

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Цифровая обработка сигналов»
Тема: Частотный анализ полиномиальных приближений

Студенты гр. 8383

Ларин А.

Бобенко Н. С.

Преподаватель

Середа А. И.

Санкт-Петербург

2021

Цель работы.

Получение практических навыков выполнения фильтрации дискретных последовательностей с помощью рекурсивных и нерекурсивных фильтров, а также анализа получаемых результатов с помощью дискретного преобразования Фурье.

Основные теоретические положения.

Лабораторная работа потребует знаний:

- в области дискретизации непрерывного сигнала;
- фильтрации дискретного сигнала с помощью дискретных нерекурсивных и рекурсивных фильтров;
- дискретного преобразования Фурье (ДПФ) для дискретных последовательностей;

и умений:

- в организации вычислительных процессов;
- в проведении компьютерных расчетов с визуализацией получаемых результатов;
- проведения анализа полученных результатов и формулировка выводов.

Постановка задачи.

Для заданного дискретного сигнала применить соответствующие фильтры. Полученные результаты содержательно проинтерпретировать.

Порядок выполнения работы.

1. Сформировать дискретный сигнал посредством дискретизации с шагом $T=1$ непрерывного сигнала, представляющего собой линейную комбинацию косинусоид вида $A_k \cos(\omega_k t + \varphi_k)$. Частота каждой из гармоник не должна превышать π . Всего одиннадцать гармоник с упорядоченными по возрастанию частотами от 0 до π ,

изменяющимися с шагом $\Delta\omega=0.1\pi$. Амплитуды гармоник A_k представляют собой целые числа со значениями от 1 до 11, определяемые случайным образом с помощью датчика равномерно распределенных случайных чисел. При необходимости нормализовать коэффициенты линейной комбинации посредством деления их на сумму полученных случайным образом амплитуд. Начальные фазы φ_k представляют собой случайные числа в промежутке от 0 до 0.5. Дискретная последовательность должна включать в себя 32 отсчета ($N=31$).

2. Визуализировать исходные аналоговый и дискретизированный сигналы.
3. С помощью ДПФ найти дискретные отсчеты спектра дискретного сигнала и визуализировать их.
4. Для дискретного сигнала применить линейное сглаживание по 5-ти и 9-ти точкам, представить формулу для $H(\omega)$ – передаточной функции (частотной характеристики) фильтра.
5. Визуализировать полученный после фильтрации дискретный сигнал совместно с исходным дискретным сигналом.
6. С помощью ДПФ найти дискретные отсчеты спектра дискретного сигнала после его фильтрации и визуализировать их совместно с отсчетами спектра исходного дискретного сигнала.
7. Проанализировать результат на соответствие значениям $H(\omega)$. Сделать выводы.
8. Повторить пункты 4)-7) для следующих фильтров:
 - a. Сглаживание полиномом 2-ой степени по 5 и 9 узлам.
 - b. Сглаживание полиномом 4-ой степени по 7 и 11 узлам.
 - c. Дискретный фильтр, соответствующий численному дифференцированию 1-го порядка.
 - d. Дискретный фильтр, соответствующий численному интегрированию (прямоугольников, трапеций, Симпсона).
9. Содержательно проинтерпретировать результаты выполнения лабораторной работы, сделать выводы.

Выполнение работы.

1. Сформирован дискретный сигнал как линейная комбинация косинусойд с частотами 0.1π , 0.2π .. π . Начальные фазы и амплитуды сгенерированы случайно. Амплитуды сигнала на рис 1.1. Сигнал дискретизирован с шагом $T=1$.
2. Исходный сигнал представлен на рис. 2.1. Дискретизированный сигнал на рис. 2.2.

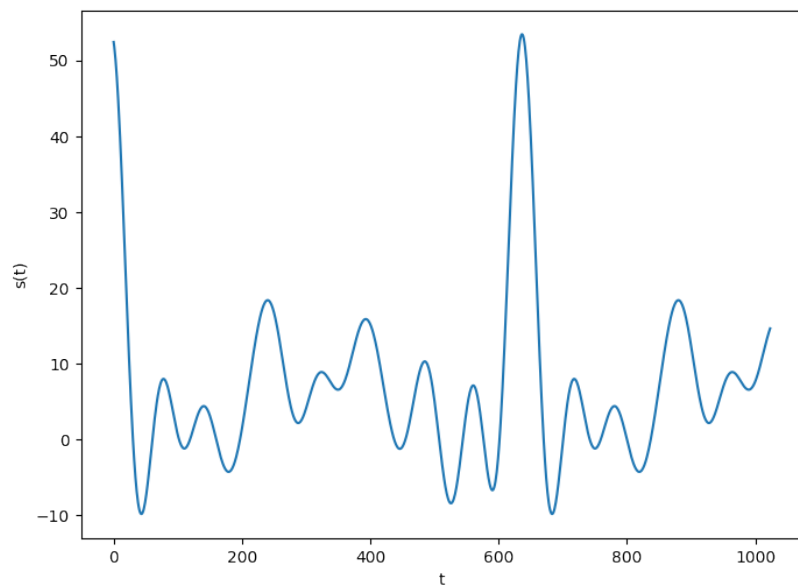


Рисунок 2.1 – Сгенерированный сигнал

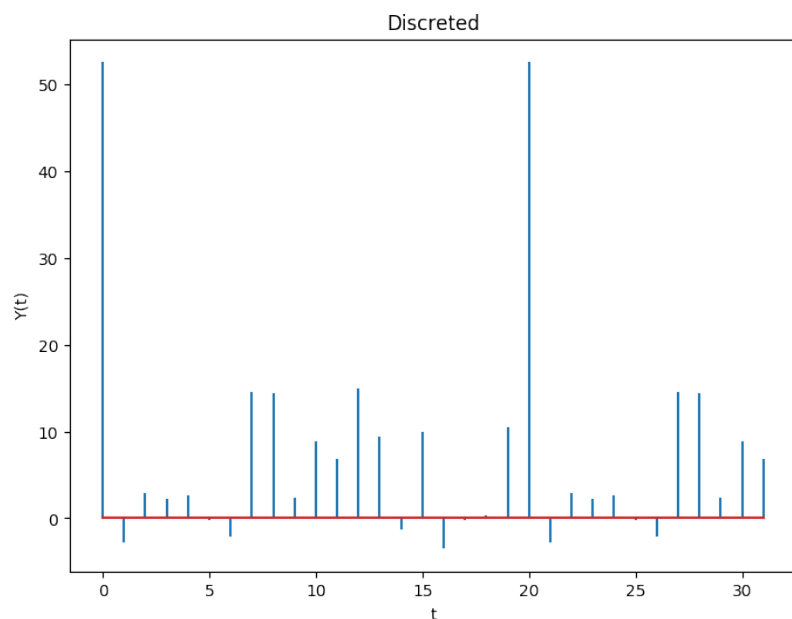


Рисунок 2.2 – Дискретизированный сигнал

3. Т.к. сигнал периодичен от него можно посчитать преобразование фурье. С помощью дискретного преобразования фурье найдены дискретные отсчеты спектра дискретного сигнала.

Вещественная часть сигнала представлена на рис. 3.1

Мнимая часть на рис 3.2.

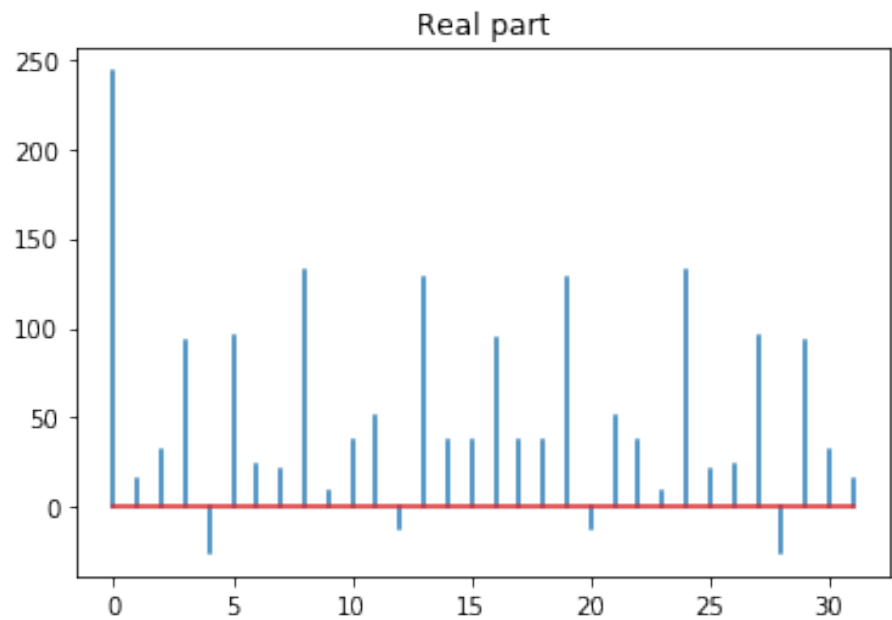


Рисунок 3.1 – Вещественная часть спектра

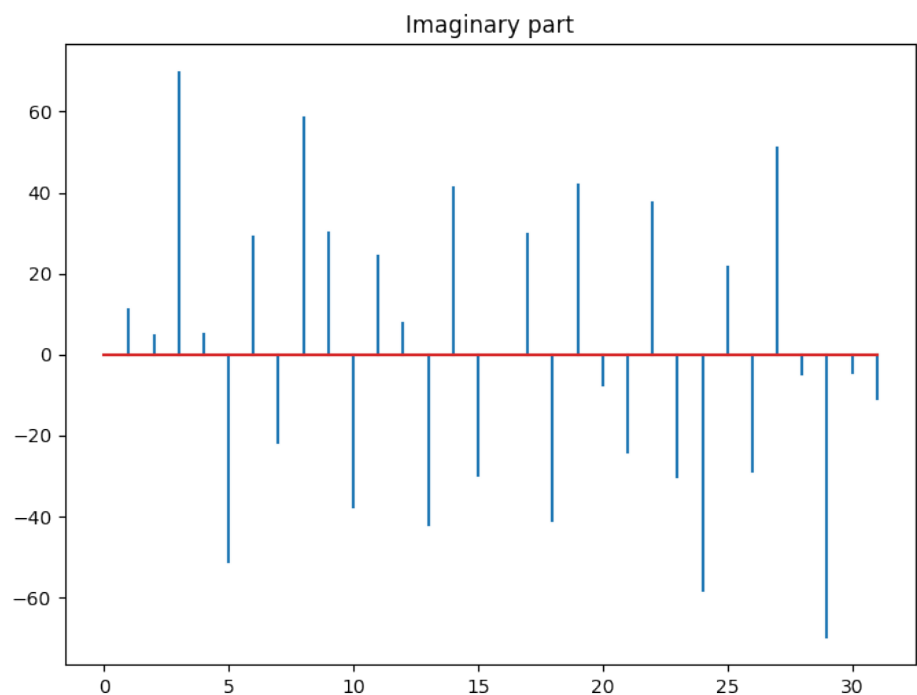


Рисунок 3.2 – Мнимая часть спектра

4. Для дискретного сигнала применено линейное сглаживание по пяти и девяти точкам. %omega

$$H_5(\omega) = \frac{1}{5}(1 + 2\cos \omega + 2\cos 2\omega)$$

$$H_9(\omega) = \frac{1}{5}(1 + 2\cos \omega + 2\cos 2\omega + 2\cos 3\omega + 2\cos 4\omega)$$

Формула для линейного сглаживания по 5 точкам и 9 точкам:

$$y_{n,5} = \frac{1}{5} \sum_{k=n-2}^{n+2} x_k$$

$$y_{n,9} = \frac{1}{9} \sum_{k=n-4}^{n+4} x_k$$

5. Сглаженные сигналы изображены на рис.4.1,2 с исходным (синий и красный соотв.)

