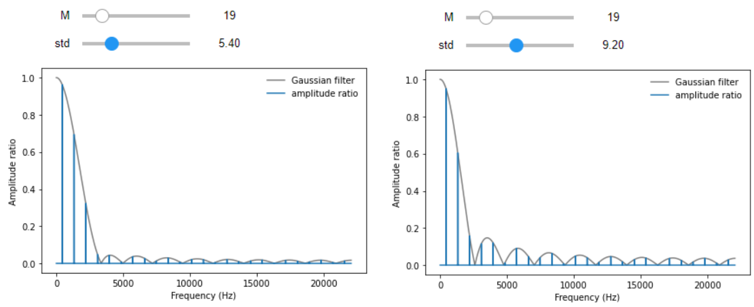

Рис. 2: 2

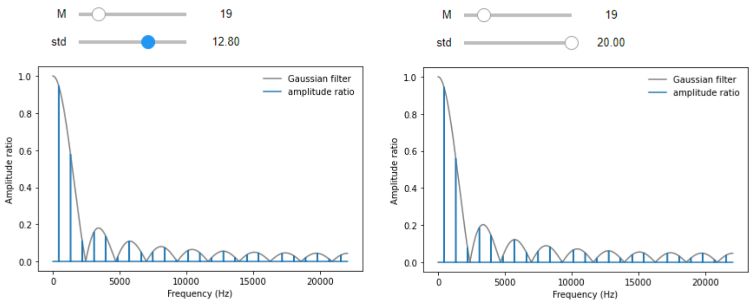

Рис. 3: 3

2 Упражнение 8.2

1. Задание

   Преобразование Фурье гауссовой кривой - также гауссовая кривая. Для дискретного преобразования Фурье это соотношение приблизительно верно.

   Попробуйте его на нескольких примерах. что происходит с преобразованием Фурье, если меняется `std`

2. Ход работы

   Рассмотрим Гауссиан

```
import scipy.signal

gaussian = scipy.signal.gaussian(M=32, std=2)
gaussian /= sum(gaussian)
plt.plot(gaussian)
decorate(xlabel='Index')
```

Листинг 1: Построение Гауссиана

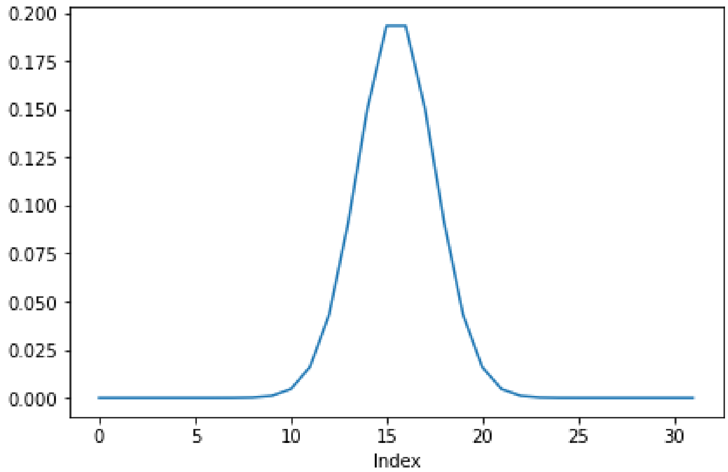

Рис. 4: Гауссиан

Применим к нему БПФ

```
fft_gaussian = np.fft.fft(gaussian)
plt.plot(abs(fft_gaussian))
decorate(xlabel='Frequency (Hz)', ylabel='Amplitude')
```

Листинг 2: Построение Гауссиана

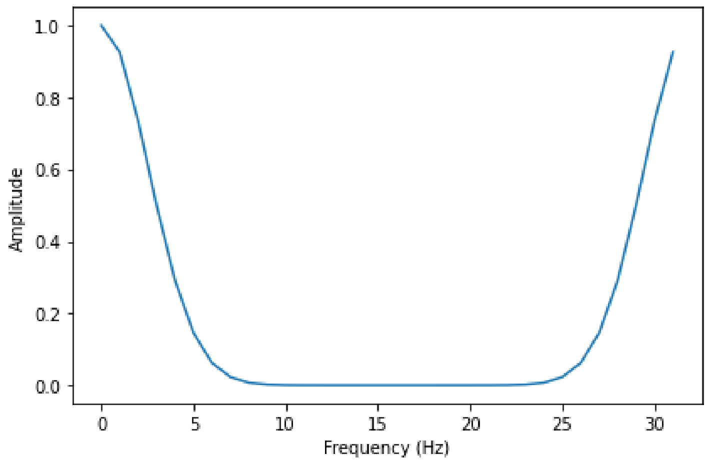

Рис. 5: Гауссиан после БПФ

Если повернуть отрицательные частоты влево, то можно четче увидеть, что это Гауссиан.

```
N = len(gaussian)
fft_rolled = np.roll(fft_gaussian, N//2)
plt.plot(abs(fft_rolled))
decorate(xlabel='Frequency (Hz)', ylabel='Amplitude')
```

