**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №6**

**по дисциплине «Цифровая обработка сигналов»**

Тема: Исследование результатов фильтрации дискретного сигнала с помощью рекурсивных фильтров, построенных на основе формул численного дифференцирования и интегрирования

|  |  |  |
| --- | --- | --- |
| Студент гр. 9381 |  | Колованов Р.А. |
| Студент гр. 9381 |  | Семенов А.Н. |
| Преподаватель |  | Середа А.-В. И. |

Санкт-Петербург

2022

**Цель работы.**

Получение практических навыков выполнения фильтрации дискретных последовательностей с помощью фильтров, основанных на формулах численного дифференцирования и интегрирования, а также анализа получаемых результатов с помощью дискретного преобразования Фурье (ДПФ).

**Основные теоретические положения.**

Дискретный сигнал: , как правило, получается при дискретизации аналогового сигнала .

Будем считать, что отчеты дискретного сигнала получены в результате равномерной дискретизации сигнала с шагом дискретизации, равным единице:

,

*,*

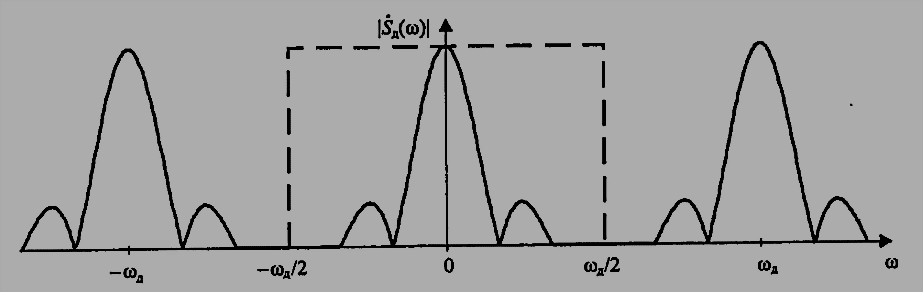
*Спектр дискретного сигнала.*

Представим дискретный сигнал в виде функции от времени:

Тогда, пользуясь свойствами преобразования Фурье, спектр дискретного сигнала можно представить в виде периодической функции с периодом, равным :

Отсюда дискретный сигнал может быть записан в виде:

А его спектр – в виде (см. рис. 0 а):



*Рисунок 0 (а). График спектра дискретного сигнала.*

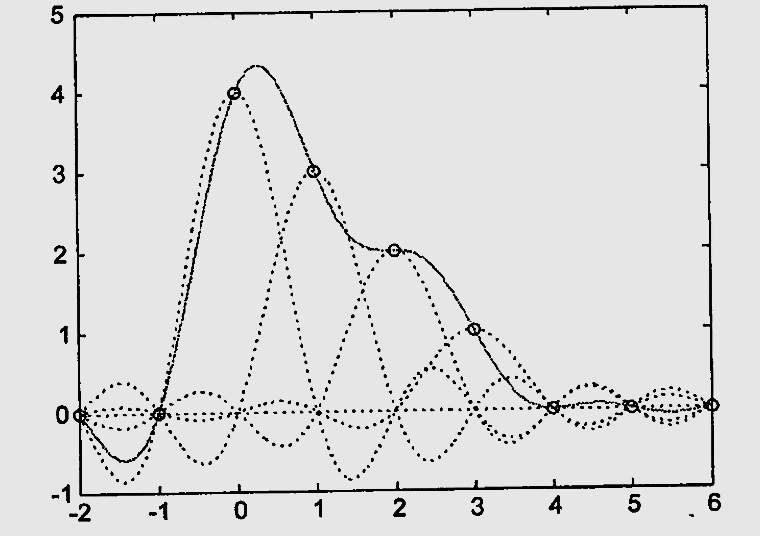
Из рисунка 0 (а) можно заметить, что расстояние между копиями равно .

*Теорема Котельникова.*

Сигнал , не содержащий гармоник с частотами, превышающими некоторого значения , может быть представлен без потери информации своими дискретными отчетами , взятыми с интервалом , удовлетворяющим условию:

При этом восстановление исходного сигнала (рис. 0 б) осуществляется по формуле, представляемой собой разложение в ряд по системе функций:

где базис Котельникова:



*Рис. 0 (б). Восстановление сигнала по его дискретным отчетам.*

*Дискретное преобразование Фурье.*

Пусть последовательность отсчетов является периодической с периодом :

Рассмотрим фрагмент последовательности из отсчетов. Например, Тогда дискретная функция

– тоже будет периодической, с периодом . Здесь – период дискретизации.