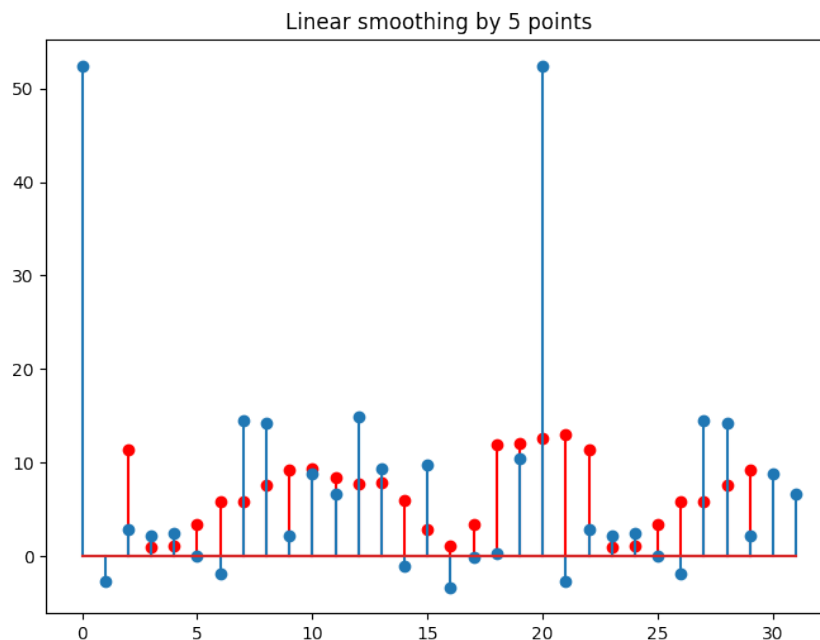


Рисунок 5.1 – Сглаженный сигнал по пяти точкам

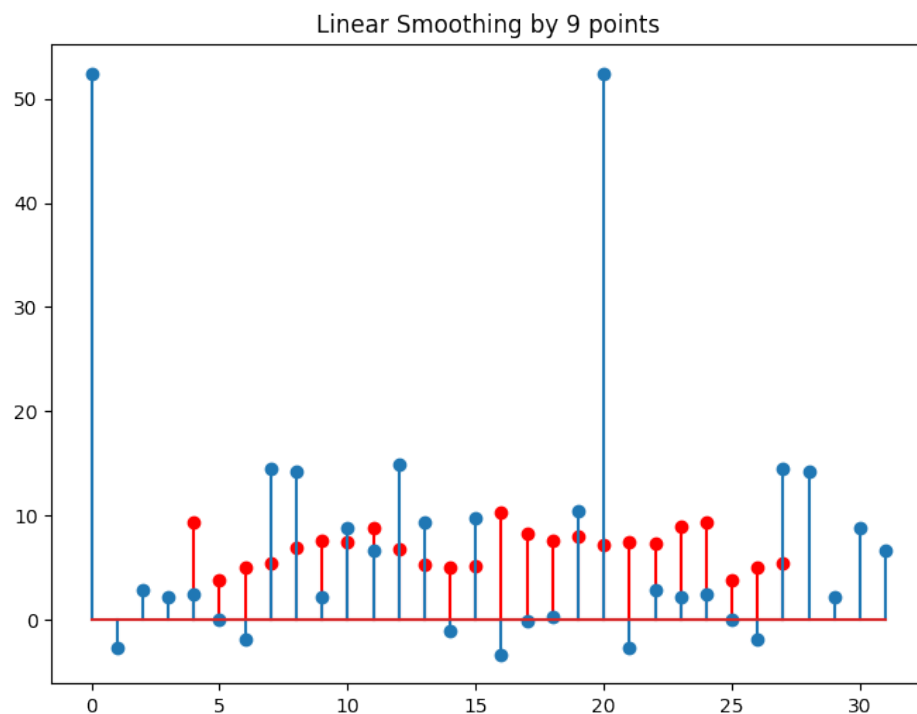


Рисунок 5.2 – Сглаженный сигнал по девяти точкам

6. с помощью ДПФ найдены дискретные спектры сигнала

Спектры для сглаживания по пяти точкам представлены на рис 6.2,6.2, для девяти точке на рис 6.3,6.4

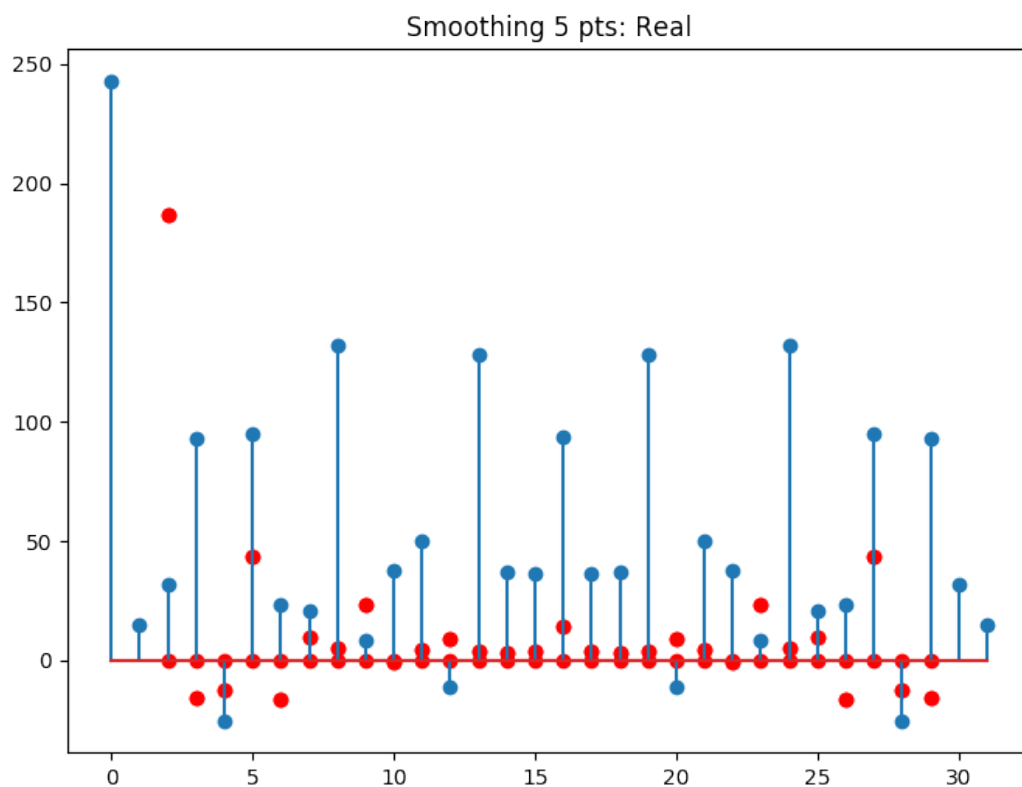


Рисунок 6.1 – Вещественная часть спекта(сигнал сгл. по 5 точкам)

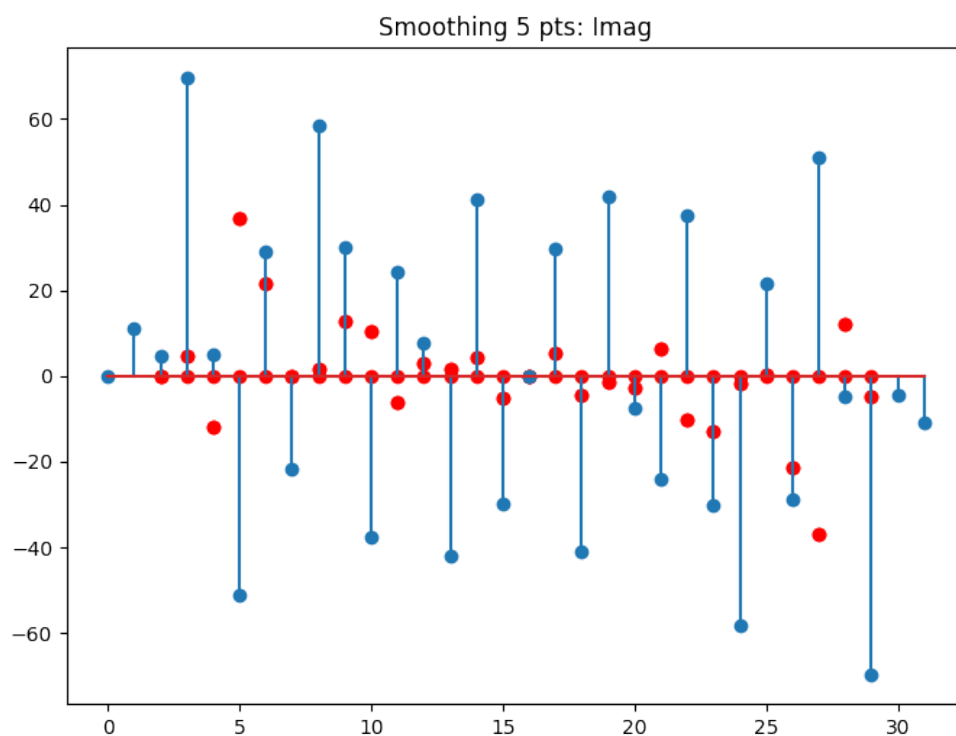


Рисунок 6.2 – Мнимая часть спекта(сигнал сгл. по 5 точкам)

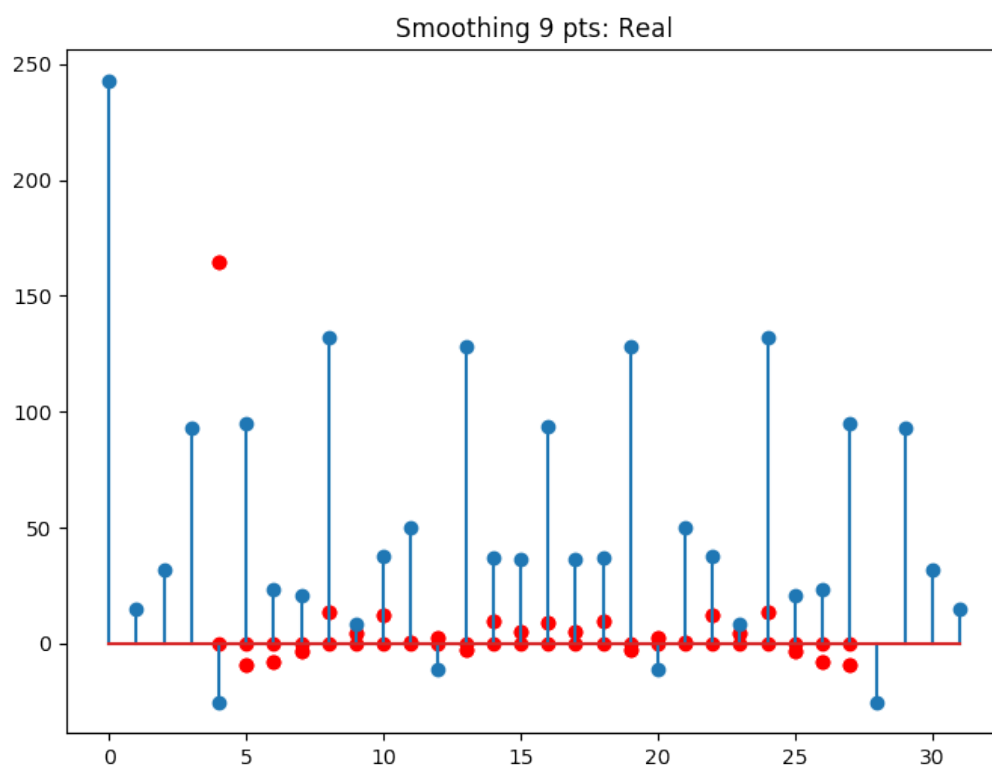


Рисунок 6.3 – Вещественная часть спекта(сигнал сгл. по 9 точкам)

