**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Цифровая обработка сигналов»**

**Тема: Исследование результатов фильтрации дискретного сигнала**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8383 |  | Киреев К.А. |
| Студент гр. 8383 |  | Муковский Д.В. |
| Преподаватель |  | Середа А.-В.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.

## Выполнение работы.

## *Формирование дискретного сигнала*

Был сформирован дискретный сигнал посредством дискретизации с шагом непрерывного сигнала, представляющего собой линейную комбинацию косинусоид вида . Коэффициенты линейной комбинации были нормализованы посредством деления их на сумму полученных случайным образом амплитуд. Дискретная последовательность включает в себя 32 отсчета. Аналоговый сигнал представлен на рис. 1, дискретный сигнал – на рис. 2.

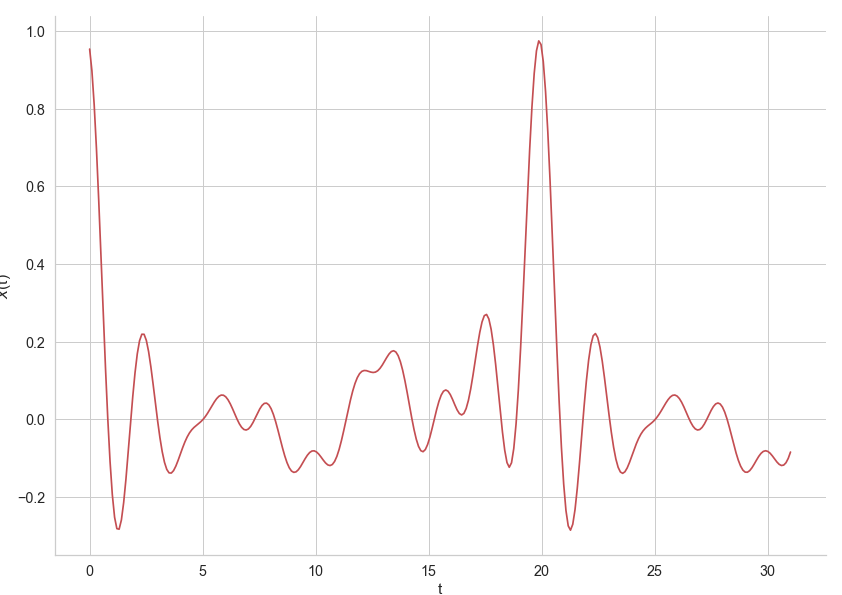


Рисунок 1 – Аналоговый сигнал

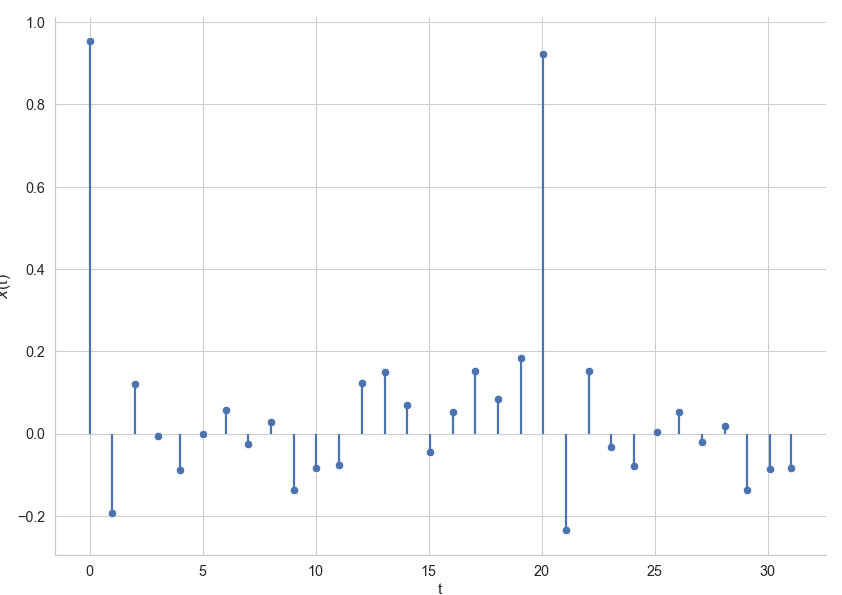


Рисунок 2 – Дискретный сигнал

## *Нахождение дискретных отсчетов спектра дискретного сигнала с помощью ДПФ*

Представим дискретный сигнал в виде функции от времени:

Тогда спектр дискретного сигнала:

С другой стороны, дискретный сигнал можно представить в виде:

Сумму можно представить в виде комплексного ряда Фурье, тогда

Полученный с помощью дискретного преобразования Фурье спектр дискретного сигнала представлен на рис. 3.

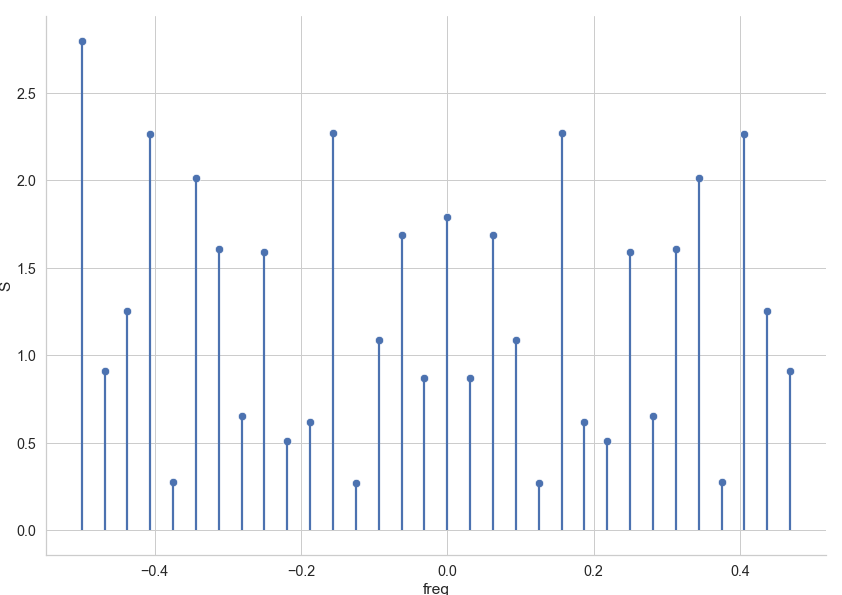


Рисунок 3 – Дискретные отсчеты спектра дискретного сигнала

Спектр симметричен относительно нуля, он представляет собой разложение исходного сигнала на линейную комбинацию простых синусоидальных функций и отражает амплитуды этих функций на разных частотах. Спектр имеет периодичность с шагом 1.

## *Применение линейного сглаживания по 5-ти и 9-ти точкам*

Линейное сглаживание по 5-ти и 9-ти точкам осуществляется с помощью полинома первой степени, а коэффициенты в передаточной функции одинаковы. Передаточные функции 5-точечного и 9-точечного НЦФ в -области:

Общая формула передаточной функции в частотной области:

На рис. 4 представлено сравнение исходного сигнала (синий) и сигнала после применения линейного сглаживания по 5-ти точкам (оранжевый).

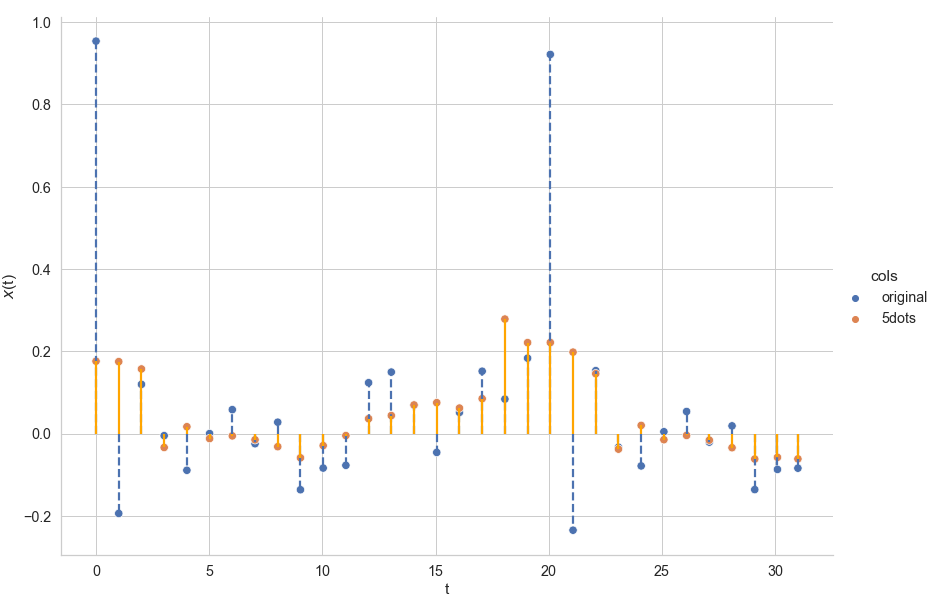


Рисунок 4 – Линейное сглаживание по 5-ти точкам

На рис. 5 представлено сравнение исходного сигнала (синий) и сигнала после применения линейного сглаживания по 9-ти точкам (оранжевый).