Спектр также должен быть периодическим и дискретным с расстоянием между гармониками – .

Поскольку – периодическая функция, ее можно разложить в ряд Фурье, коэффициенты которого вычисляются по формуле дискретного преобразования Фурь*е*:

Обратное дискретное преобразование Фурье:

*Свойства дискретного преобразования Фурье.*

Пусть , – дискретные последовательности с периодом и ДПФ, а ДПФ. Тогда верны следующие свойства:

* Линейность:
* Задержка:
* Симметрия:
* ДПФ произведения:
* ДПФ вычисляет дискретные отсчеты спектра дискретного сигнала:

**Постановка задачи.**

Для заданного дискретного сигнала применить фильтры, основанные на формулах численного дифференцирования и интегрирования. Полученные результаты содержательно проинтерпретировать.

**Выполнение работы.**

1. Сформируем дискретный сигнал посредством дискретизации с шагом непрерывного сигнала, представляющего собой линейную комбинацию косинусоид вида . Всего имеется одиннадцать гармоник с упорядоченными по возрастанию частотами от 0 до π, изменяющимися с шагом Δω = 0.1π. Амплитуды гармоник представляют собой целые числа со значениями от 1 до 11, определяемые случайным образом с помощью датчика равномерно распределенных случайных чисел. Начальные фазы представляют собой случайные числа в промежутке от 0 до 0.5. Дискретная последовательность включает в себя 32 отсчета (N = 31).

Исходный аналоговый сигнал:

.

Сформированный дискретный сигнал:

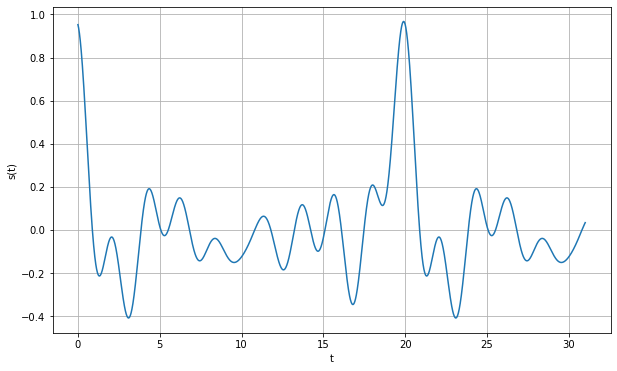
.

При помощи генератора псевдослучайных чисел были сгенерированы значения коэффициентов и (коэффициенты были нормализованы посредством деления их на сумму полученных ):

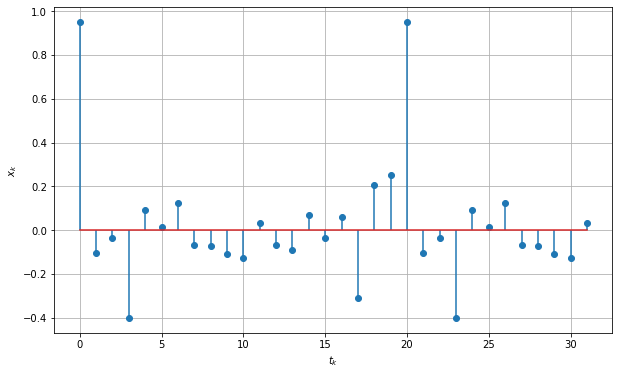
|  |  |  |  |
| --- | --- | --- | --- |
| *k* |  |  |  |
| 0 | 0.02040816 | 0 | 0.16838626 |
| 1 | 0.12244898 | 0.1π | 0.18390124 |
| 2 | 0.10204082 | 0.2π | 0.48714459 |
| 3 | 0.18367347 | 0.3π | 0.42935826 |
| 4 | 0.16326531 | 0.4π | 0.15484798 |
| 5 | 0.08163265 | 0.5π | 0.47822058 |
| 6 | 0.02040816 | 0.6π | 0.08505703 |
| 7 | 0.04081633 | 0.7π | 0.4713078 |
| 8 | 0.02040816 | 0.8π | 0.0025991 |
| 9 | 0.14285714 | 0.9π | 0.06435555 |
| 10 | 0.10204082 | π | 0.06435555 |

2. Визуализируем исходные аналоговый и дискретизированный сигналы.