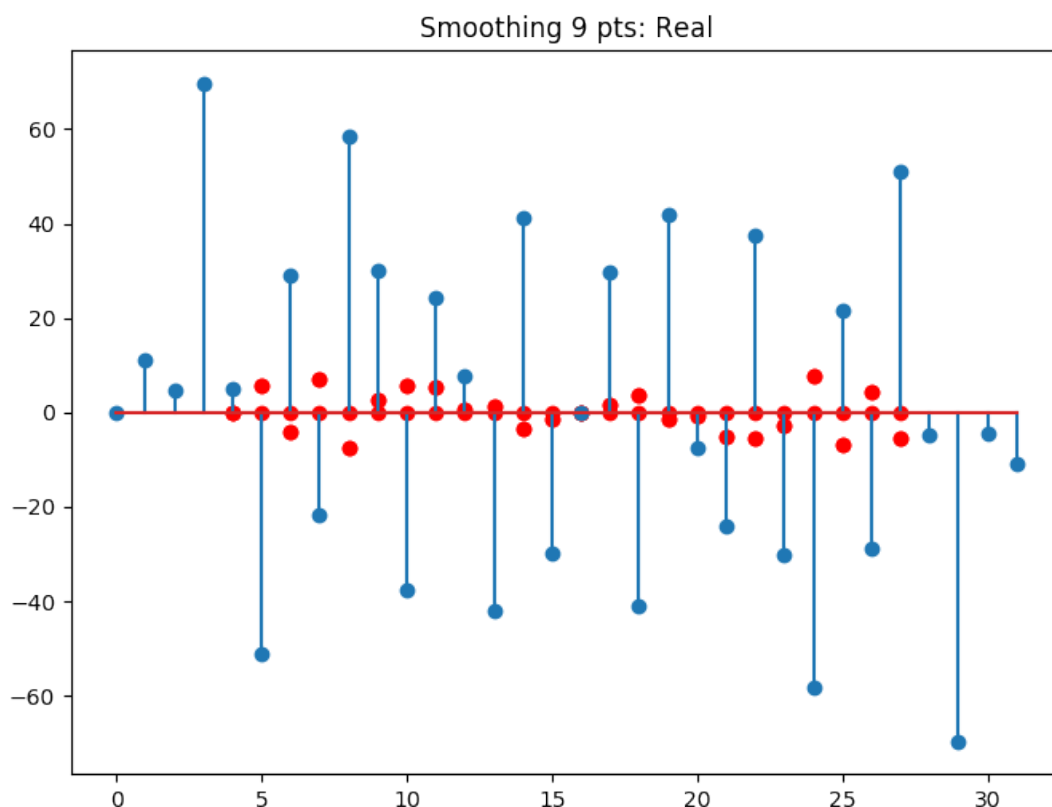


Рисунок 6.4 – Мнимая часть спекта(сигнал сгл. по 9 точкам)

7. По спектру сглаженного сигнала можно сказать, что при более сильном сглаживании искажаются всё более низкие частоты, а также уменьшается амплитуда.

8. Пункты 4-7 для других фильтров

а. Сглаживание полиномом 2-ой степени по 5 и 9 узлам.

$$y_5(n) = \frac{1}{35} (-3x_{k-2} + 12x_{k-1} + 17x_k + 12x_{k+1} - 3x_{k+2})$$

$$y_9(n) = \frac{1}{231} (-21x_{k-4} + 14x_{k-3} + 39x_{k-2} + 54x_{k-1} + 59x_k + 54x_{k+1} + 39x_{k+2} + 14x_{k+3} - 21x_{k+4})$$

Формула для передаточной функции $H(\omega)$

$$H_5(\omega) = \frac{1}{35} (17 + 24 \cos \omega - 6 \cos 2 \omega)$$

$$H_9(\omega) = \frac{1}{231} (59 + 108 \cos \omega + 78 \cos 2 \omega + 28 \cos 3 \omega - 42 \cos 4 \omega)$$

Сигнал после фильтрации по пяти и девяти точкам на рис.8.1.1-2

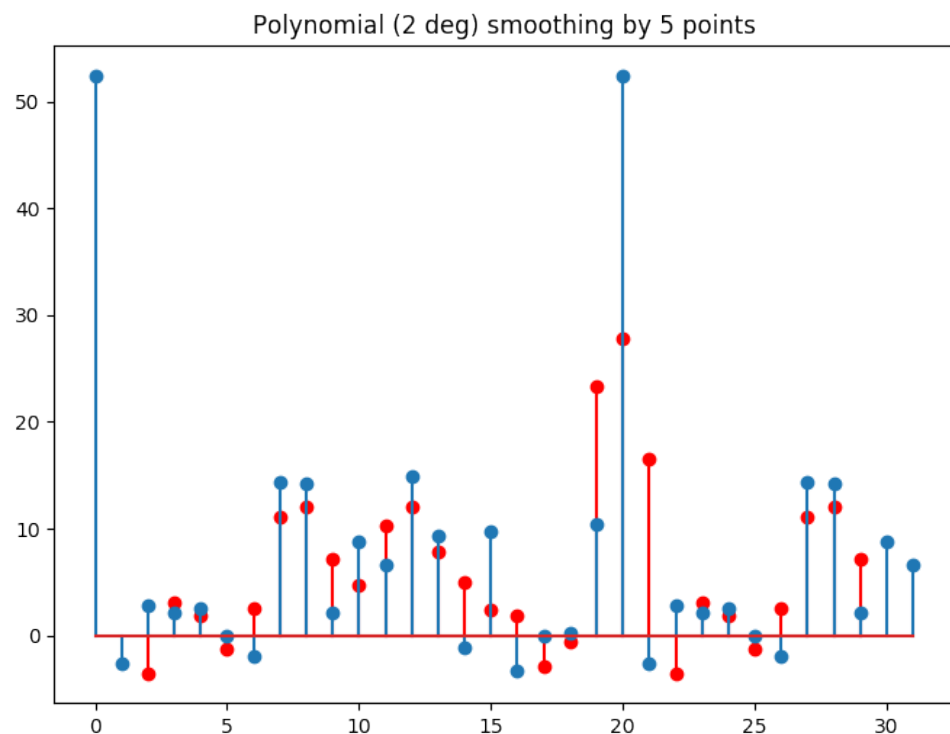


Рисунок 8.1.1 – Сигнал сглаженный полиномом 2-й степени по 5 точкам

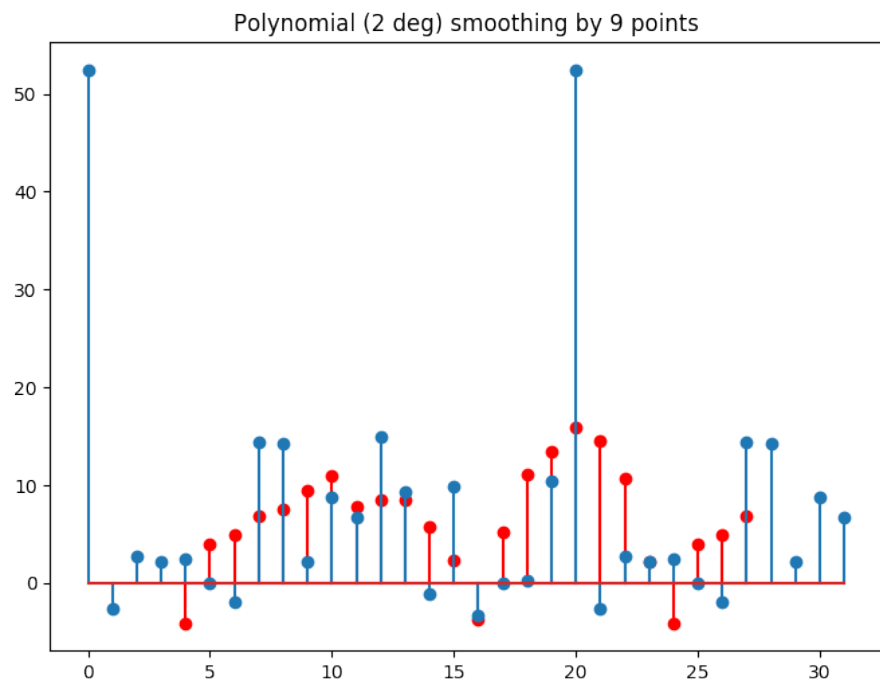


Рисунок 8.1.2 – Сигнал сглаженный полиномом 2-й степени по 9 точкам

Дискретный спектр для сглаживания по пяти и девяти точкам представлены на рис.8.1.3-6

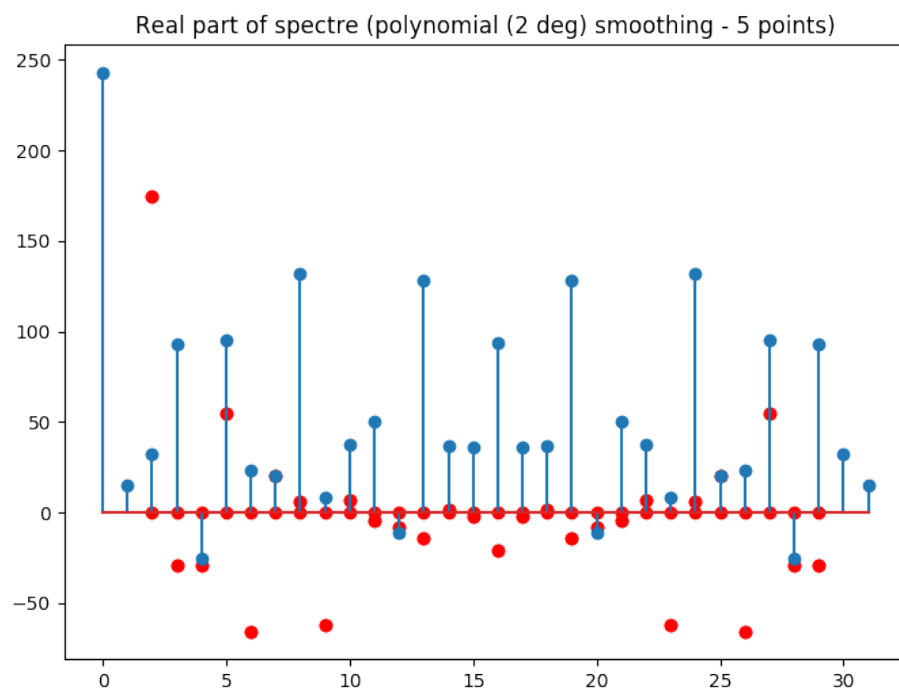


Рисунок 8.1.3 – Спектр после фильтрации, вещественная часть
(5 точек)

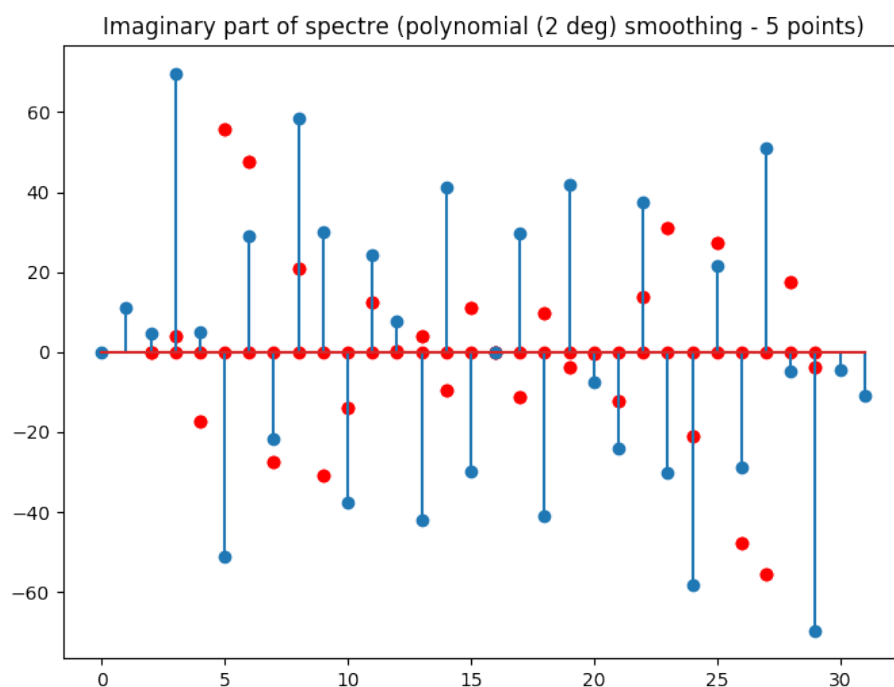


Рисунок 8.1.4 – Спектр после фильтрации, мнимая часть (5
точек)