Листинг 3: Построение Гауссиана

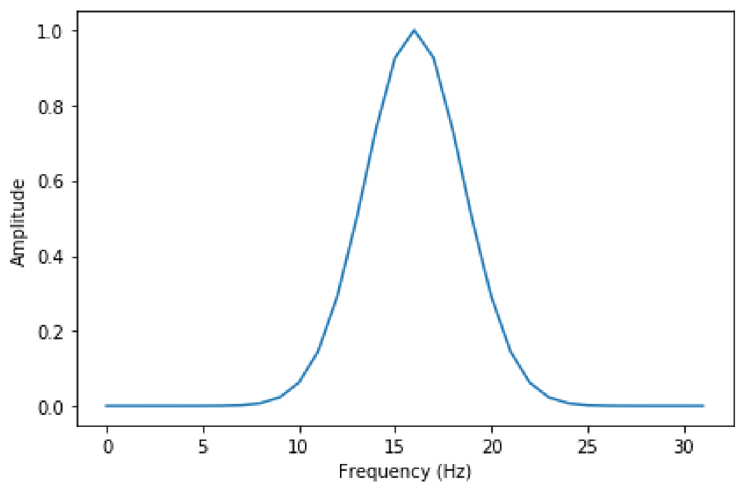

Рис. 6: Гауссиан четче

Рассмотрим функцию, которая отображает Гауссиан и БПФ Гауссиан для сравнения

```
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 4: Функция `plot_gaussian()`

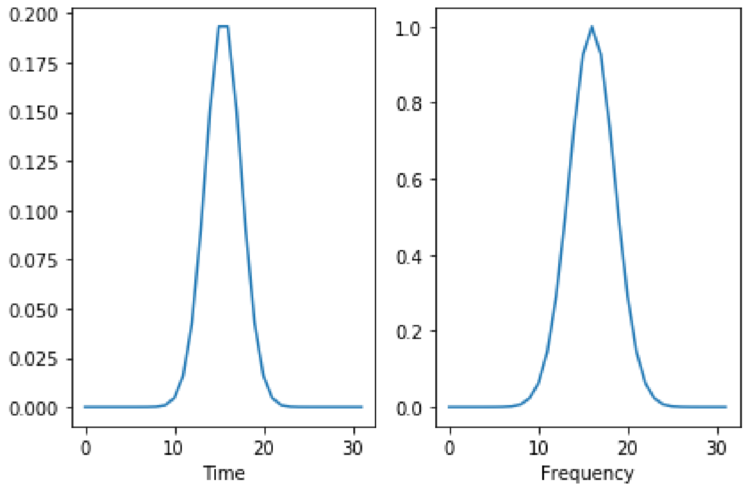

Рис. 7: Гауссиан и его БПФ

Теперь проверим, что произойдет при изменении `std`

```
from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets

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

Листинг 5: Изменяем `std`

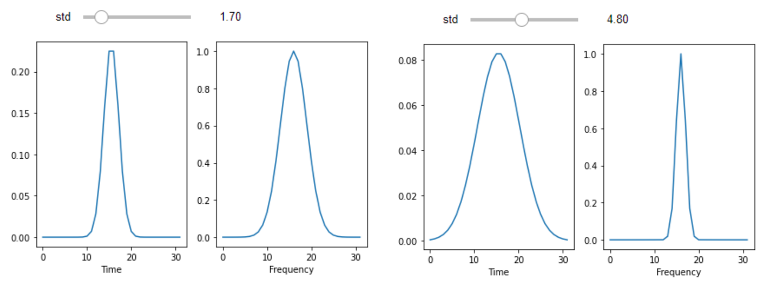

Рис. 8: Изменяем `std`

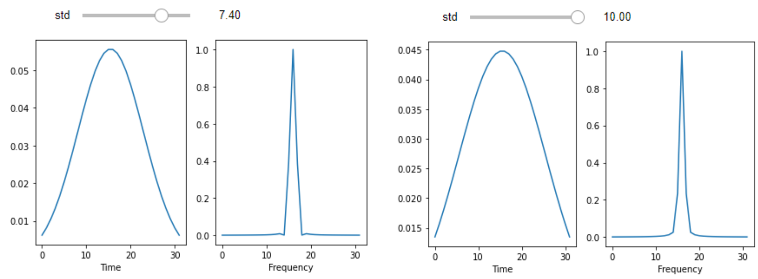

Рис. 9: Изменяем `std`

Видим, что по мере увеличения `std` Гауссиан становится шире, а его БПФ сужается.

Если $f(x) = e^{-ax^2}$, что является Гауссианом со средним значением $= 0$ и стандартным отклонением $= 1/a$, то тогда преобразование Фурье имеет вид

$F(k) = \sqrt{\frac{\pi}{a}} e^{-\pi^2 k^2 / a}$

что является Гауссианом со стандартным отклонением $= a/\pi^2$. Таким образом, существует обратная зависимость между стандартными отклонениями f и F.