Графики исходного аналогово и дискретного сигналов приведены на рисунках 1 и 2.



*Рисунок 1* *– Исходный аналоговый сигнал.*



*Рисунок 2* *– Исходный дискретный сигнал.*

3. При помощи дискретного преобразования Фурье найдем дискретные отсчеты спектра дискретного сигнала и визуализируем их.

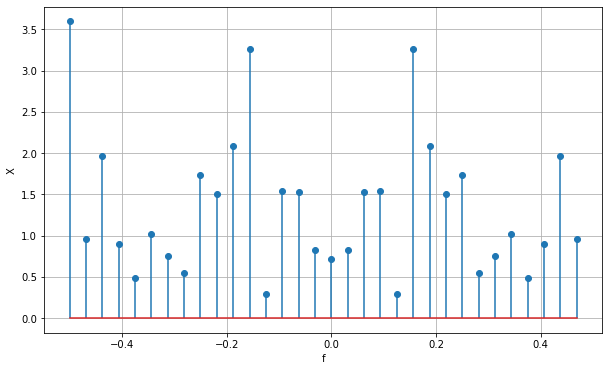
Пусть последовательность отсчетов является периодической с периодом , то есть . Рассмотрим фрагмент последовательности из отсчетов Тогда дискретная функция

также будет периодической, с периодом , где – период дискретизации.

Спектр также должен быть периодическим и дискретным с расстоянием между гармониками – .

Поскольку – периодическая функция, ее можно разложить в ряд Фурье, коэффициенты которого вычисляются по формуле дискретного преобразования Фурье:

Полученные при помощи дискретного преобразования Фурье дискретные отсчеты спектра исходного дискретного сигнала представлены на рисунке 3.



*Рисунок 3* *– Дискретные отсчеты спектра исходного 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.

4. Для дискретного сигнала применим линейное сглаживание по 5-ти и 9-ти точкам, представим формулы для передаточных функций *H(ω)* – частотной характеристики фильтра.

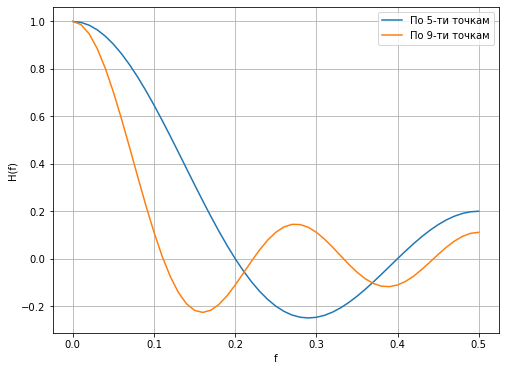
Формулы для линейного сглаживания по 5 точкам и 9 точкам:



Передаточные функции для линейного сглаживания по 5-ти и 9-ти точкам:



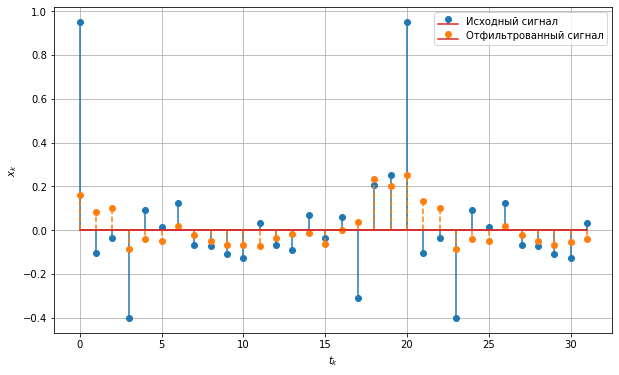
Графики передаточных функций для линейного сглаживания по 5 и 9 точкам представлены на рисунке 4.