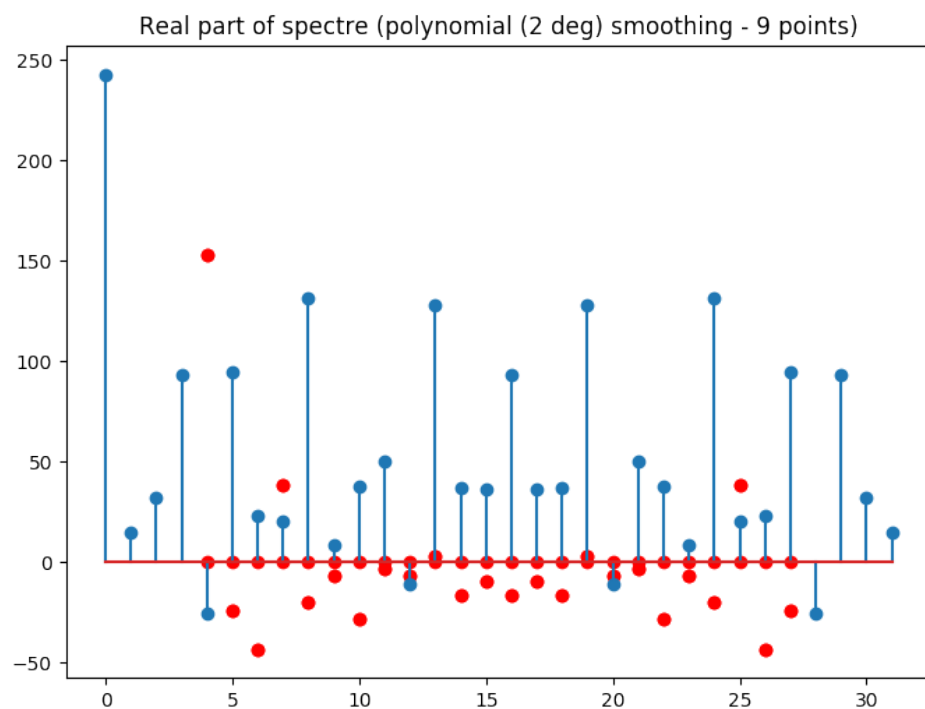


Рисунок 8.1.5 – Спектр после фильтрации, вещественная часть
(9 точек)

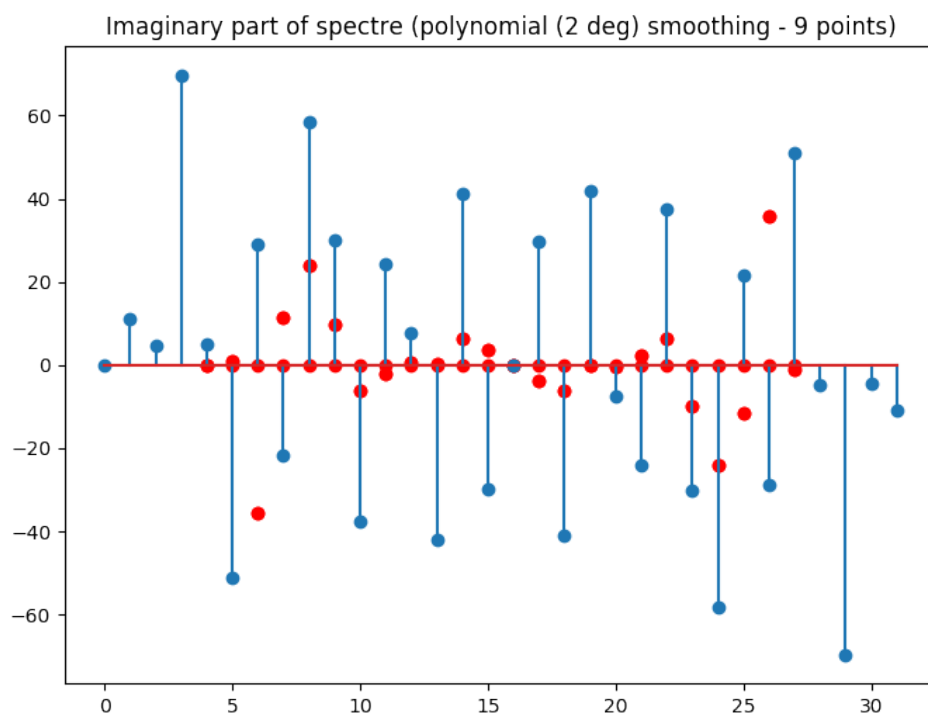


Рисунок 8.1.6 – Спектр после фильтрации, мнимая часть (9 точек)

б. Сглаживание полиномом четвёртой степени по 7 и 11 узлам:

$$y_7(n) = \frac{1}{231} (5x_{k-3} - 30x_{k-2} + 75x_{k-1} + 131x_k + 75x_{k+1} - 30x_{k+2} + 5x_{k+3})$$

$$y_{11}(n) = \frac{1}{429} (13x_{k-5} - 45x_{k-4} - 10x_{k-3} + 60x_{k-2} + 120x_{k-1} + 143x_k + 120x_{k+1} + 60x_{k+2} - 10x_{k+3} - 45x_{k+4} + 13x_{k+5})$$

Формула передаточной функции $H(\omega)$

$$H_7(\omega) = \frac{1}{231} (131 + 150 \cos \omega - 60 \cos 2\omega + 10 \cos 3\omega)$$

$$H_{11}(\omega) = \frac{1}{429} (143 + 240 \cos \omega + 120 \cos 2\omega - 20 \cos 3\omega - 90 \cos 4\omega + 26 \cos 5\omega)$$

Сигнал после фильтрации по 7-ми и 11-ти точкам на рис.8.2.1-2

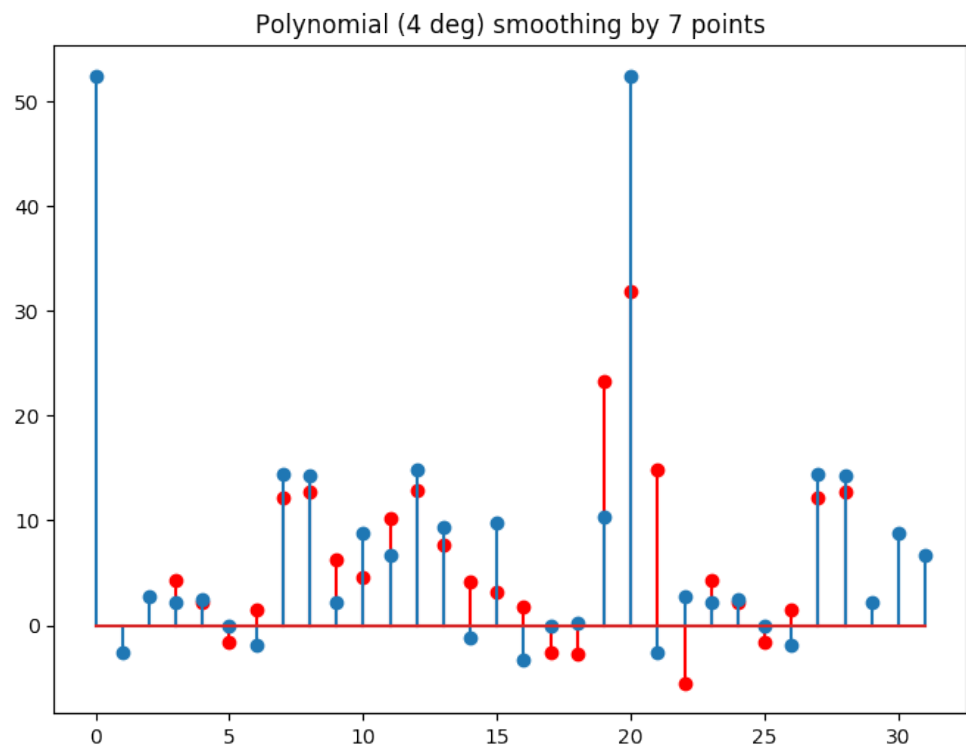


Рисунок 8.2.1 – Сигнал сглаженный полиномом 4-й степени по 7-ми точкам

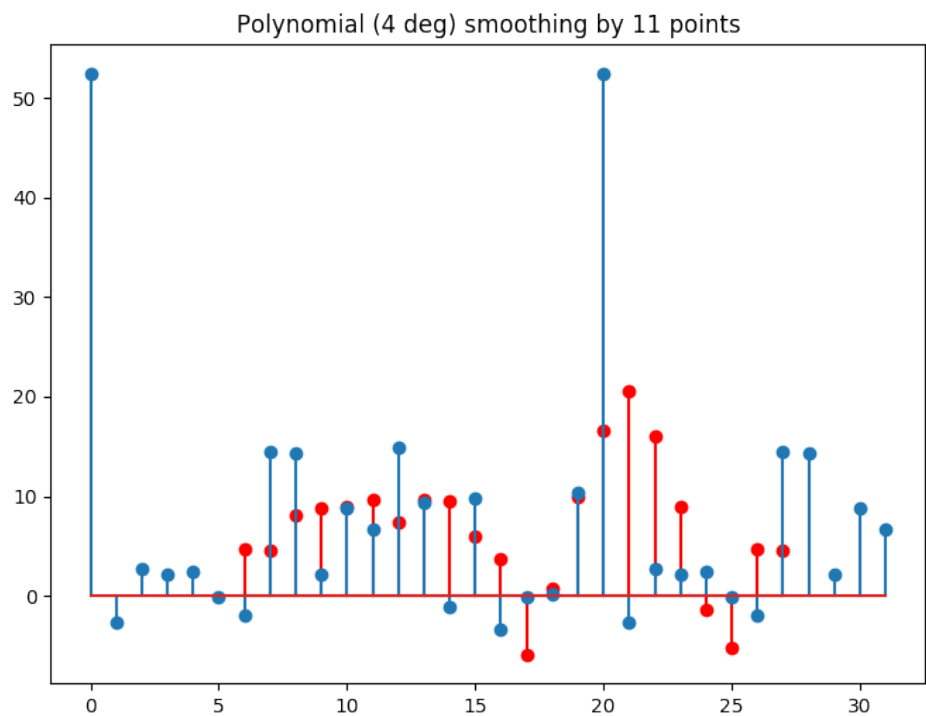


Рисунок 8.2.2 – Сигнал сглаженный полиномом 4-й степени по 11-ти точкам

Дискретный спектр для сглаживания по пяти и девяти точкам представлены на рис.8.2.3-6

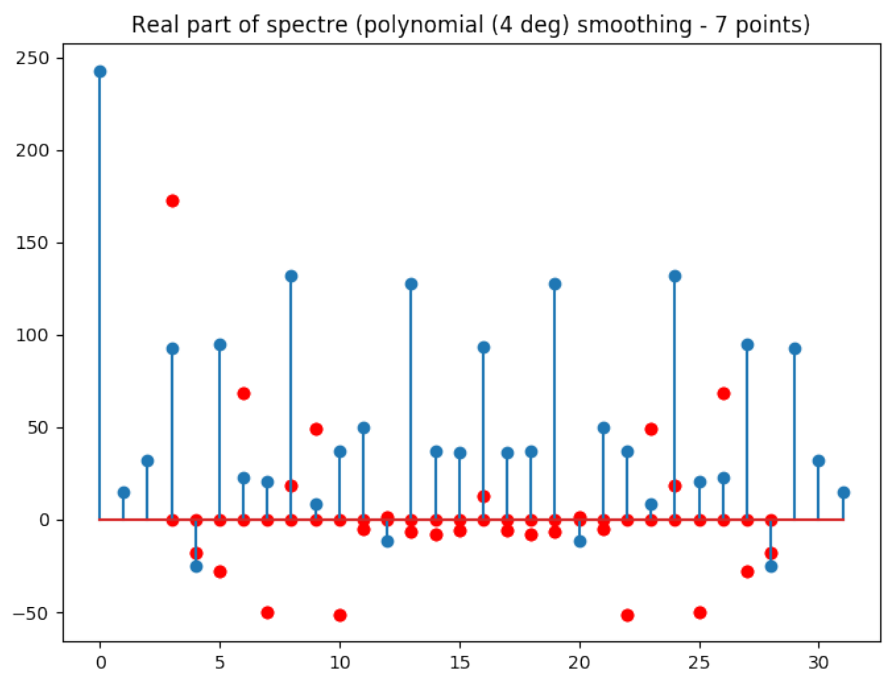


Рисунок 8.2.3 – Спектр после фильтрации, вещественная часть (7 точек)