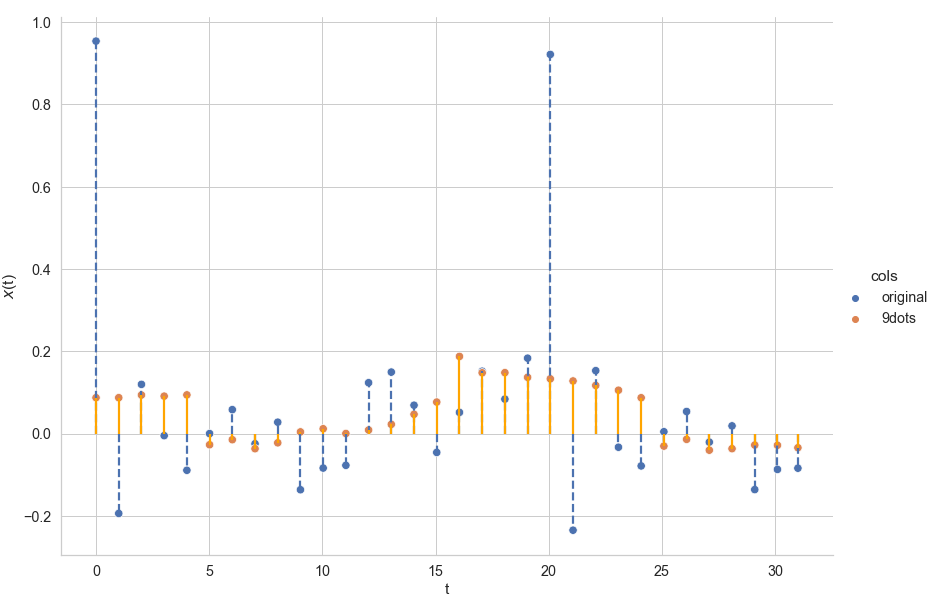


Рисунок 5 – Линейное сглаживание по 9-ти точкам

Также были получены спектры сигнала после фильтрации. На рис. 6 и 7 представлены сравнения спектров исходного сигнала и сигнала после линейного сглаживания по 5-ти и 9-ти точкам.

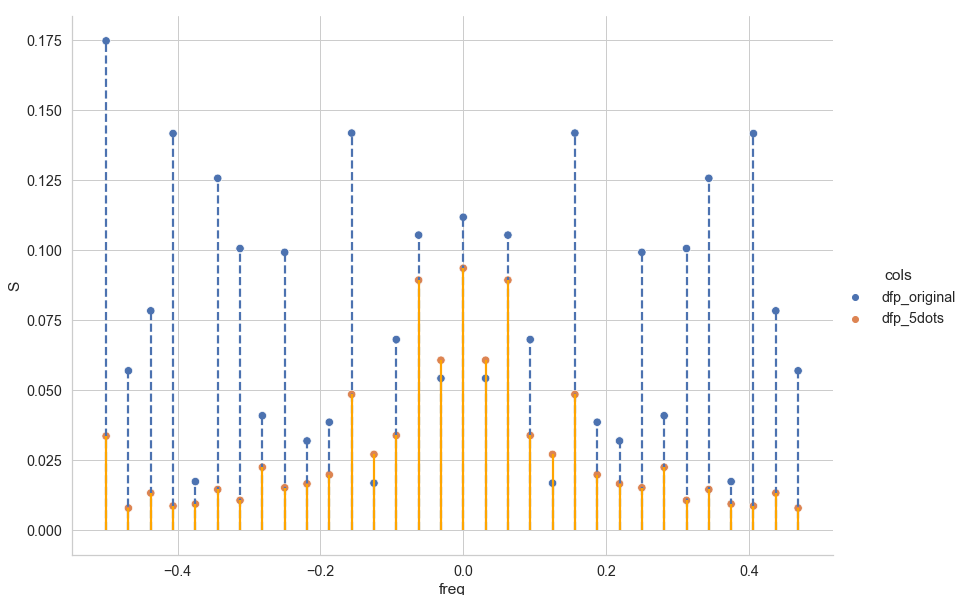


Рисунок 6 – Дискретные отсчеты спектра после линейного сглаживания

по 5-ти точкам

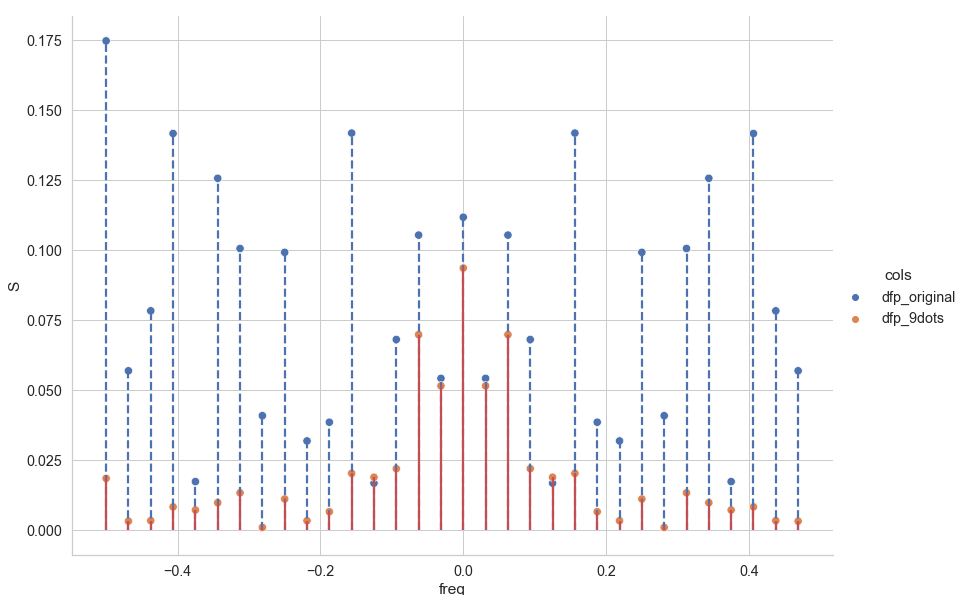


Рисунок 7 – Дискретные отсчеты спектра после линейного сглаживания

по 9-ти точкам

Из спектра видно, что без ослабления пропускается только сигнал постоянного уровня (нулевой частоты). С увеличением числа точек полоса пропускания становится уже.

График передаточной функции линейного сглаживания представлен

на рис. 8.

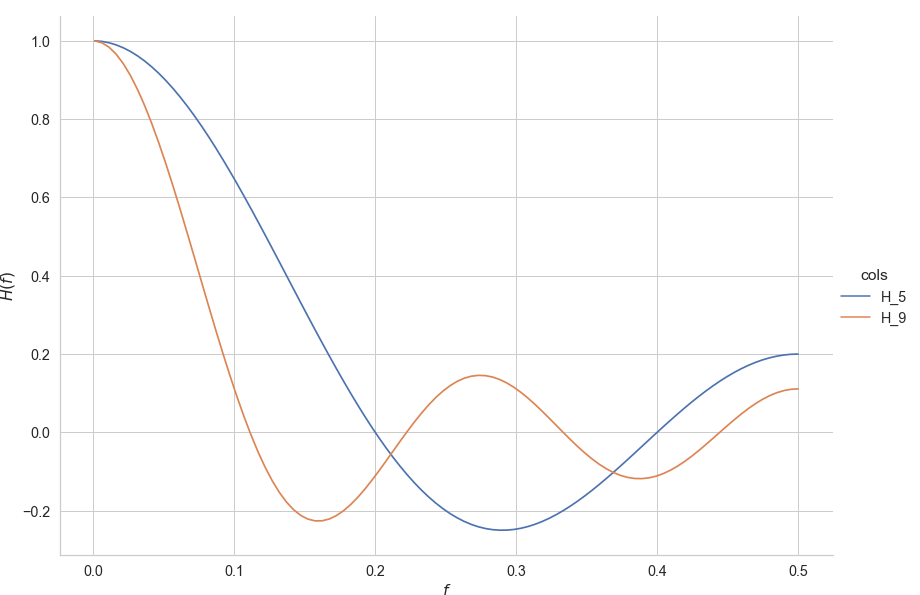


Рисунок 8 – График передаточной функции линейного сглаживания

График передаточной функции подтверждает ранее сделанные выводы относительно полосы пропускания и ослабления уровня.

## *Сглаживание* *полиномом 2-й степени по 5 и 9 узлам*

Формулы передаточных функций для сглаживания полиномом второй степени:

На рис. 9 и 10 представлено сравнение исходного сигнала и сигнала после применения сглаживания полиномом 2-й степени по 5-ти и 9-ти точкам.