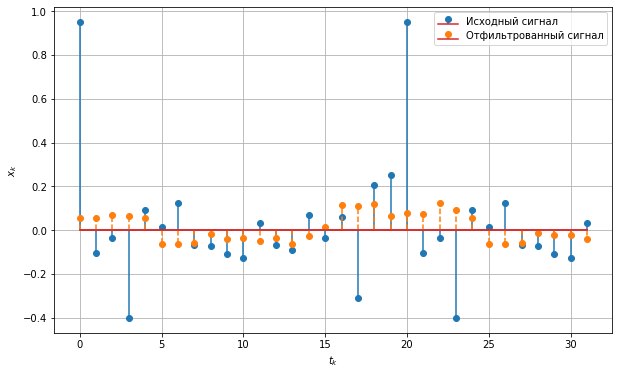
*Рисунок 3* *– Передаточная функция для линейного сглаживания по 5 и 9 точкам.*

5. Визуализируем полученные после фильтрации дискретные сигналы совместно с исходным дискретным сигналом.

Графики исходного сигнала и сигнала после применения линейного сглаживания по 5-ти и по 9-ти точкам представлены на рисунке 5 и 6.



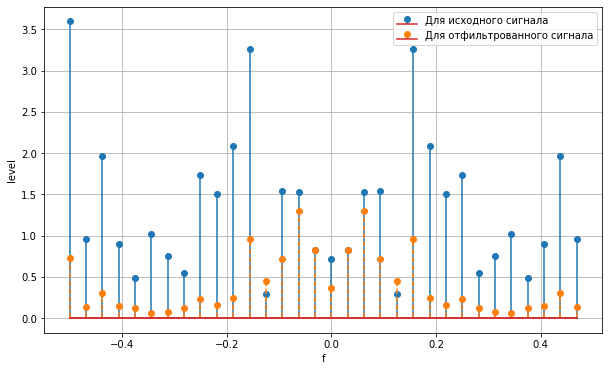
*Рисунок 5* *– Исходный дискретный сигнал и сигнал после применения сглаживания по 5-ти точкам.*



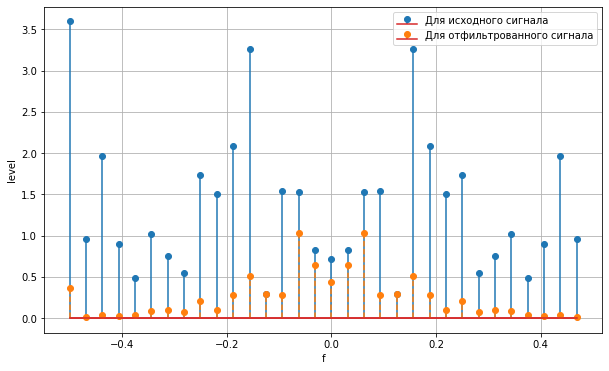
*Рисунок 6* *– Исходный дискретный сигнал и сигнал после применения сглаживания по 9-ти точкам.*

6. При помощи ДПФ найдем дискретные отсчеты спектра дискретного сигнала после его фильтрации и визуализируем их совместно с отчетами спектра исходного дискретного сигнала.

Графики дискретных отсчетов спектра, полученных при помощи ДПФ, для исходного сигнала и сигнала после фильтрации представлены на рисунке 7 и 8.



*Рисунок 7* *– Дискретные отсчеты спектра для исходного дискретного сигнала и сигнала после применения сглаживания по 5-ти точкам.*



*Рисунок 8* *– Дискретные отсчеты спектра для исходного дискретного сигнала и сигнала после применения сглаживания по 9-ти точкам.*

7. Проанализируем результаты на соответствие значениям соответствующих передаточных функций *H(ω)*:

Из полученного спектра видно, что без ослабления пропускается только сигнал постоянного уровня (нулевой частоты). Сигналы с частотами, близкими к 0, ослабевают не сильно, а сигнал с большими частотами значительно ослабевает. С увеличением числа точек полоса пропускания становится меньше. Графики передаточных функций, представленные на рисунке 3, подтверждают данный вывод.

8. Повторим действия из пунктов 4 – 7 для других фильтров.