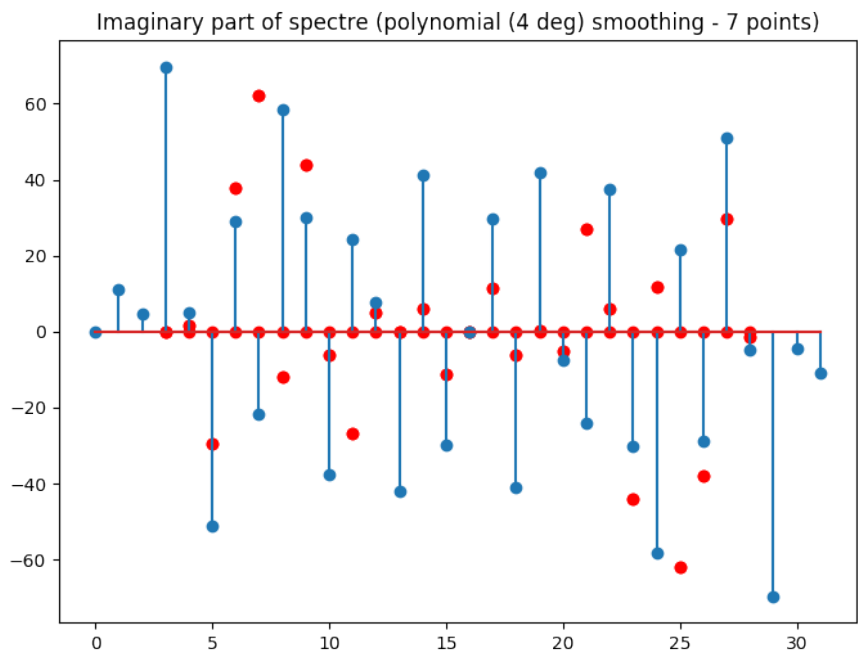


Рисунок 8.2.4 – Спектр после фильтрации, мнимая часть (7 точек)

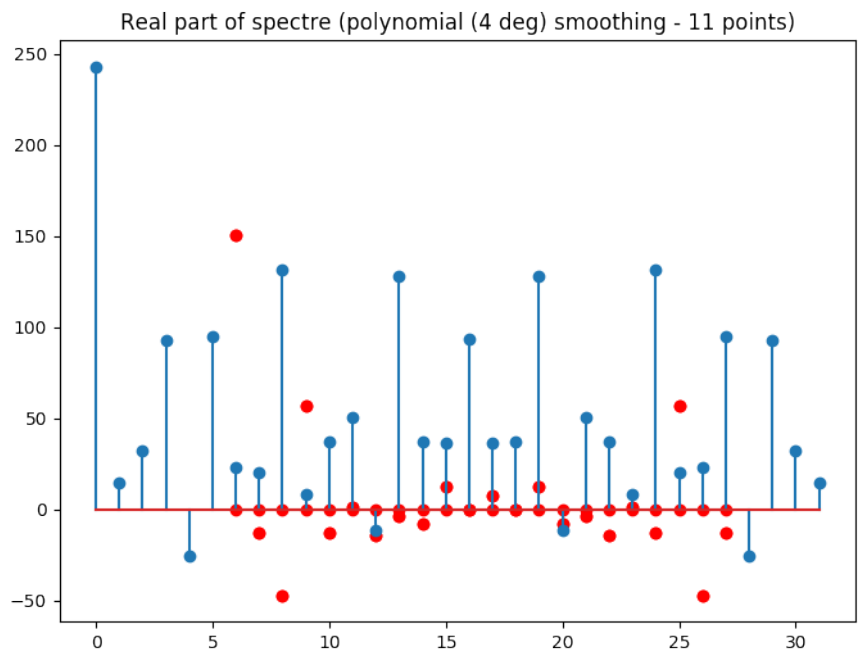


Рисунок 8.2.5 – Спектр после фильтрации, вещественная часть (11 точек)

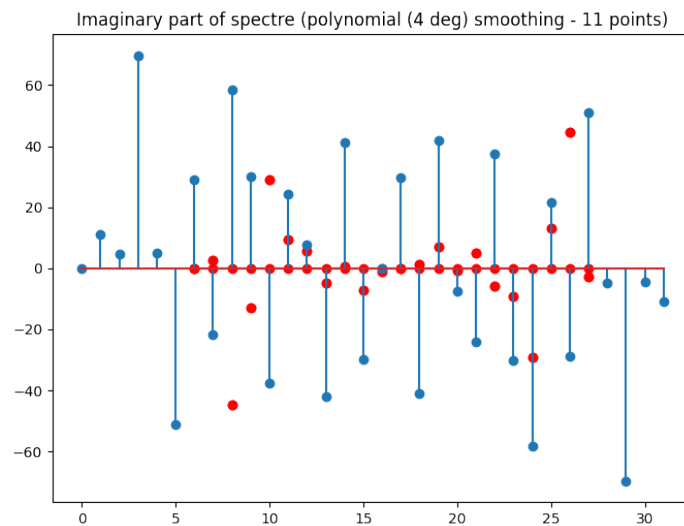


Рисунок 8.2.6 – Спектр после фильтрации, мнимая часть (11 точек)

с. Дискретный фильтр, соответствующий численному дифференцированию 1-го порядка:

$$y'(n) = \frac{1}{2}(x_{k+1} - x_{k-1})$$

Формула передаточной функции $H(\omega)$

$$H(\omega) = i \sin \omega$$

Сигнал после фильтрации представлен на рис.8.3.1

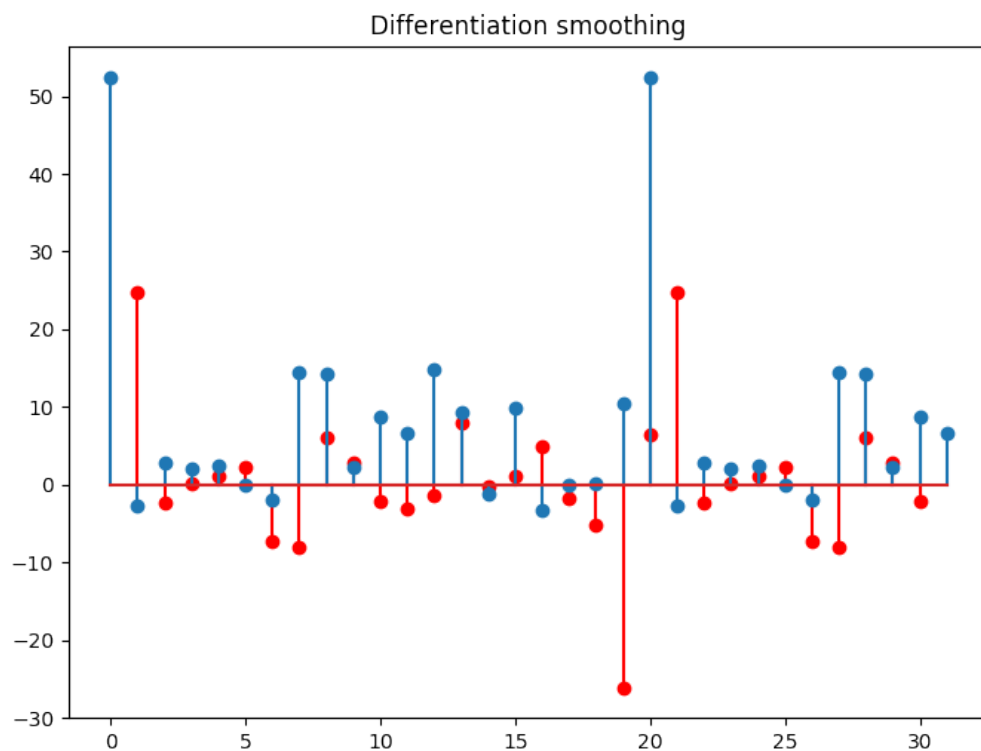


Рисунок 8.3.1 – Сигнал сглаженный дискретным фильтром, соответствующий численному дифференцированию 1-го порядка.

Дискретный спектр для сглаживания по пяти и девяти точкам представлены на рис.8.3.2-3

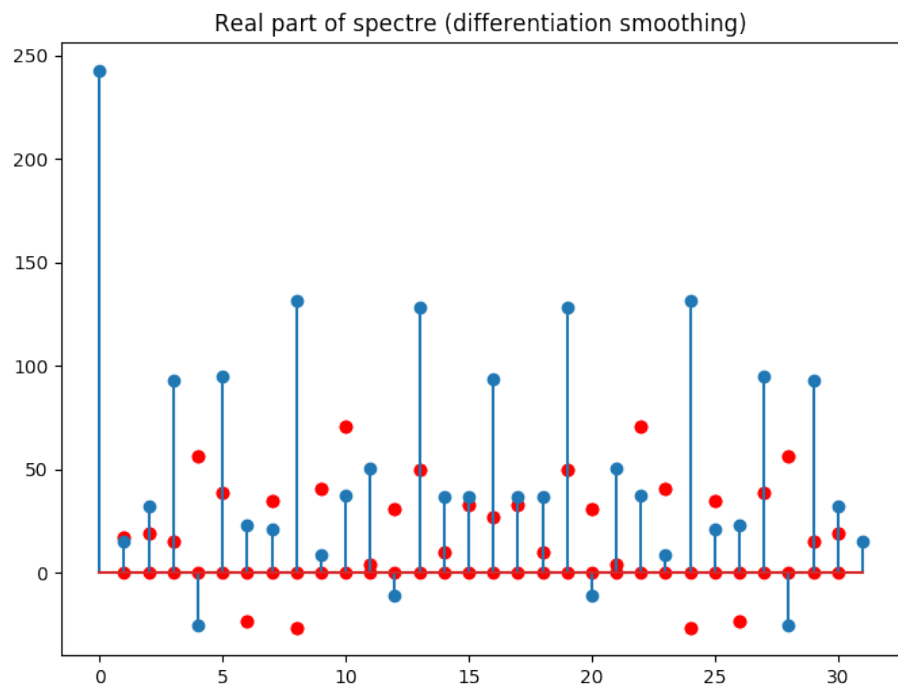


Рисунок 8.3.2 – Спектр после применения фильтра соответствующего численному дифференцированию, вещественная часть