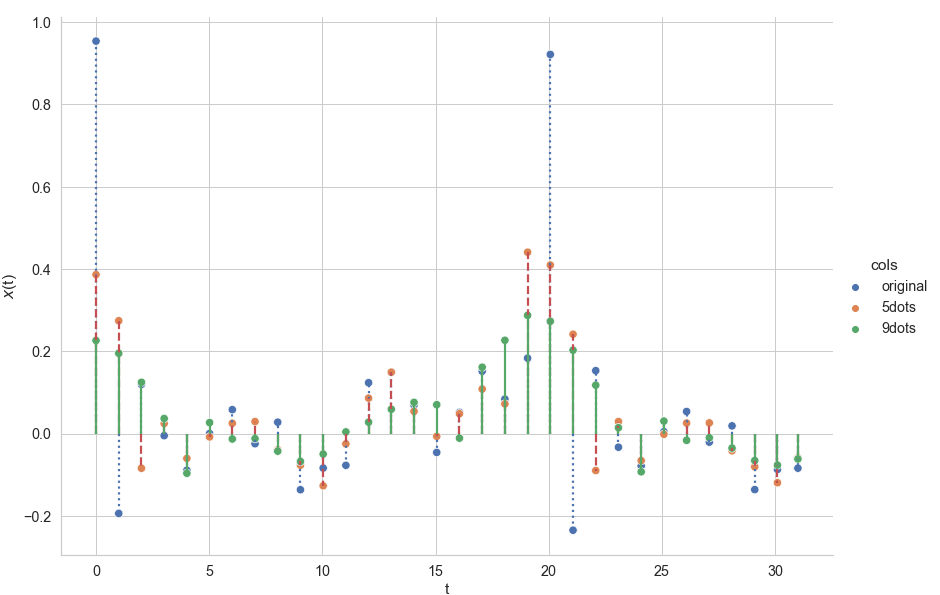


Рисунок 9 – Сглаживание полиномом 2-й степени по 5-ти и 9-ти узлам

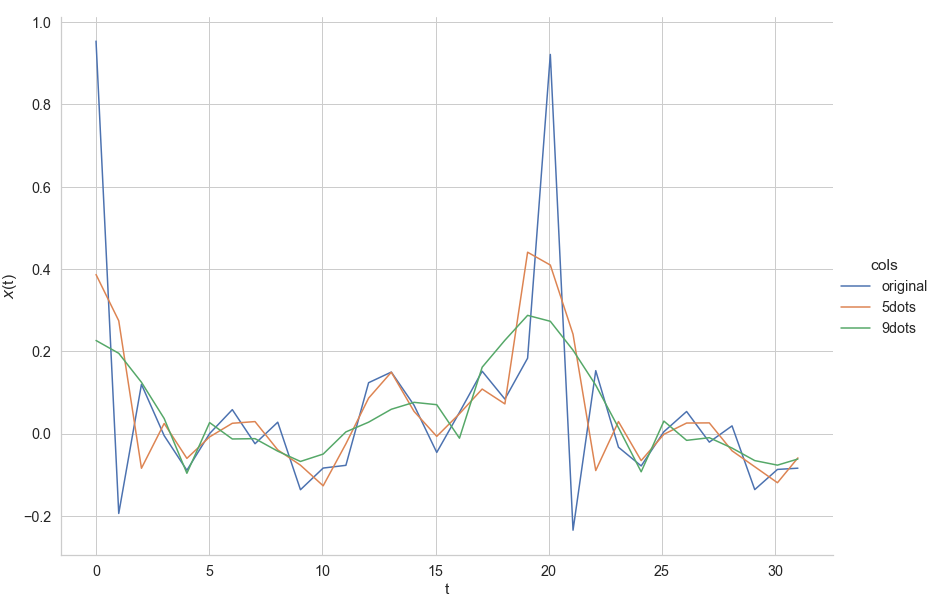


Рисунок 10 – Сглаживание полиномом 2-й степени по 5-ти и 9-ти узлам (точки соединены для лучшей визуализации)

Также были получены спектры сигнала после фильтрации. На рис. 11 и 12 представлено сравнение спектров исходного сигнала и сигнала после сглаживания полиномом 2-й степени по 5-ти и 9-ти узлам.

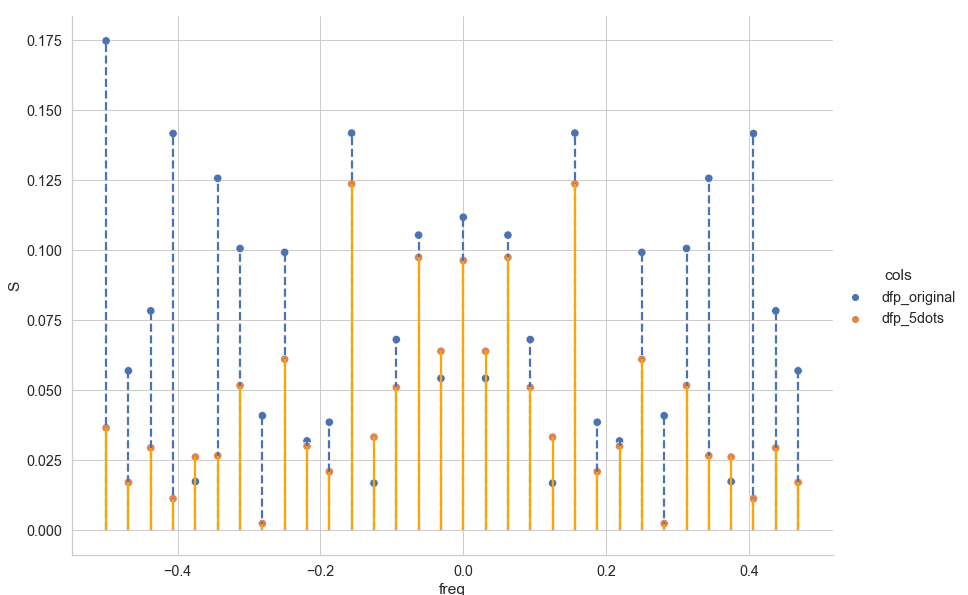


Рисунок 11 – Спектры сигналов после сглаживание фильтром 2-го порядка

по 5-ти узлам

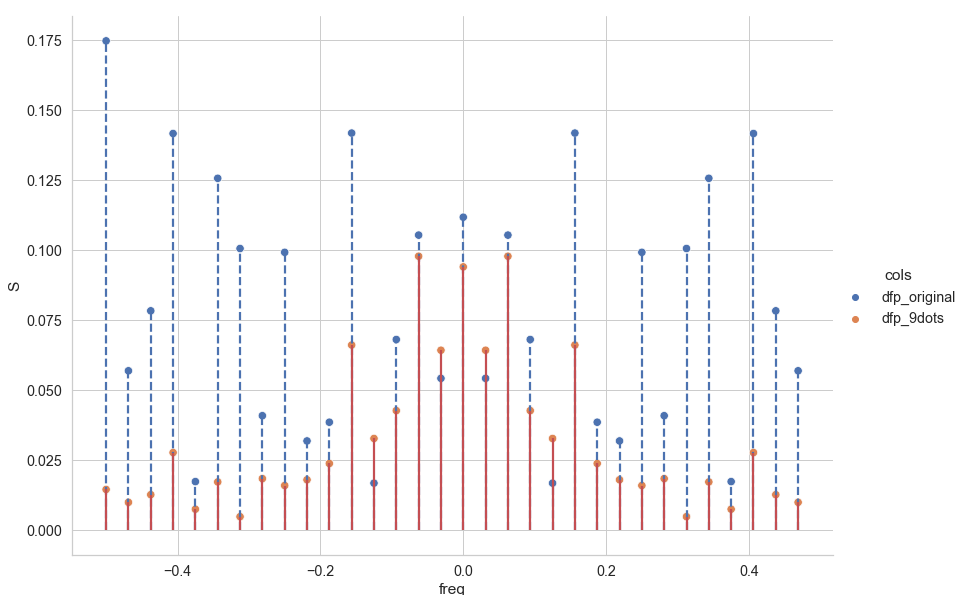


Рисунок 12 – Спектры сигналов после сглаживание фильтром 2-го порядка

по 9-ти узлам

На рис. 13 представлены графики передаточных функций фильтра сглаживания полиномом второй степени.