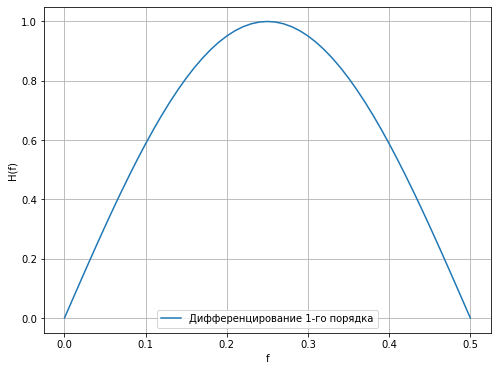
***А. Дискретный фильтр, соответствующий численному дифференцированию 1-го порядка:***

Формула для дискретного фильтра, соответствующего численному дифференцированию 1-го порядка:



Передаточная функция для дискретного фильтра, соответствующего численному дифференцированию 1-го порядка:

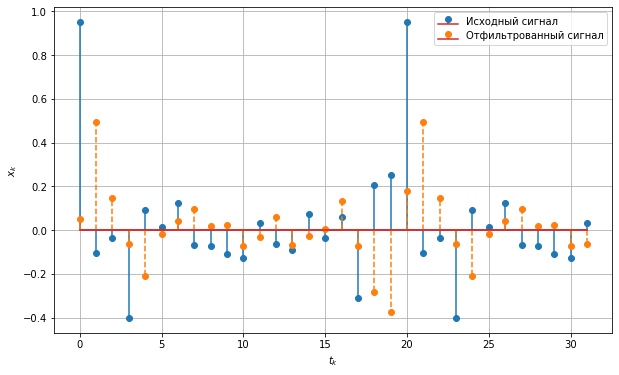
График передаточной функции для дискретного фильтра, соответствующего численному дифференцированию 1-го порядка, представлен на рисунке 9.



*Рисунок 9* *– Передаточная функция для дискретного фильтра, соответствующего численному дифференцированию 1-го порядка.*

Визуализируем полученные после фильтрации дискретные сигналы совместно с исходным дискретным сигналом.

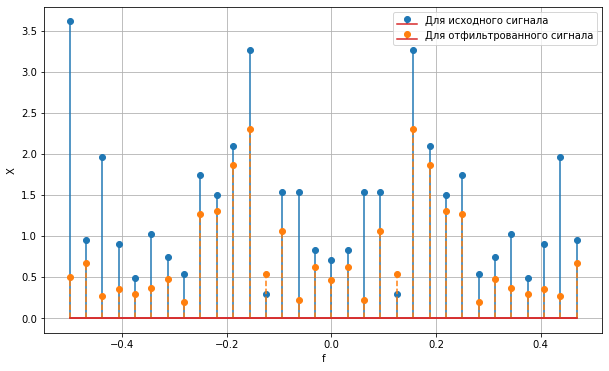
График исходного сигнала и сигнала после применения численного дифференцирования 1-го порядка представлен на рисунке 10.



*Рисунок 10* *– Исходный дискретный сигнал и сигнал после применения сглаживания полиномом второй степени по 5-ти точкам.*

При помощи ДПФ найдем дискретные отсчеты спектра дискретного сигнала после его фильтрации и визуализируем их совместно с отчетами спектра исходного дискретного сигнала.

График дискретных отсчетов спектра, полученных при помощи ДПФ, для исходного сигнала и сигнала после фильтрации представлен на рисунке 11.



*Рисунок 11* *– Дискретные отсчеты спектра для исходного дискретного сигнала и сигнала после применения численного дифференцирования 1-го порядка.*

Проанализируем результаты на соответствие значениям соответствующих передаточных функций *H(ω)*:

Из графика передаточной функции на рисунке 9 видно, что рассматриваемый фильтр подавляет низкие и высокие частоты. При этом средние частоты остаются подавляются незначительно.

Видно, что график передаточной функции, представленный на рисунке 9, объясняют изменение амплитуд в спектре сигнала.

***Б. Дискретный фильтр, соответствующий численному интегрированию (прямоугольников, трапеций, Симпсона):***

Далее приведены формулы для дискретного фильтра, соответствующего численному интегрированию.