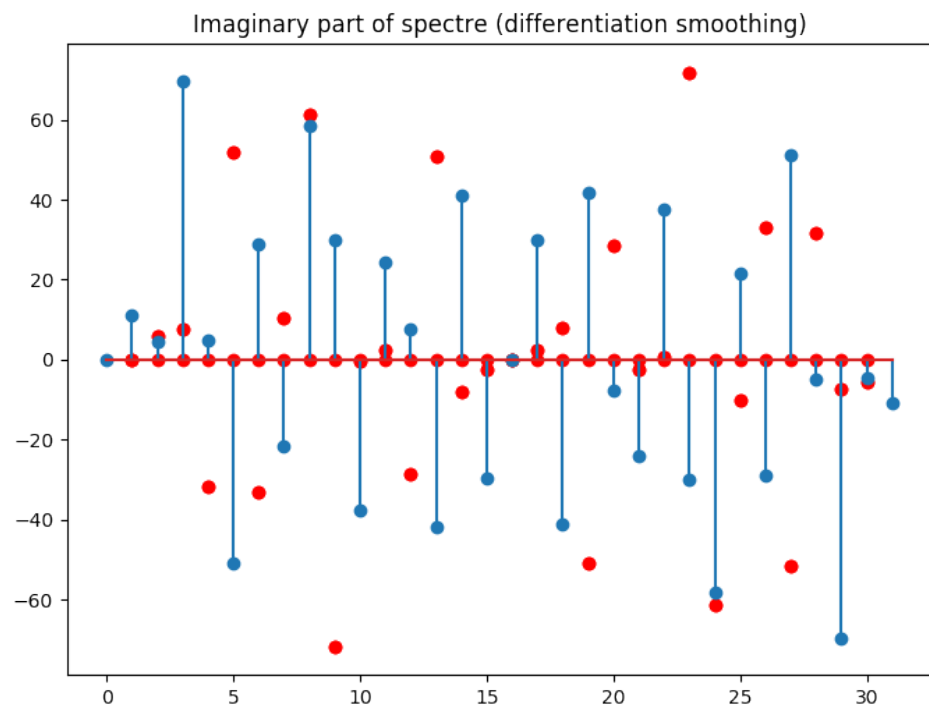


Рисунок 8.3.3 – Спектр после применения фильтра соответствующего численному дифференцированию, вещественная часть

d. Дискретный фильтр, соответствующий численному интегрированию (прямоугольников $y_1(n)$, трапеций $y_2(n)$, Симпсона $y_3(n)$):

$$y_1(n) = y_1(n-1) + x_{2n+1}$$

$$y_2(n) = y_2(n-1) + \frac{1}{2}(x_n + x_{n+1})$$

$$y_3(n+1) = y_3(n-1) + \frac{1}{3}(x_{n-1} + 4x_n + x_{n+1})$$

Формула передаточной функции $H(\omega)$:

$$H_1(\omega) = \frac{1}{2i \sin \frac{\omega}{2}}$$

$$H_2(\omega) = \frac{\cos \frac{\omega}{2}}{2i \sin \frac{\omega}{2}}$$

$$H_3(\omega) = \frac{(\cos \omega + 2)}{3i \sin \omega}$$

Сигнал после фильтрации представлены на рис.8.4.1-3

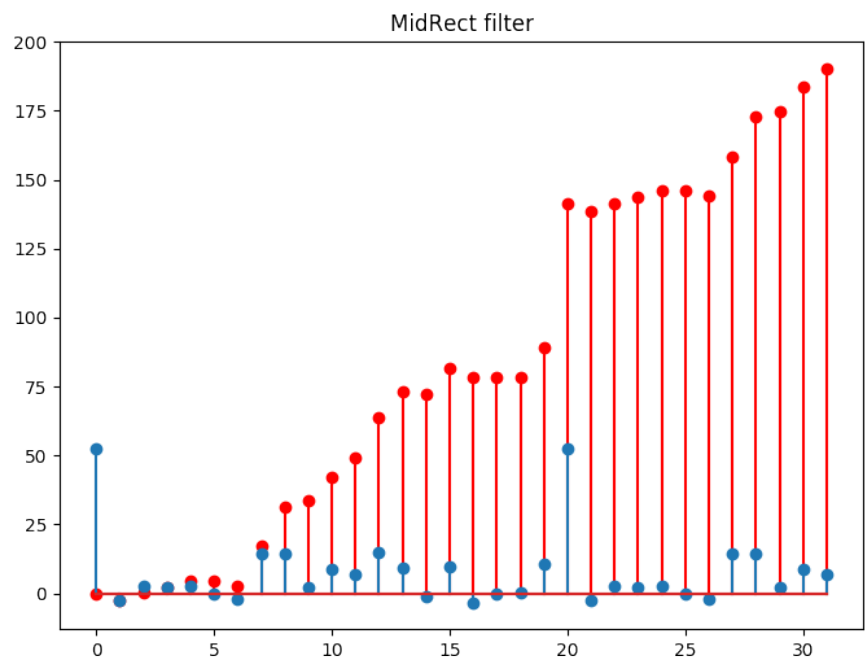


Рисунок 8.4.1 – Сигнал после фильтра, соответствующего формуле средних прямоугольников

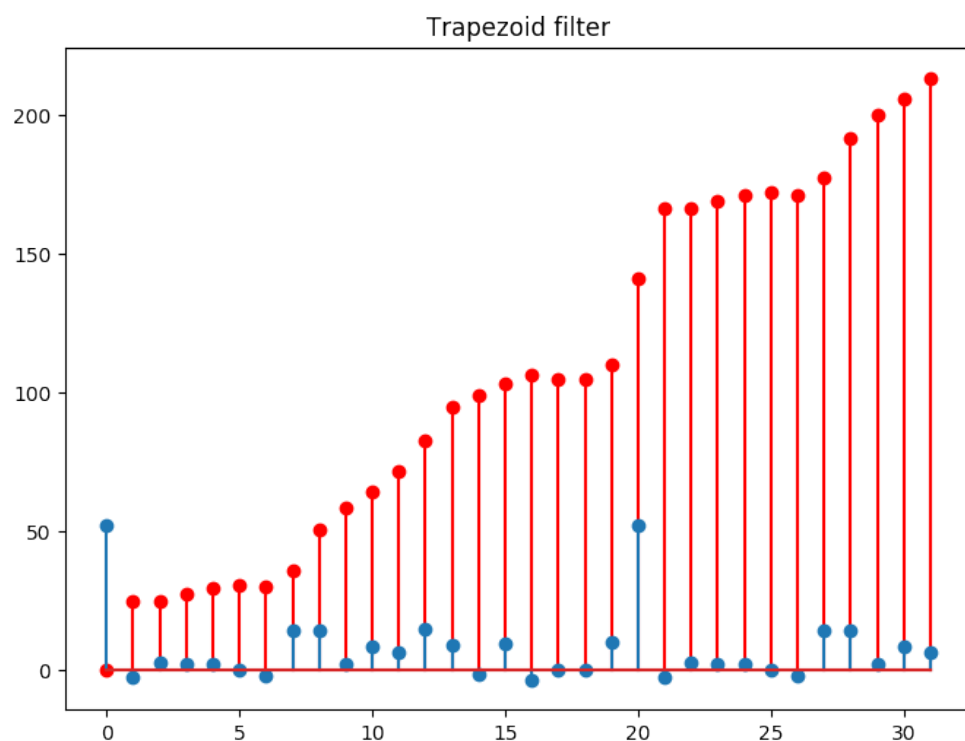


Рисунок 8.4.2 – Сигнал после фильтра, соответствующего формуле трапеций

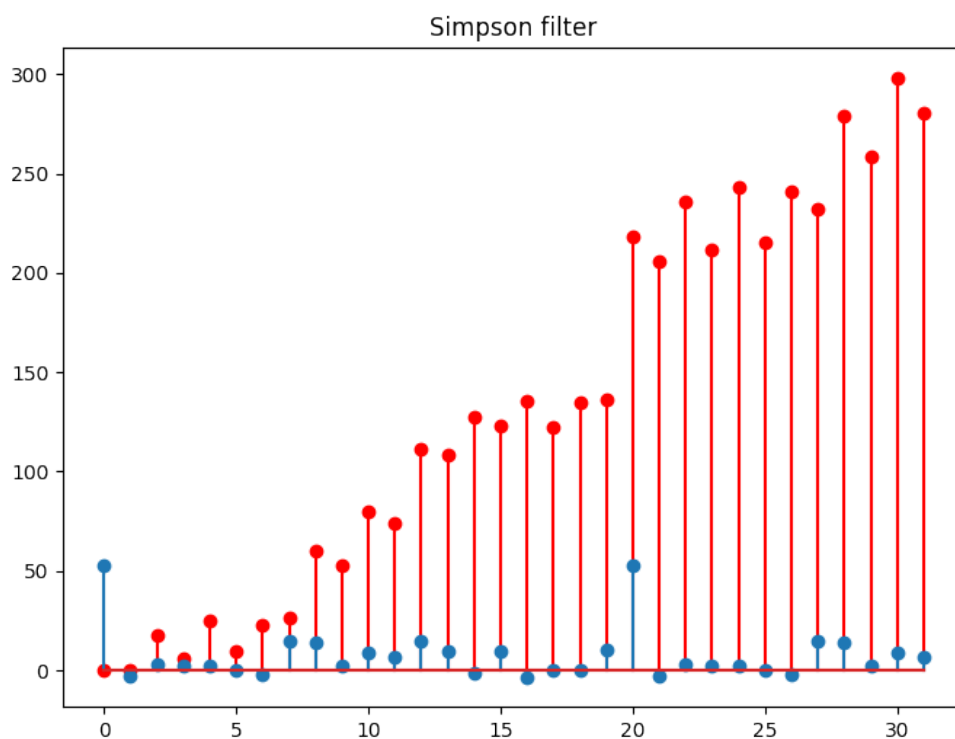


Рисунок 8.4.3 – Сигнал после фильтра, соответствующего формуле Симпсона

Дискретный спектр для отфильтрованных сигналов представлены на рис.8.4.4-9

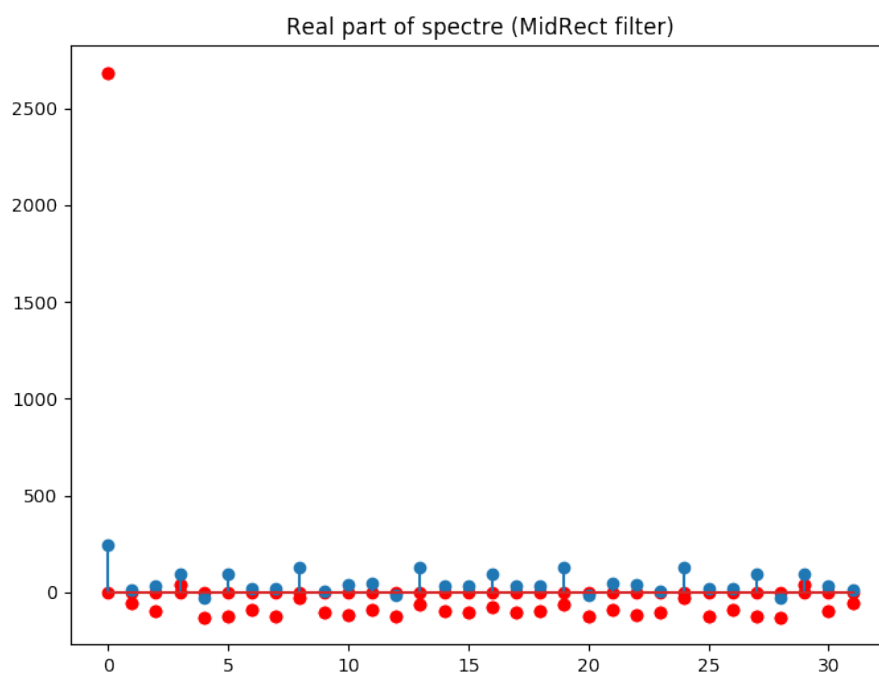


Рисунок 8.4.4 – Спектр после фильтрации, вещественная часть
(формула средних прямоугольников)