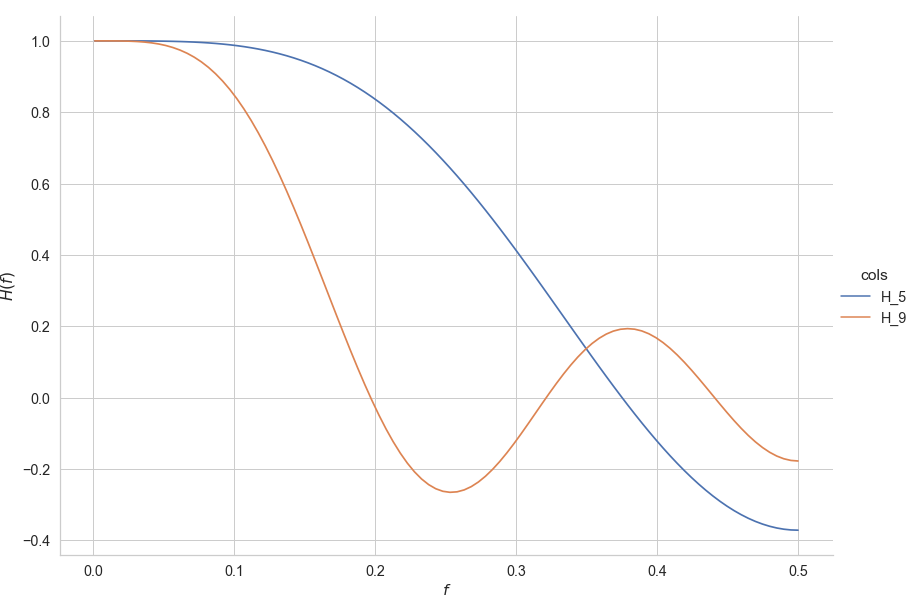


Рисунок 13 – Передаточная функция фильтра 2-го порядка

На графике видно, что при увеличении числа точек увеличивается крутизна среза и уменьшается полоса пропускания.

Также присутствует более пологая часть в полосе пропускания около 0, в отличие от линейного фильтра, поэтому пропускаются сигналы низкой частоты почти без потери амплитуды в более широком диапазоне, что видно на графиках спектра.

## *Сглаживание полиномом 4-й степени по 7 и 11 узлам*

Формулы передаточных функций для сглаживания полиномом четвертой степени:

На рис. 14 и 15 представлено сравнение исходного сигнала и сигнала после применения сглаживания полиномом 4-й степени по 7-ми и 11-ти точкам.

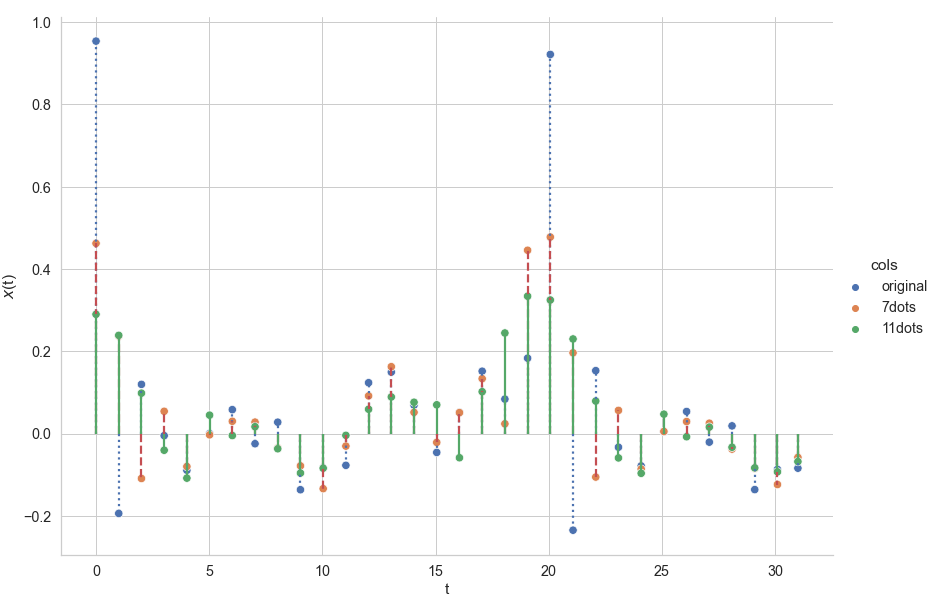


Рисунок 14 – Сглаживание полиномом 4-й степени по 7-ми и 11-ти точкам

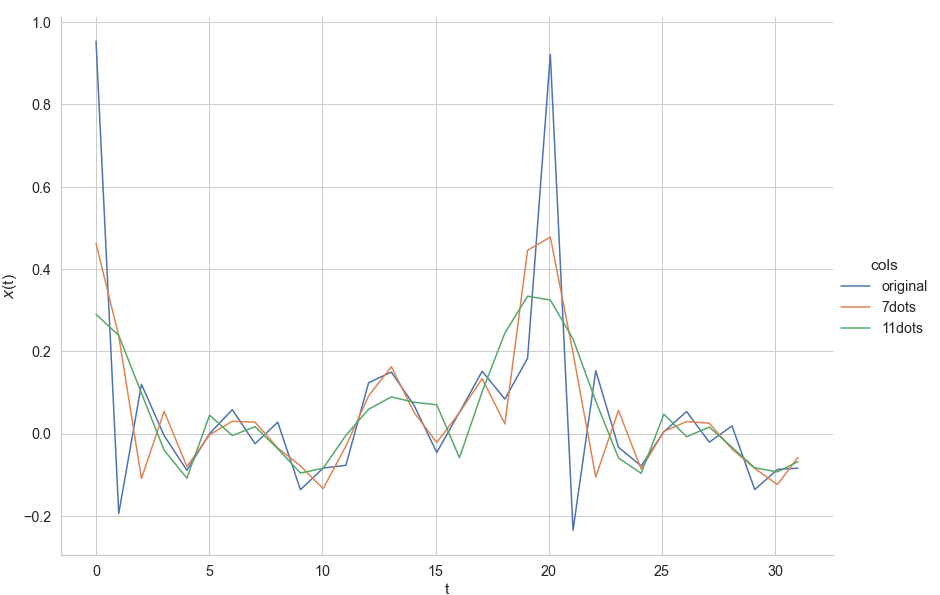


Рисунок 15 – Сглаживание полиномом 4-й степени по 7-ми и 11-ти точкам

Также были получены спектры сигнала после фильтрации. На рис. 16 и 17 представлено сравнение спектров исходного сигнала, сигнала после сглаживания полиномом 4-й степени по 7-ми и 11-ти точкам.

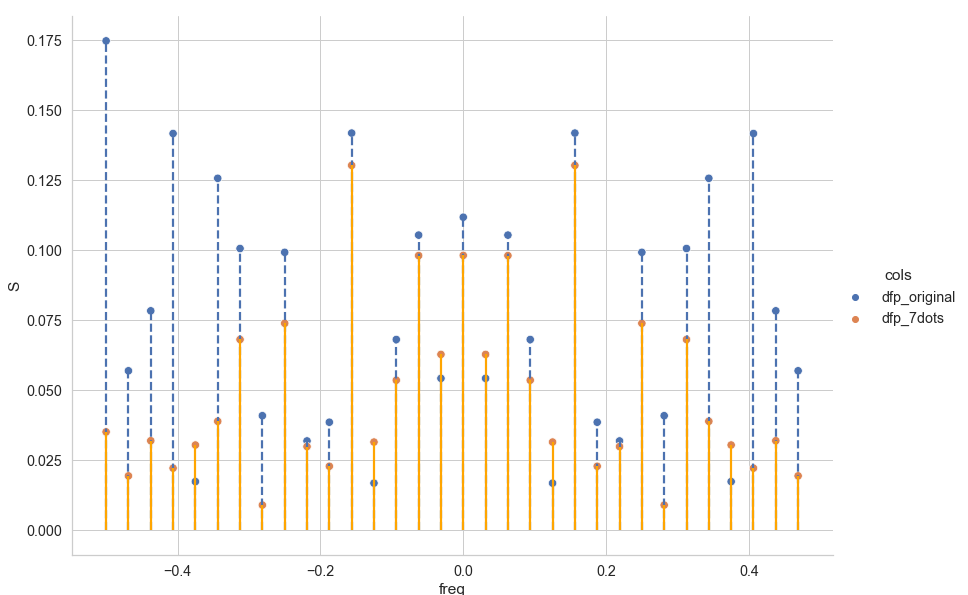


Рисунок 16 – Спектры сигналов после сглаживание фильтром 4-го порядка

по 7-ми точками

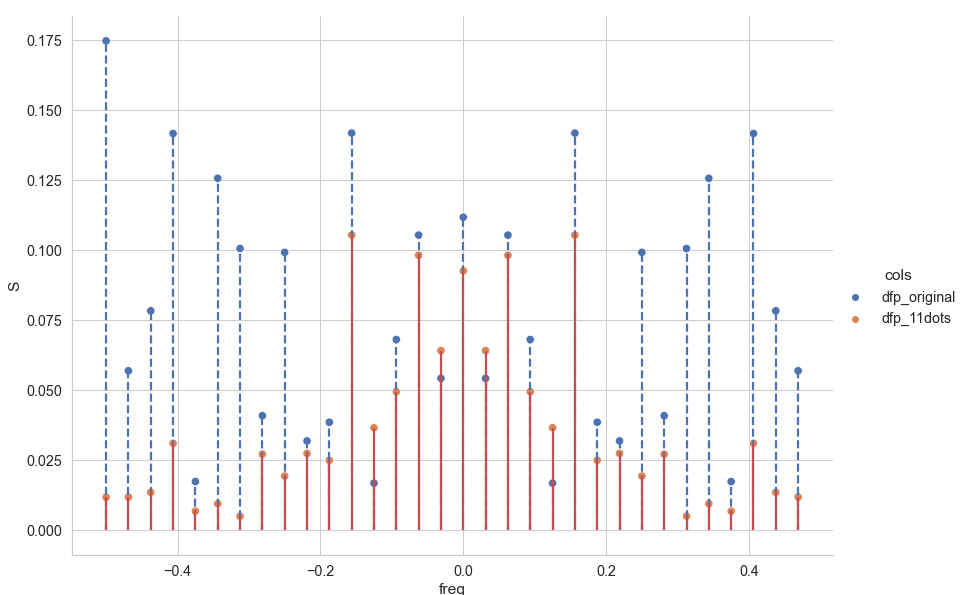


Рисунок 17 – Спектры сигналов после сглаживание фильтром 4-го порядка

по 9-ти точками

На рис. 18 представлены графики передаточных функций фильтра сглаживания полиномом четвертой степени.