Формула прямоугольников:



Формула трапеций:



Формула Симпсона:

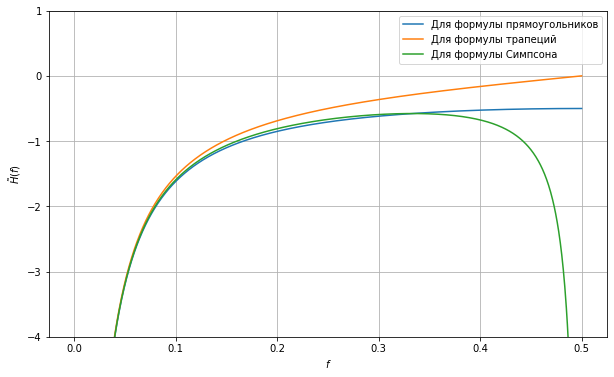


Передаточная функция для численного интегрирования по формуле прямоугольников:

Передаточная функция для численного интегрирования по формуле трапеций:

Передаточная функция для численного интегрирования по формуле Симпсона:

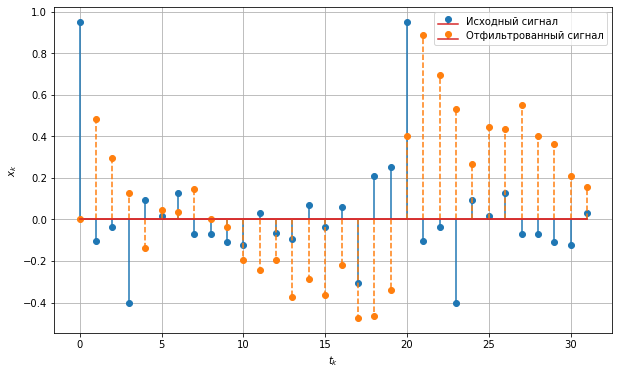
Графики передаточных функций для дискретного фильтра, соответствующего численному интегрированию, представлены на рисунке 12.



*Рисунок 12* *– Передаточная функция для дискретного фильтра, соответствующего численному интегрированию по формулам прямоугольников, трапеций и Симпсона.*

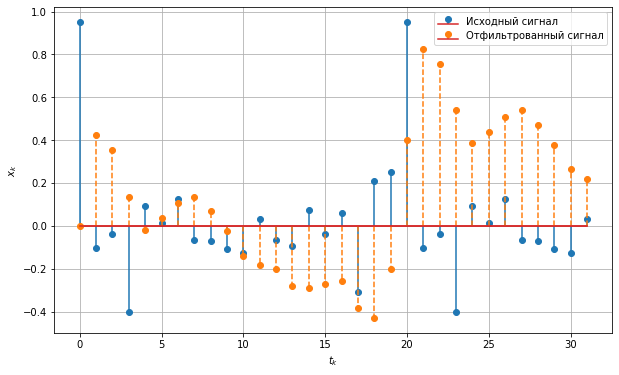
Визуализируем полученные после фильтрации дискретные сигналы совместно с исходным дискретным сигналом.

График исходного сигнала и сигнала после применения численного интегрирования по формуле прямоугольников представлен на рисунке 13.



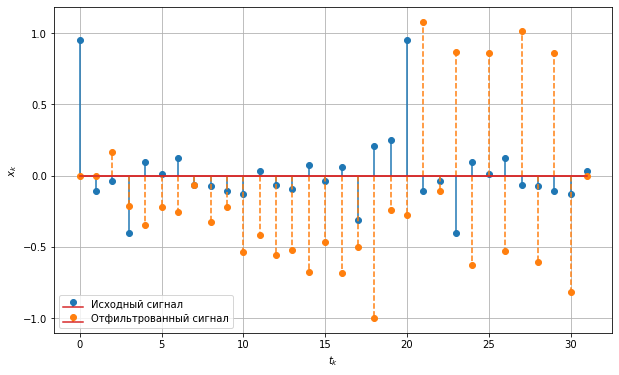
*Рисунок 13* *– Исходный дискретный сигнал и сигнал* после *численного интегрирования по формуле прямоугольников.*

График исходного сигнала и сигнала после применения численного интегрирования по формуле трапеций представлен на рисунке 14.



*Рисунок 14* *– Исходный дискретный сигнал и сигнал* после *численного интегрирования по формуле трапеций.*

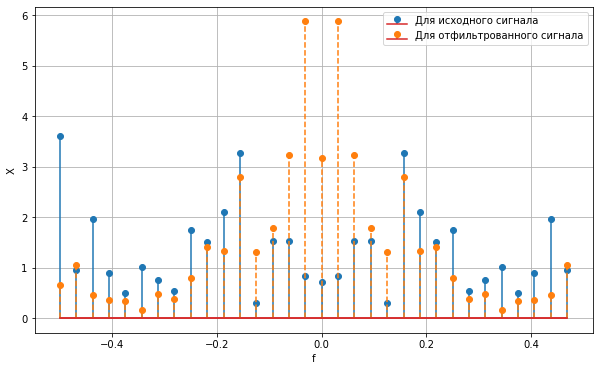
График исходного сигнала и сигнала после применения численного интегрирования по формуле Симпсона представлен на рисунке 15.