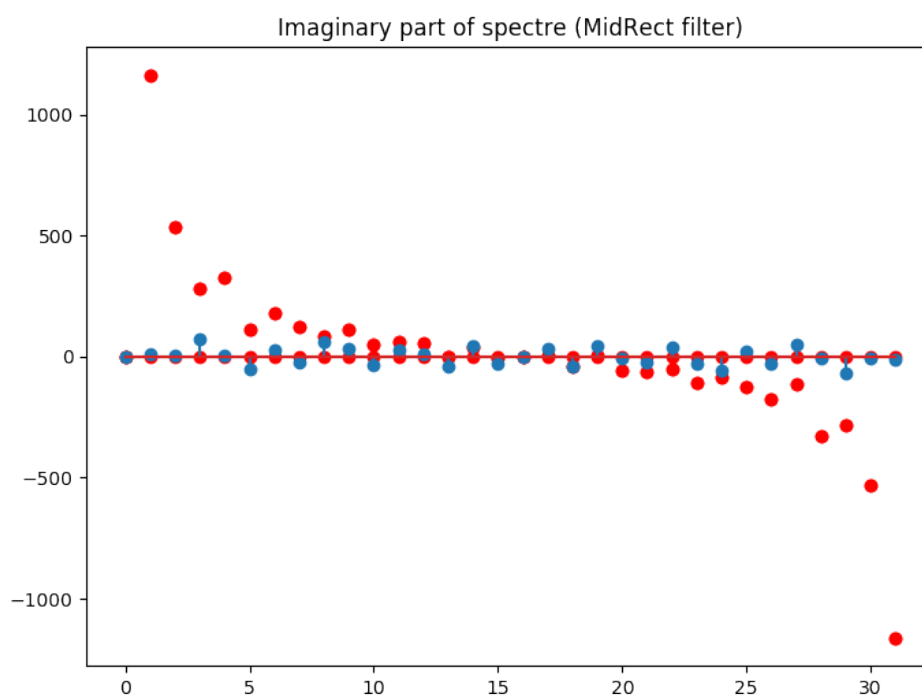


Рисунок 8.4.5 – Спектр после фильтрации, мнимая часть
(формула средних прямоугольников)

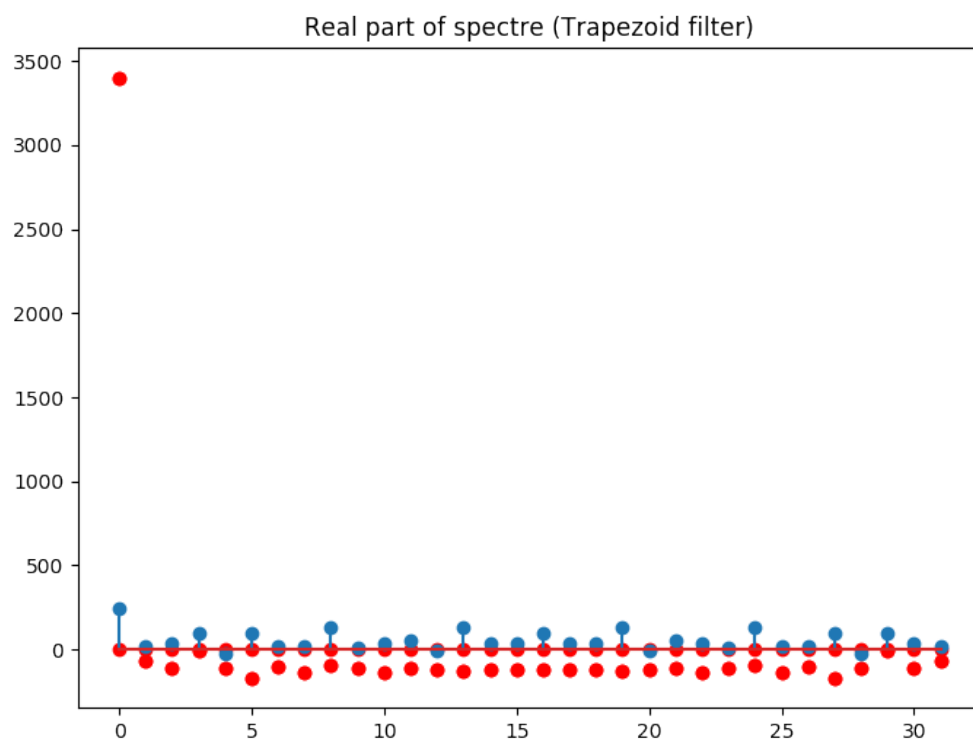


Рисунок 8.4.6 – Спектр после фильтрации, вещественная часть
(формула трапеций)

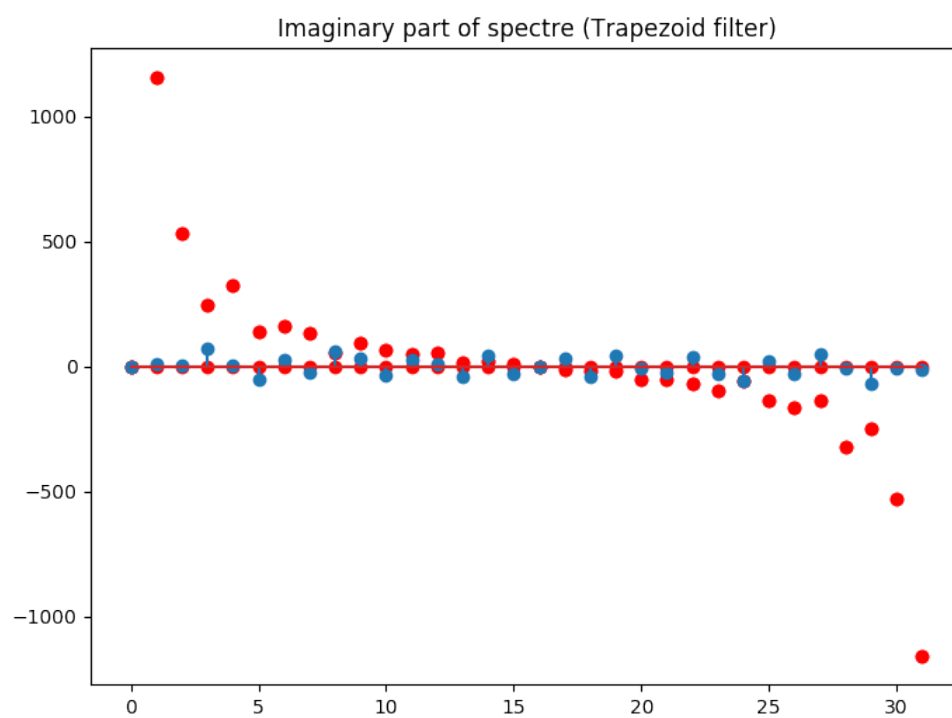


Рисунок 8.4.7 – Спектр после фильтрации, мнимая часть
(формула трапеций)

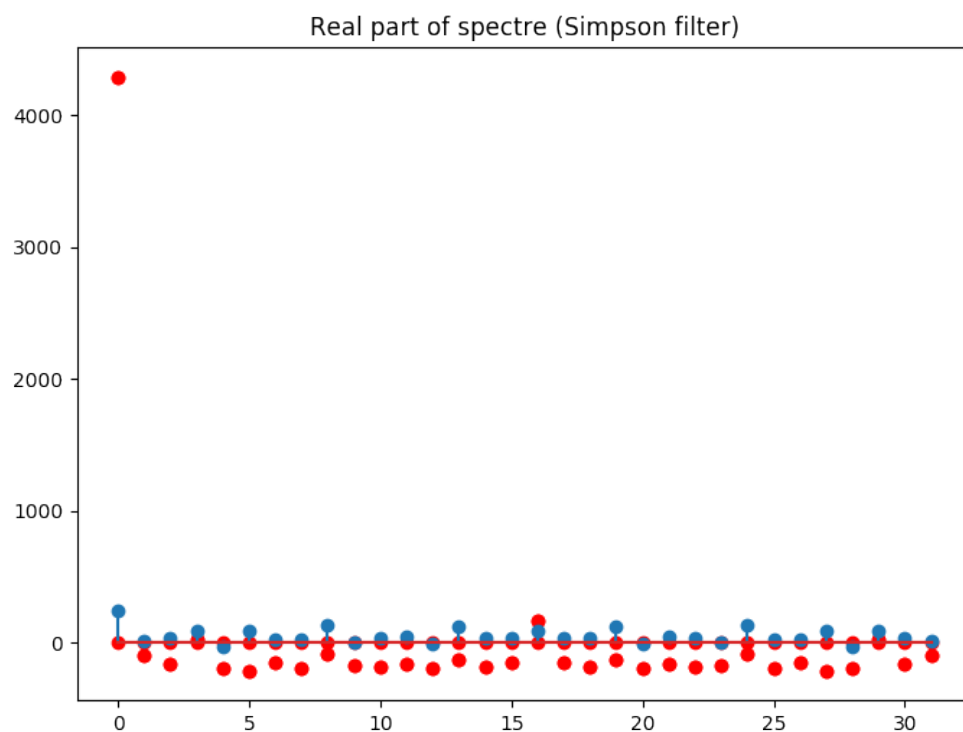


Рисунок 8.4.8 – Спектр после фильтрации, вещественная часть
(формула Симпсона)

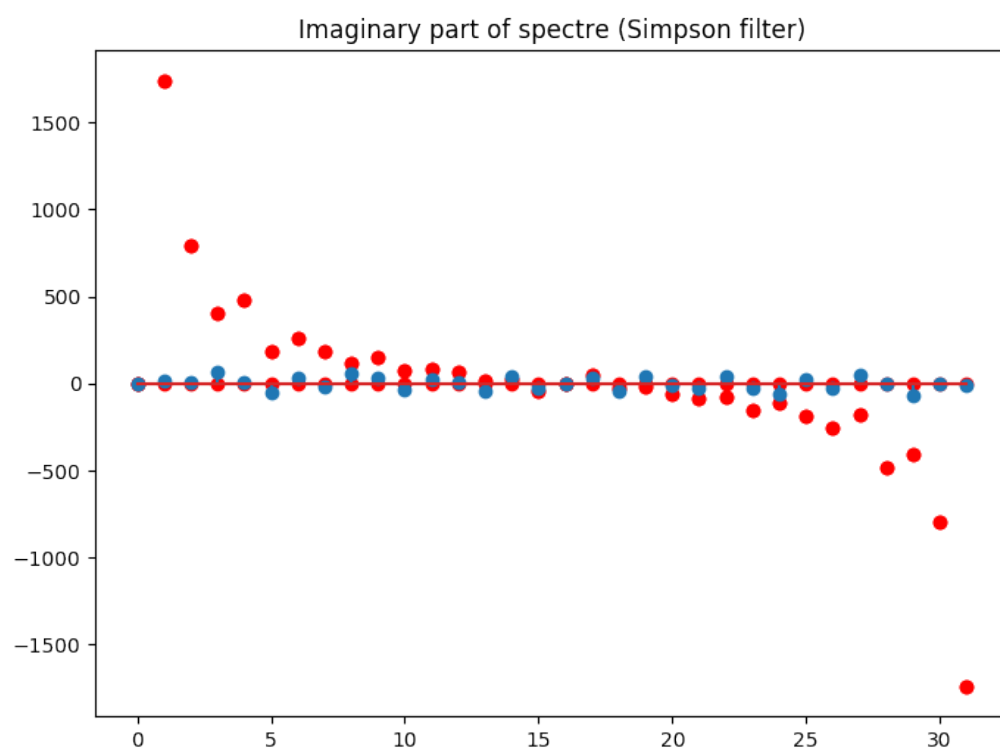


Рисунок 8.4.9 – Спектр после фильтрации, мнимая часть
(формула Симпсона)

Выводы.

В данной работе было исследовано влияние полиномиальных приближений на частотные характеристики сигнала.

Был сгенерирован сигнал, являющийся суммой гармоник, дискретизирован, после чего над ним произведено ДПФ. Преобразование Фурье раскладывает сигнал в сумму гармоник. Результирующие пики близки к частотам и амплитудам исходных гармоник. То же преобразование выполнено на сигналах, пропущенных через фильтры на основе полиномиальных приближений.

По результатам видно, что фильтры на полиномах малых степеней в первую очередь искажают высокочастотные составляющие сигнала, в то время как низкие частоты затрагиваются меньше. Так же видно, что более точные приближения (полиномами большей степени) лучше сохраняют частоты и при разложении дают более близкие гармоники.