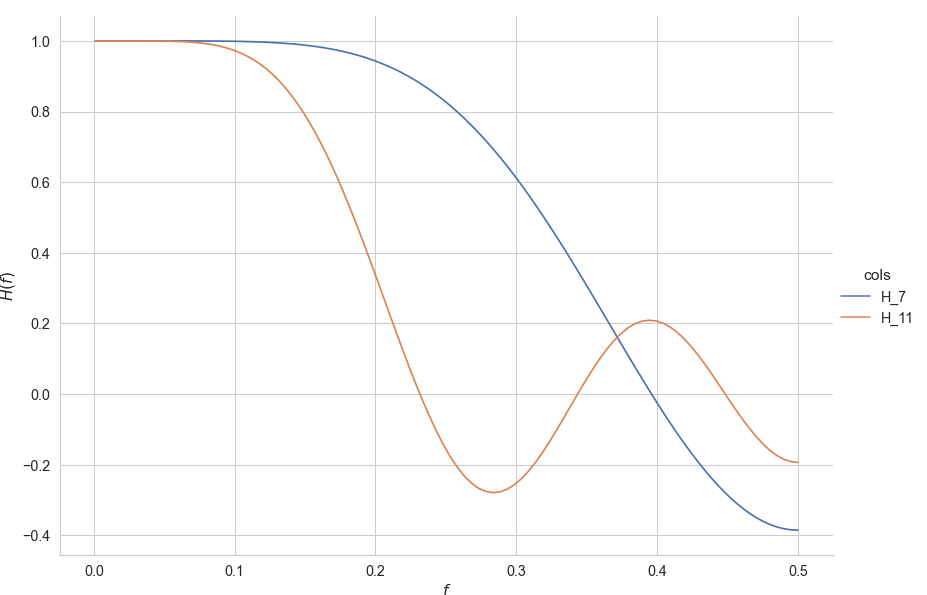


Рисунок 18 – Передаточная функция фильтра 4-го порядка

Полученные спектры после фильтрации схожи с таковыми при сглаживании полиномом 2-й степени, однако здесь наблюдается еще более широкая пологая часть пропускания низких частот почти без ослабления.

При сглаживании по 11-ти точкам можно увидеть, что полоса пропускания становится уже, что видно на спектре.

## *Дискретный фильтр, соответствующий численному дифференцированию 1-го порядка*

Передаточная функция имеет вид:

На рис. 19 и 20 представлено сравнение исходного сигнала и сигнала после применения сглаживания фильтром дифференцирования первого порядка:

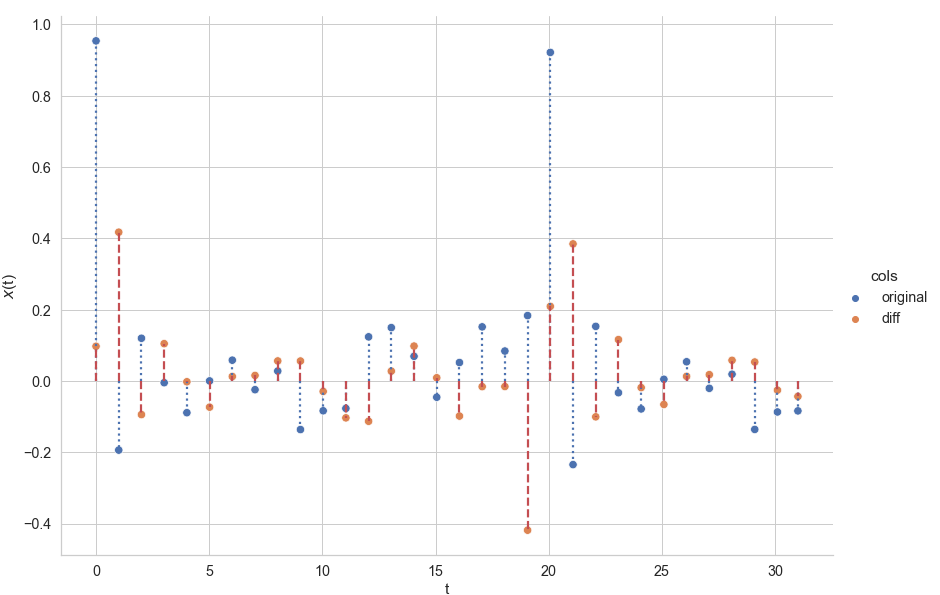


Рисунок 19 – Сглаживание фильтром дифференцирования первого порядка

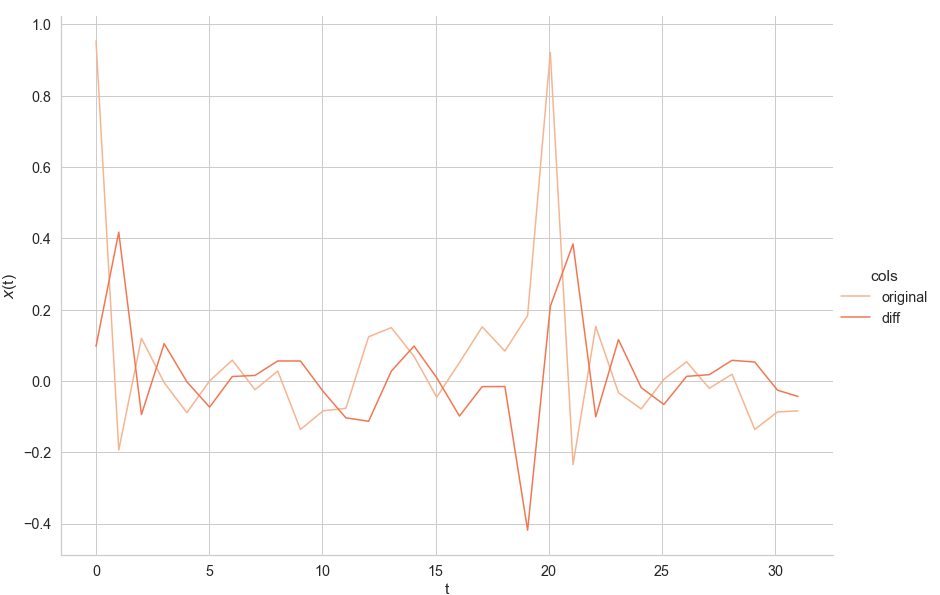


Рисунок 20 – Сглаживание фильтром дифференцирования первого порядка

Также был получен спектр сигнала после фильтрации. Сравнение спектров исходного и фильтрованного сигнала представлено на рис. 21.

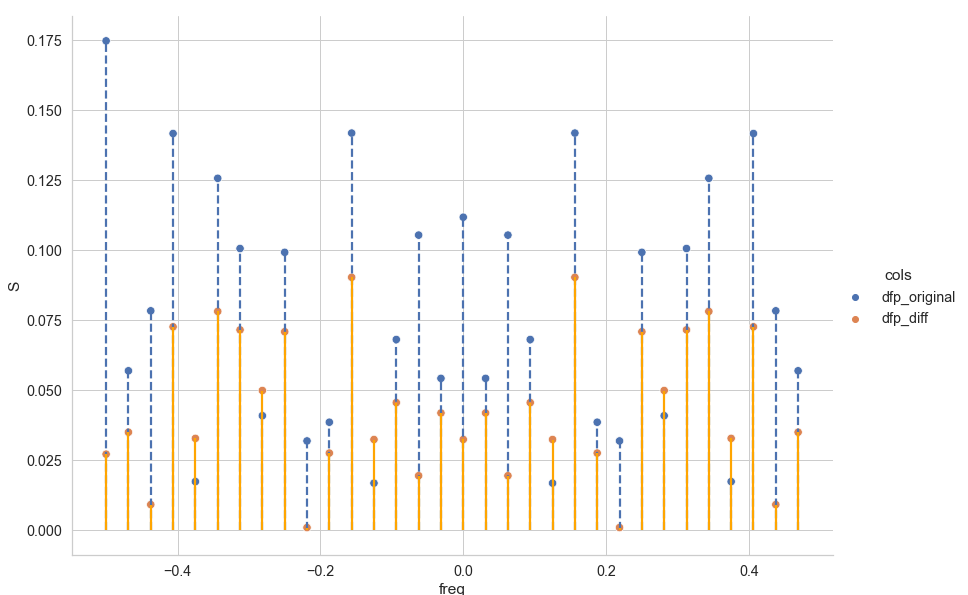


Рисунок 21 – Сравнение спектров исходного и фильтрованного сигнала

График передаточной функции представлен на рис. 22.