3 Упражнение 8.3

(a) Задание

В дополнение к Гауссовскому окну создайте окно Хемминга тех же размеров. Дополните окно нулями и напечатайте его ДПФ. Какое окно больше подходит для фильтра НЧ? Полезно напечатать ДПФ с логарифмическим масштабом `y`

Поэкспериментируйте с разными окнами и размерами этих окон.

(b) Ход работы

Создадим волну с частотой дискретизации $= 44.1$ КГц

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

Листинг 6: Создаем волну

Теперь создадим несколько окон со стандартным отклонением Гаусса.

```
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 7: Создаем окна

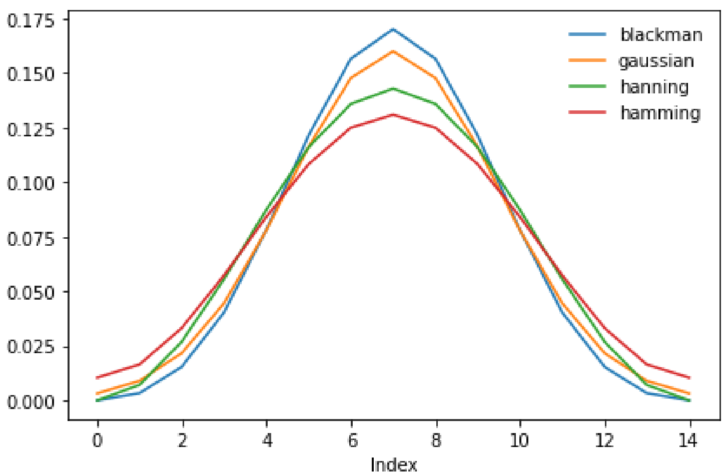

Рис. 10: Окна

Видим, что все окна похожи друг на друга. Теперь применим

ДПФ

```
def zero_pad(array, n):
res = np.zeros(n)
res[:len(array)] = array
return res

def plot_window_dfts(windows, names):
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)

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

Листинг 8: Применяем ДПФ

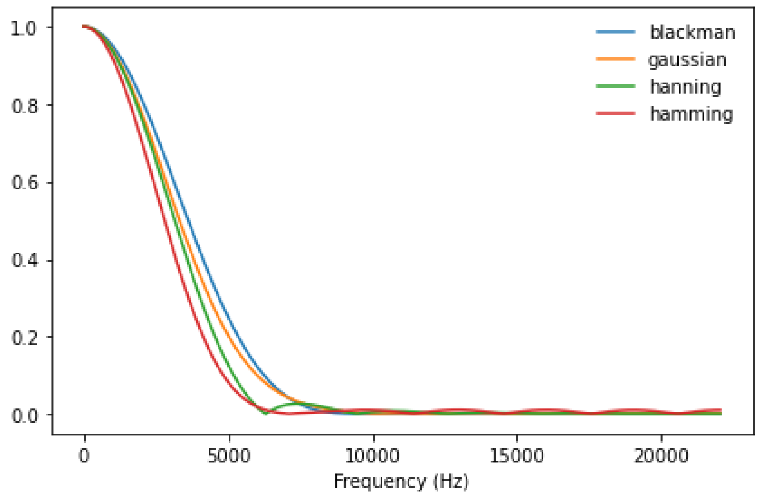

Рис. 11: Окна после ДПФ

Хэмминг падает быстрее всех. Блэкмэн падает медленней всех. У Хэннинга самые заметные отскоки.

Теперь посмотрим на ДПФ с логарифмическим масштабом

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

Листинг 9: ДПФ с log-масштабом

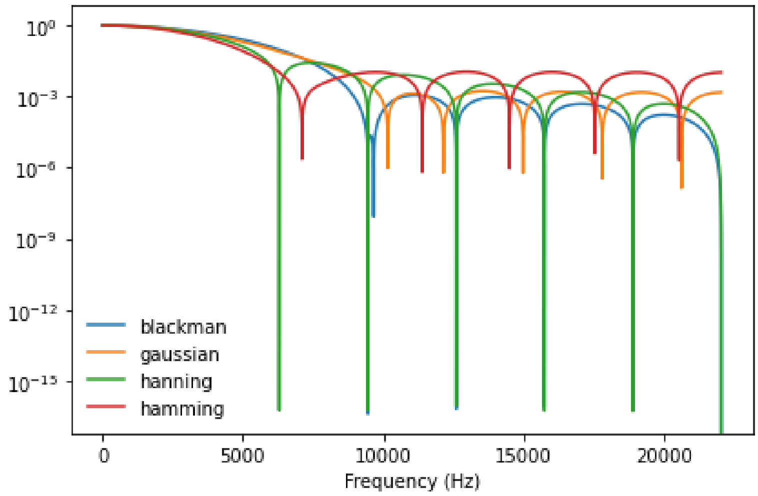

Рис. 12: ДПФ с log-масштабом

Хэмминг и Хэннинг падают быстрее остальных. Хэмминг и Гаусс имеют скачки. В итоге можно сказать, что окно Хэннинга имеет минимальные отскоки в сочетании с быстрым падением.

4 Вывод

В результате выполнения лабораторной работы получены навыки работы со свертками, построением различных окон и применения к ним ДПФ.