ПРИЛОЖЕНИЕ А

ЛИСТИНГ

```
###

import math

import matplotlib.pyplot as plt
import numpy as np
from scipy.fft import fft

DPI = 100

###

def gen_signal(n_vals=32):
    harmonics = []

    amplitudes = np.random.randint(1, 11, 11)
    frequencies = [0.1 * math.pi * i for i in range(0, 11)]
    phases = np.random.uniform(0, 0.5, 11)
    for a, f, p in zip(amplitudes, frequencies, phases):
        harmonics.append([a * np.cos(f * (t * 32. / n_vals) + p) for t in
range(0, n_vals)])

    print("Amplitudes: " + str(amplitudes))
    print("Frequencies: " + str(frequencies))
    print("Phases: " + str(phases))
    return [sum(y) for y in zip(*harmonics)]

###

def plot_signal(signal):
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.xlabel('t')
    plt.ylabel('Y(t)')
    x = np.linspace(0, 31, num=32, endpoint=True)
    plt.title('Discreted')
    plt.stem(x, signal, markerfmt=' ')
    plt.show()

###

def fft_signal(signal):
    spectre = fft(signal)
    real = [y.real for y in spectre]
    imag = [y.imag for y in spectre]
    return real, imag

###

def split(signal, n):
    for i in range(0, len(signal) - n + 1):
        yield signal[i:i + n]

def linear(signal):
    smoothed = []
    for part in split(signal, 5):
        smoothed.append(sum(part) / 5)
    plt.figure(figsize=(8, 6), dpi=DPI)
```

```

plt.title('Linear smoothing by 5 points')
plt.stem(range(2, 30), smoothed, 'r', markerfmt='ro')
plt.stem(range(0, 32), signal)
plt.show()
real_part, imag_part = fft_signal(signal)

spec_real_part, spec_imag_part = fft_signal(smoothed)
plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Smoothing 5 pts: Real')
plt.stem(range(2, 30), spec_real_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), real_part)
plt.show()
plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Smoothing 5 pts: Imag')
plt.stem(range(2, 30), spec_imag_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), imag_part)
plt.show()
plt.figure(figsize=(8, 6), dpi=DPI)

smoothed = []
for part in split(signal, 9):
    smoothed.append(sum(part) / 9)
plt.title('Linear Smoothing by 9 points')
plt.stem(range(4, 28), smoothed, 'r', markerfmt='ro')
plt.stem(range(0, 32), signal)
plt.show()
plt.figure(figsize=(8, 6), dpi=DPI)

spec_real_part, spec_imag_part = fft_signal(smoothed)
plt.title('Smoothing 9 pts: Real')
plt.stem(range(4, 28), spec_real_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), real_part)
plt.show()
plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Smoothing 9 pts: Imag')
plt.stem(range(4, 28), spec_imag_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), imag_part)
plt.show()

#%%

def deg2(signal):
    real_part, imag_part = fft_signal(signal)

    smoothed = []
    for part in split(signal, 5):
        smoothed.append((-3 * part[0] + 12 * part[1] + 17 * part[2] + 12 *
part[3] - 3 * part[4]) / 35)

    plt.figure(figsize=(8, 6), dpi=DPI)

    plt.title('Polynomial (2 deg) smoothing by 5 points')
    plt.stem(range(2, 30), smoothed, 'r', markerfmt='ro')
    plt.stem(range(0, 32), signal)
    plt.show()

    plt.figure(figsize=(8, 6), dpi=DPI)

    spectre_real, spectre_imag = fft_signal(smoothed)
    plt.title('Real part of spectre (polynomial (2 deg) smoothing - 5 points)')
    plt.stem(range(2, 30), spectre_real, 'ro', markerfmt='ro')

```

```

plt.stem(range(0, 32), real_part)
plt.show()

plt.figure(figsize=(8, 6), dpi=DPI)

plt.title('Imaginary part of spectre (polynomial (2 deg) smoothing - 5
points)')
plt.stem(range(2, 30), spectre_imag, 'ro', markerfmt='ro')
plt.stem(range(0, 32), imag_part)
plt.show()

smoothed = []
for part in split(signal, 9):
    smoothed.append((-21 * part[0] + 14 * part[1] + 39 * part[2] + 54 *
part[3] + 59 * part[4] + 54 * part[5] + 39 *
part[6] + 14 * part[7] - 21 * part[8]) / 231)
plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Polynomial (2 deg) smoothing by 9 points')
plt.stem(range(4, 28), smoothed, 'r', markerfmt='ro')
plt.stem(range(0, 32), signal)
plt.show()

plt.figure(figsize=(8, 6), dpi=DPI)

spectre_real, spectre_imag = fft_signal(smoothed)
plt.title('Real part of spectre (polynomial (2 deg) smoothing - 9 points)')
plt.stem(range(4, 28), spectre_real, 'ro', markerfmt='ro')
plt.stem(range(0, 32), real_part)
plt.show()

plt.figure(figsize=(8, 6), dpi=DPI)

plt.title('Imaginary part of spectre (polynomial (2 deg) smoothing - 9
points)')
plt.stem(range(4, 28), spectre_imag, 'ro', markerfmt='ro')
plt.stem(range(0, 32), imag_part)
plt.show()

#%%

def deg4(signal):
    real_part, imag_part = fft_signal(signal)

    smoothed = []
    for part in split(signal, 7):
        smoothed.append((5 * part[0] - 30 * part[1] + 75 * part[2] + 131 *
part[3] + 75 * part[4] - 30 * part[5] + 5 *
part[6]) / 231)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Polynomial (4 deg) smoothing by 7 points')
    plt.stem(range(3, 29), smoothed, 'r', markerfmt='ro')
    plt.stem(range(0, 32), signal)
    plt.show()

plt.figure(figsize=(8, 6), dpi=DPI)

spec_real_part, spec_imag_part = fft_signal(smoothed)
plt.title('Real part of spectre (polynomial (4 deg) smoothing - 7 points)')
plt.stem(range(3, 29), spec_real_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), real_part)
plt.show()

```

```

plt.figure(figsize=(8, 6), dpi=DPI)

plt.title('Imaginary part of spectre (polynomial (4 deg) smoothing - 7
points)')
plt.stem(range(3, 29), spec_imag_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), imag_part)
plt.show()

smoothed = []
for part in split(signal, 11):
    smoothed.append((18 * part[0] - 45 * part[1] - 10 * part[2] + 60 *
part[3] + 120 * part[4] + 143 * part[
5] + 120 * part[6] + 60 * part[7] - 10 * part[8] - 45 * part[9] + 18
* part[10]) / 429)

plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Polynomial (4 deg) smoothing by 11 points')
plt.stem(range(6, 28), smoothed, 'r', markerfmt='ro')
plt.stem(range(0, 32), signal)
plt.show()

spec_real_part, spec_imag_part = fft_signal(smoothed)
plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Real part of spectre (polynomial (4 deg) smoothing - 11 points)')
plt.stem(range(6, 28), spec_real_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), real_part)
plt.show()

plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Imaginary part of spectre (polynomial (4 deg) smoothing - 11
points)')
plt.stem(range(6, 28), spec_imag_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), imag_part)
plt.show()

#%%

def diff(signal):
    real_part, imag_part = fft_signal(signal)

    smoothed = []
    for part in split(signal, 3):
        smoothed.append((part[0] - part[2]) / 2)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Differentiation smoothing')
    plt.stem(range(1, 31), smoothed, 'r', markerfmt='ro')
    plt.stem(range(0, 32), signal)
    plt.show()
    spec_real_part, spec_imag_part = fft_signal(smoothed)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Real part of spectre (differentiation smoothing)')
    plt.stem(range(1, 31), spec_real_part, 'ro', markerfmt='ro')
    plt.stem(range(0, 32), real_part)
    plt.show()
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Imaginary part of spectre (differentiation smoothing)')
    plt.stem(range(1, 31), spec_imag_part, 'ro', markerfmt='ro')
    plt.stem(range(0, 32), imag_part)
    plt.show()

```

```
#%%
```

```
def intr(signal):
    real_part, imag_part = fft_signal(signal)

    smoothed = [0]
    for i in range(1, len(signal)):
        smoothed.append(smoothed[i - 1] + signal[i])
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('MidRect filter')
    plt.stem(range(0, 32), smoothed, 'r', markerfmt='ro')
    plt.stem(range(0, 32), signal)
    plt.show()
    spec_real_part, spec_imag_part = fft_signal(smoothed)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Real part of spectre (MidRect filter)')
    plt.stem(range(0, 32), spec_real_part, 'ro', markerfmt='ro')
    plt.stem(range(0, 32), real_part)
    plt.show()
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Imaginary part of spectre (MidRect filter)')
    plt.stem(range(0, 32), spec_imag_part, 'ro', markerfmt='ro')
    plt.stem(range(0, 32), imag_part)
    plt.show()

    smoothed = [0]
    for i in range(1, len(signal)):
        smoothed.append(smoothed[i - 1] + 1 / 2 * (signal[i - 1] + signal[i]))
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Trapezoid filter')
    plt.stem(range(0, 32), smoothed, 'r', markerfmt='ro')
    plt.stem(range(0, 32), signal)
    plt.show()
    spec_real_part, spec_imag_part = fft_signal(smoothed)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Real part of spectre (Trapezoid filter)')
    plt.stem(range(0, 32), spec_real_part, 'ro', markerfmt='ro')
    plt.stem(range(0, 32), real_part)
    plt.show()
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Imaginary part of spectre (Trapezoid filter)')
    plt.stem(range(0, 32), spec_imag_part, 'ro', markerfmt='ro')
    plt.stem(range(0, 32), imag_part)
    plt.show()

    smoothed = [0, 0]
    for i in range(2, len(signal)):
        smoothed.append(smoothed[i - 2] + 1 / 3 * (signal[i - 2] + 4 * signal[i
- 1] + 4 * signal[i]))
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Simpson filter')
    plt.stem(range(0, 32), smoothed, 'r', markerfmt='ro')
    plt.stem(range(0, 32), signal)
    plt.show()
    spec_real_part, spec_imag_part = fft_signal(smoothed)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.title('Real part of spectre (Simpson filter)')
    plt.stem(range(0, 32), spec_real_part, 'ro', markerfmt='ro')
    plt.stem(range(0, 32), real_part)
    plt.show()
    plt.figure(figsize=(8, 6), dpi=DPI)
```

```

plt.title('Imaginary part of spectre (Simpson filter)')
plt.stem(range(0, 32), spec_imag_part, 'ro', markerfmt='ro')
plt.stem(range(0, 32), imag_part)
plt.show()

#%%

# task 1
ref_signal = gen_signal(n_vals=32 * 32)

#%%

plt.figure(figsize=(8, 6), dpi=DPI)
plt.xlabel("t")
plt.ylabel("s(t)")
plt.plot(ref_signal)

#%%

signal = [ref_signal[i] for i in range(0, 32 * 32, 32)] #gen_signal()
signal

#%%
# Generated for lab3, Larin.
signal = [52.42160524949278,
-2.6622798256625684,
2.7728057253343987,
2.1267205879055755,
2.477840965885533,
-0.05699028687124996,
-1.9528794559681883,
14.419836005390446,
14.277366398903238,
2.1827466469408936,
8.745696695227814,
6.635553761180789,
14.875858508006182,
9.30485316368824,
-1.1544773457313025,
9.795910665588115,
-3.3488447098733496,
-0.07801520641934068,
0.19842850405271162,
10.388204228476138,
52.4216052494928,
-2.6622798256624884,
2.772805725334382,
2.1267205879055524,
2.4778409658855383,
-0.056990286871262175,
-1.9528794559681841,
14.419836005390335,
14.277366398903299,
2.1827466469408625,
8.745696695227794,
6.635553761180775]

#%% task 2

#
plot_signal(signal)

```

```

# %%

# task 3
real_part, imag_part = fft_signal(signal)
plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Real part')
plt.stem(range(0, 32), real_part, markerfmt=' ')
plt.show()
plt.figure(figsize=(8, 6), dpi=DPI)
plt.title('Imaginary part')
plt.stem(range(0, 32), imag_part, markerfmt=' ')
plt.show()

# %% task 4,5,6
linear(signal)

# %% task 8a
deg2(signal)

# %% task 8b
deg4(signal)

# %% task 8c
diff(signal)

# %% task 8d
intr(signal)
# %%

```