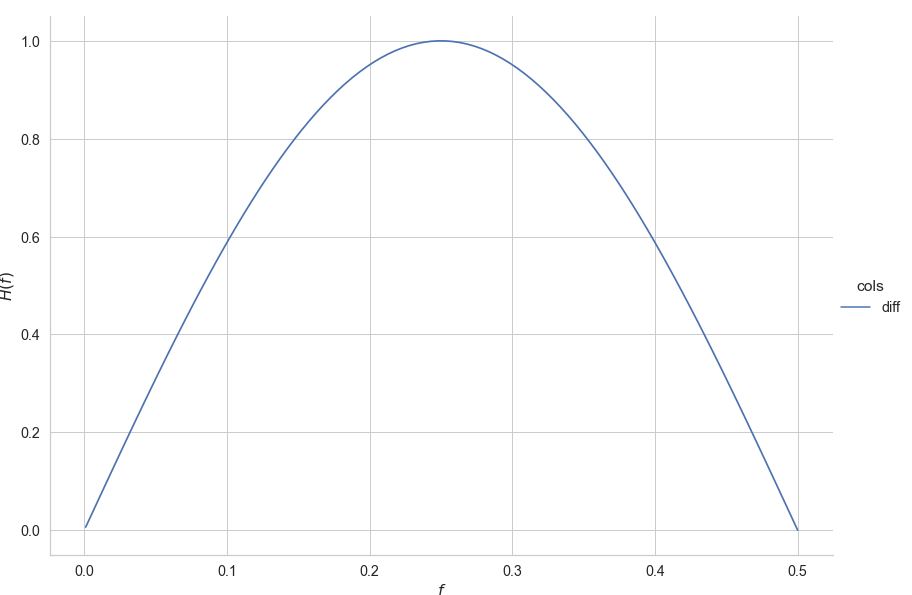


Рисунок 22 – Передаточная функция

По графику спектра видно, что средние частоты действительно усиливаются, при этом низкие и высокие частоты сглаживаются, что подтверждается графиком передаточной функции.

## *Дискретный фильтр, соответствующий численному интегрированию (прямоугольников, трапеций, Симпсона)*

Передаточная функция для рекурсивного фильтра, соответствующего численному интегрированию прямоугольников:

Передаточная функция для рекурсивного фильтра, соответствующего численному интегрированию трапеций:

Передаточная функция для рекурсивного фильтра, соответствующего численному интегрированию по формуле Симпсона:

На рис. 23 представлено сравнение исходного сигнала и сигнала после применения сглаживания фильтрами интегрирования.

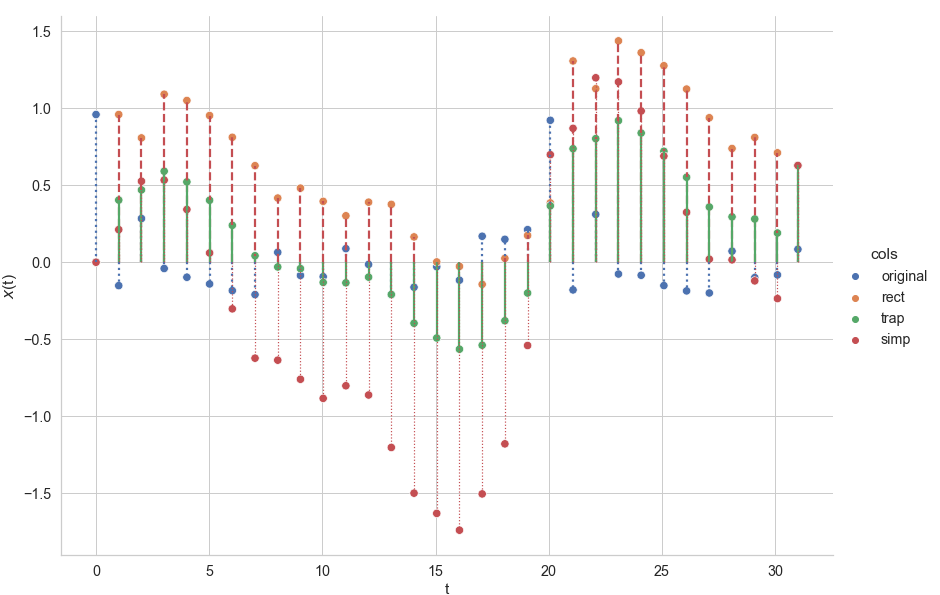


Рисунок 23 – Сравнение исходного и отфильтрованного сигналов

Также был получен спектр сигнала после фильтрации. Сравнение спектров исходного и фильтрованного сигнала представлено на рис. 24, 25 и 26.

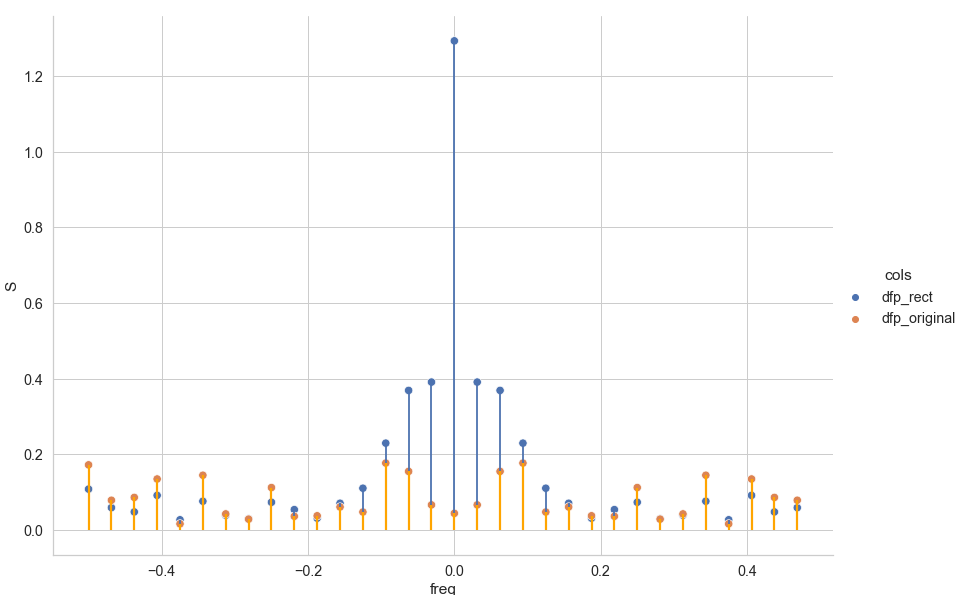


Рисунок 24 – Спектр исходного и отфильтрованных сигналов

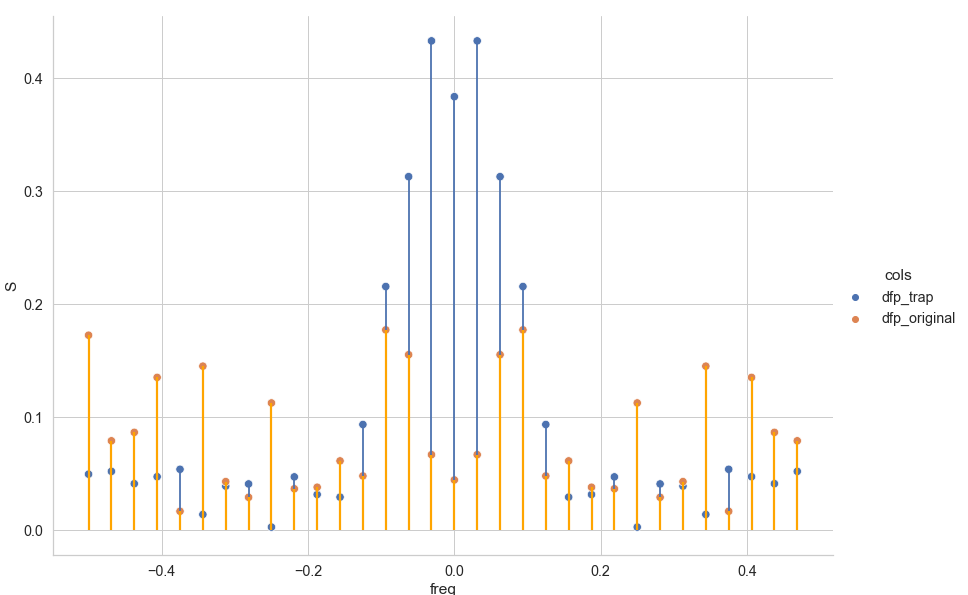


Рисунок 25 – Спектр исходного и отфильтрованных сигналов

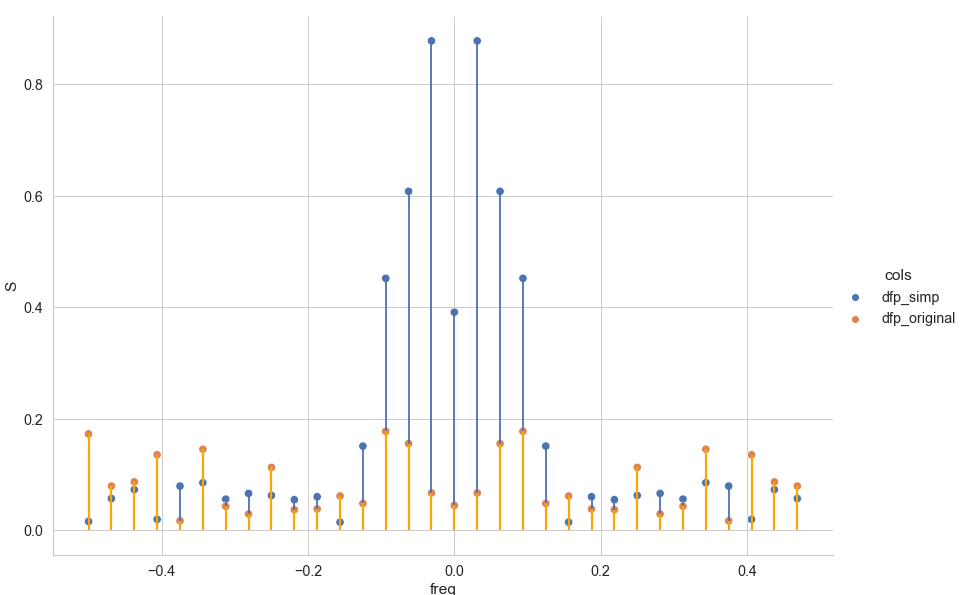


Рисунок 26 – Спектр исходного и отфильтрованных сигналов

Из спектров можно увидеть отличие сглаживаний интегрированием с помощью различных формул: сглаживание по формулам прямоугольников значительно усиливает низкие частоты в небольшом диапазоне; сглаживание по формулам трапеции усиливает низкие частоты, но в диапазоне шире и с меньшим усилением; сглаживание по формулам Симпсона усиливает весь спектр частот, и особенно нижние. Графики передаточных функций представлены на рис. 27.

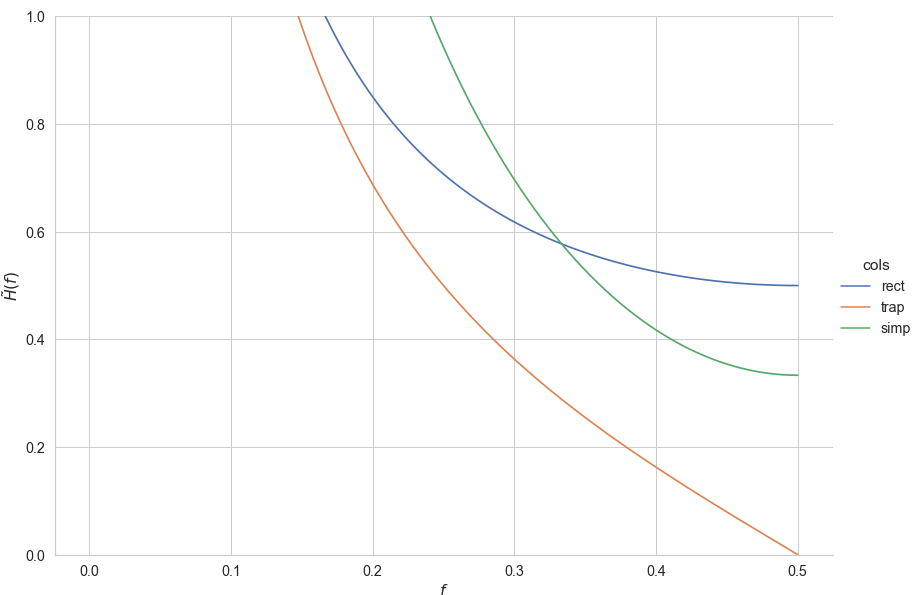


Рисунок 27 – Передаточная функция

## Выводы

В результате выполнения лабораторной работы был сгенерирован аналоговый сигнал, дискретизирован, построен спектр дискретного сигнала. Было выяснено, что спектр дискретного сигнала симметричен относительно 0, в спектре представлено множество частот. Были применены фильтры: линейного сглаживания по 5, 9 узлам, сглаживания полиномом 2-й по 5, 9 узлам и 4-й степени по 7, 11 узлам, построены графики сигнала и спектра, передаточной функции. В результате по спектру было определено, что обеспечивается фильтрация высоких частот и ширина полосы пропускания уменьшается с увеличением числа точек. Был применен дискретный фильтр, соответствующий численному дифференцированию, построен график сигнала, спектр, передаточная функция. В результате графики спектра и передаточной функции показали, что фильтр имеет полосу пропускания в области средних частот, уменьшает амплитуду низких и высоких частот. Были применены фильтры, соответствующие численному интегрированию по формулам прямоугольников, трапеций, Симпсона. Фильтры отличаются усилением сигнала в области низких частот.

# Приложение А.

# Исходный код программы

import pandas as pd

import numpy as np

from scipy.fftpack import fft, ifft, fftshift, rfft

import seaborn as sns

import matplotlib.pyplot as plt

from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast\_node\_interactivity = "all"

# ## Пункт 1

def signal(n):

t, Y = np.linspace(0, 31, n), 0

A = np.random.randint(1, 11, 11)

W = np.arange(0, 1.01\*np.pi, 0.1\*np.pi)

Phi = np.random.uniform(0, 0.5, [11,])

for a, w, phi in zip(A, W, Phi):

Y += a\*np.cos(w\*t+phi)

Y = Y/np.sum(A)

return t, Y

# ## Пункт 2

t, Y = signal(310)

df\_a = pd.DataFrame({'t': t, 'Y': Y})

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_a, x='t', y='Y', linewidth=1.7, color='r',

kind='line', height=8.27, aspect=11.7/8.27)

ax.set\_axis\_labels('t', r'$x$(t)')

plt.savefig('pics/2\_1.png')

plt.show()

df = df\_a.iloc[np.concatenate((np.linspace(0, 300, 31), [309]), axis=0)]

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df, x='t', y='Y', color='b', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df['t'], ymin=0, ymax=df['Y'], linewidth=2.2)

ax.set\_axis\_labels('t', r'$x$(t)')

plt.savefig('pics/2\_2.png')

plt.show()

# ## Пункт 3

def DFT(x):

x = np.asarray(x, dtype=float)

N = x.shape[0]

n = np.arange(N)

k = n.reshape((N, 1))

M = np.exp(-2j \* np.pi \* k \* n / N)

return np.dot(M, x)

df\_3 = df.copy()

df\_3['Y'] = 2 \* np.abs(fft(df\_3['Y'].values)) / 32

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_3, x='t', y='Y', color='b', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_3['t'], ymin=0, ymax=df\_3['Y'], linewidth=2.2)

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/3.png')

plt.show()

# ## Пункт 5

#

# ### Линейное сглаживание по 5 и 9 точкам

df\_47 = df.copy()

df\_47['original'] = df\_47['Y']

df\_47['5dots'] = np.convolve(df['Y'], np.ones(5), 'same') / 5

df\_47['9dots'] = np.convolve(df['Y'], np.ones(9), 'same') / 9

df\_47m5 = df\_47.melt(id\_vars='t', value\_vars=['original', '5dots'], var\_name='cols', value\_name='vals')

df\_47m9 = df\_47.melt(id\_vars='t', value\_vars=['original', '9dots'], var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_47m5, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_47['t'], ymin=0, ymax=df\_47['original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_47['t'], ymin=0, ymax=df\_47['5dots'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('t', r'$x$(t)')

plt.savefig('pics/5\_1.png')

plt.show()

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_47m9, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_47['t'], ymin=0, ymax=df\_47['original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_47['t'], ymin=0, ymax=df\_47['9dots'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('t', r'$x$(t)')

plt.savefig('pics/5\_2.png')

plt.show()

# ## Пункт 6

df\_47['dfp\_original'] = 2 \* np.abs(fft(df\_47['Y'].values)) / 32

df\_47['dfp\_5dots'] = 2 \* np.abs(fft(df\_47['5dots'].values)) / 32

df\_47['dfp\_9dots'] = 2 \* np.abs(fft(df\_47['9dots'].values)) / 32

df\_47m5dfp = df\_47.melt(id\_vars='t', value\_vars=['dfp\_original', 'dfp\_5dots'], var\_name='cols', value\_name='vals')

df\_47m9dfp = df\_47.melt(id\_vars='t', value\_vars=['dfp\_original', 'dfp\_9dots'], var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_47m5dfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_47['t'], ymin=0, ymax=df\_47['dfp\_original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_47['t'], ymin=0, ymax=df\_47['dfp\_5dots'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/6\_1.png')

plt.show()

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_47m9dfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_47['t'], ymin=0, ymax=df\_47['dfp\_original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_47['t'], ymin=0, ymax=df\_47['dfp\_9dots'], linewidth=2.2, color='r')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/6\_2.png')

plt.show()

# ## Пункт 8

#

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

df\_8a = df.copy()

df\_8a['original'] = df\_8a['Y']

df\_8a = df\_8a.drop(['Y'], axis=1)

df\_8a['5dots'] = np.convolve(df['Y'], np.array([-3, 12, 17, 12, -3]), 'same') / 35

df\_8a['9dots'] = np.convolve(df['Y'], np.array([-21, 14, 39, 54, 59, 54, 39, 14, -21]), 'same') / 231

df\_8a\_m = df\_8a.melt(id\_vars='t', var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8a\_m, x='t', y='vals', hue='cols',

kind='line', height=8.27, aspect=11.7/8.27)

ax.set\_axis\_labels('t', r'$x$(t)')

plt.savefig('pics/8a\_1.png')

plt.show()

df\_8a['dfp\_original'] = 2 \* np.abs(fft(df\_8a['original'].values)) / 32

df\_8a['dfp\_5dots'] = 2 \* np.abs(fft(df\_8a['5dots'].values)) / 32

df\_8a['dfp\_9dots'] = 2 \* np.abs(fft(df\_8a['9dots'].values)) / 32

df\_8a\_m5dfp = df\_8a.melt(id\_vars='t', value\_vars=['dfp\_original', 'dfp\_5dots'], var\_name='cols', value\_name='vals')

df\_8a\_m9dfp = df\_8a.melt(id\_vars='t', value\_vars=['dfp\_original', 'dfp\_9dots'], var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8a\_m5dfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_8a['t'], ymin=0, ymax=df\_8a['dfp\_original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_8a['t'], ymin=0, ymax=df\_8a['dfp\_5dots'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/8a\_2.png')

plt.show()

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8a\_m9dfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_8a['t'], ymin=0, ymax=df\_8a['dfp\_original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_8a['t'], ymin=0, ymax=df\_8a['dfp\_9dots'], linewidth=2.2, color='r')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/8a\_3.png')

plt.show()

# ### B. Сглаживание полиномом 4-ой степени по 7 и 11 узлам

df\_8b = df.copy()

df\_8b['original'] = df\_8b['Y']

df\_8b = df\_8b.drop(['Y'], axis=1)

df\_8b['7dots'] = np.convolve(df['Y'], np.array([5, -30, 75, 131, 75, -30, 5]), 'same') / 231

df\_8b['11dots'] = np.convolve(df['Y'], np.array([13, -45, -10, 60, 120, 143, 120, 60, -10, -45, 13]), 'same') / 429

df\_8b\_m = df\_8b.melt(id\_vars='t', var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8b\_m, x='t', y='vals', hue='cols',

kind='line', height=8.27, aspect=11.7/8.27)

ax.set\_axis\_labels('t', r'$x$(t)')

plt.savefig('pics/8b\_1.png')

plt.show()

df\_8b['dfp\_original'] = 2 \* np.abs(fft(df\_8b['original'].values)) / 32

df\_8b['dfp\_7dots'] = 2 \* np.abs(fft(df\_8b['7dots'].values)) / 32

df\_8b['dfp\_11dots'] = 2 \* np.abs(fft(df\_8b['11dots'].values)) / 32

df\_8b\_m7dfp = df\_8b.melt(id\_vars='t', value\_vars=['dfp\_original', 'dfp\_7dots'], var\_name='cols', value\_name='vals')

df\_8b\_m11dfp = df\_8b.melt(id\_vars='t', value\_vars=['dfp\_original', 'dfp\_11dots'], var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8b\_m7dfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_8b['t'], ymin=0, ymax=df\_8b['dfp\_original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_8b['t'], ymin=0, ymax=df\_8b['dfp\_7dots'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/8b\_2.png')

plt.show()

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8b\_m11dfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_8b['t'], ymin=0, ymax=df\_8b['dfp\_original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_8b['t'], ymin=0, ymax=df\_8b['dfp\_11dots'], linewidth=2.2, color='r')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/8b\_3.png')

plt.show()

# ### C. Дискретный фильтр, соответствующий численному дифференцированию 1-го порядка

df\_8c = df.copy()

df\_8c['original'] = df\_8c['Y']

df\_8c = df\_8c.drop(['Y'], axis=1)

df\_8c['diff'] = np.convolve(df['Y'], np.array([-1, 0, 1]), 'same') / 2

df\_8c\_m = df\_8c.melt(id\_vars='t', var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='rocket\_r', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8c\_m, x='t', y='vals', hue='cols',

kind='line', height=8.27, aspect=11.7/8.27)

ax.set\_axis\_labels('t', r'$x$(t)')

plt.savefig('pics/8c\_1.png')

plt.show()

df\_8c['dfp\_original'] = 2 \* np.abs(fft(df\_8c['original'].values)) / 32

df\_8c['dfp\_diff'] = 2 \* np.abs(fft(df\_8c['diff'].values)) / 32

df\_8c\_m\_diffdfp = df\_8c.melt(id\_vars='t', value\_vars=['dfp\_original', 'dfp\_diff'], var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8c\_m\_diffdfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_8c['t'], ymin=0, ymax=df\_8c['dfp\_original'], linewidth=2.2, linestyles='--')

plt.vlines(x=df\_8c['t'], ymin=0, ymax=df\_8c['dfp\_diff'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/8c\_2.png')

plt.show()

# ### D. Дискретный фильтр, соответствующий численному интегрированию (прямоугольников, трапеций, Симпсона)

def rect(orig):

integr = np.empty(len(orig))

integr[0] = orig[0]

for i in range(1, len(orig)):

integr[i] = integr[i-1] + orig[i]

return integr

def simpson(orig):

integr = np.empty(len(orig))

integr[0] = (0 + 4\*orig[0] + orig[1]) / 3

for i in range(1, len(orig)-1):

integr[i] = integr[i-1] + (orig[i-1] + orig[i] + 4\*orig[i+1]) / 3

return integr

def trap(orig):

integr = np.empty(len(orig))

integr[0] = (orig[0] + orig[1]) / 2

for i in range(1, len(orig)-1):

integr[i] = integr[i - 1] + (orig[i] + orig[i+1]) / 2

return integr

df\_8d = df.copy()

df\_8d['original'] = df\_8d['Y']

df\_8d = df\_8d.drop(['Y'], axis=1)

df\_8d['rect'] = rect(df['Y'].values)

df\_8d['trap'] = trap(df['Y'].values)

df\_8d['simp'] = simpson(df['Y'].values)

df\_8d\_m = df\_8d.melt(id\_vars='t', var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8d\_m, x='t', y='vals', hue='cols',

kind='line', height=8.27, aspect=11.7/8.27)

ax.set\_axis\_labels('t', r'$x$(t)')

plt.savefig('pics/8d\_1.png')

plt.show()

df\_8d['dfp\_original'] = 2 \* np.abs(fft(df\_8d['original'].values)) / 32

df\_8d['dfp\_rect'] = 2 \* np.abs(fft(df\_8d['rect'].values)) / 32

df\_8d['dfp\_trap'] = 2 \* np.abs(fft(df\_8d['trap'].values)) / 32

df\_8d['dfp\_simp'] = 2 \* np.abs(fft(df\_8d['simp'].values)) / 32

df\_8d\_mRdfp = df\_8d.melt(id\_vars='t', value\_vars=['dfp\_rect', 'dfp\_original'], var\_name='cols', value\_name='vals')

df\_8d\_mTdfp = df\_8d.melt(id\_vars='t', value\_vars=['dfp\_trap', 'dfp\_original'], var\_name='cols', value\_name='vals')

df\_8d\_mSdfp = df\_8d.melt(id\_vars='t', value\_vars=['dfp\_simp', 'dfp\_original'], var\_name='cols', value\_name='vals')

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8d\_mRdfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_8d['t'], ymin=0, ymax=df\_8d['dfp\_rect'], linewidth=1.8, linestyles='solid')

plt.vlines(x=df\_8d['t'], ymin=0, ymax=df\_8d['dfp\_original'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/8d\_2.png')

plt.show()

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8d\_mTdfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_8d['t'], ymin=0, ymax=df\_8d['dfp\_trap'], linewidth=1.8, linestyles='solid')

plt.vlines(x=df\_8d['t'], ymin=0, ymax=df\_8d['dfp\_original'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/8d\_3.png')

plt.show()

sns.set\_theme(style="whitegrid", palette='deep', context='notebook', font\_scale=1.3)

ax = sns.relplot(data=df\_8d\_mSdfp, x='t', y='vals', hue='cols', s=70,

kind='scatter', height=8.27, aspect=11.7/8.27)

plt.vlines(x=df\_8d['t'], ymin=0, ymax=df\_8d['dfp\_simp'], linewidth=1.8, linestyles='solid')

plt.vlines(x=df\_8d['t'], ymin=0, ymax=df\_8d['dfp\_original'], linewidth=2.2, color='orange')

ax.set\_axis\_labels('freq', r'level')

plt.savefig('pics/8d\_4.png')

plt.show()