*Рисунок 15* *– Исходный дискретный сигнал и сигнал* после *численного интегрирования по формуле Симпсона.*

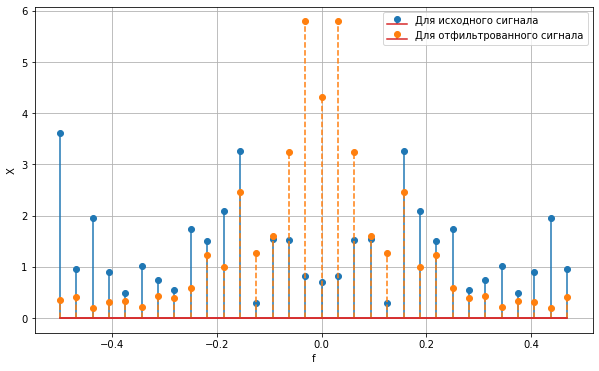
При помощи ДПФ найдем дискретные отсчеты спектра дискретного сигнала после его фильтрации и визуализируем их совместно с отчетами спектра исходного дискретного сигнала.

График дискретных отсчетов спектра, полученных при помощи ДПФ, для исходного сигнала и сигнала после применения численного интегрирования по формуле прямоугольников представлен на рисунке 16.



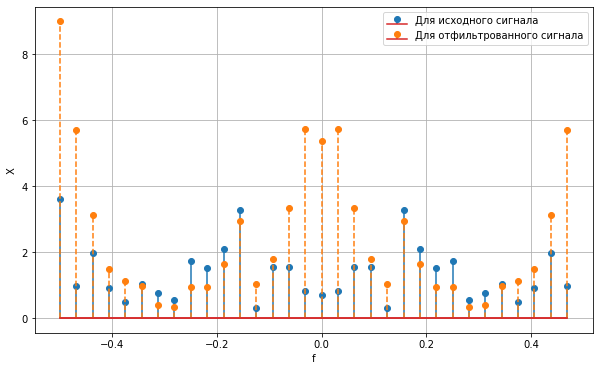
*Рисунок 16* *– Дискретные отсчеты спектра для исходного дискретного сигнала и сигнала после применения численного интегрирования по формуле прямоугольников.*

График дискретных отсчетов спектра, полученных при помощи ДПФ, для исходного сигнала и сигнала после применения численного интегрирования по формуле трапеций представлен на рисунке 17.



*Рисунок 17* *– Дискретные отсчеты спектра для исходного дискретного сигнала и сигнала после применения численного интегрирования по формуле трапеций.*

График дискретных отсчетов спектра, полученных при помощи ДПФ, для исходного сигнала и сигнала после применения численного интегрирования по формуле Симпсона представлен на рисунке 18.



*Рисунок 18* *– Дискретные отсчеты спектра для исходного дискретного сигнала и сигнала после применения численного интегрирования по формуле Симпсона.*

Проанализируем результаты на соответствие значениям соответствующих передаточных функций *H(ω)*:

Из графика передаточных функций на рисунке 12 видно, что численное интегрирование по формуле трапеций и по формуле прямоугольников значительно усиливает низкие частоты, а средние и высокие – подавляет. У формулы трапеций понижение средних и высоких частот больше, чем у формулы прямоугольников. В отличии от формул трапеций и прямоугольников, формула Симпсона значительно усиливает низкие и высокие частоты, а средние – подавляет.

Видно, что графики передаточных функций, представленные на рисунке 12, объясняют изменение амплитуд в спектре сигнала.

**Выводы.**

В ходе выполнения лабораторной работы была проведена фильтрация дискретных последовательностей при помощи рекурсивных фильтров, основанных на формулах численного дифференцирования и интегрировании, а также был произведен анализ получаемых результатов при помощи дискретного преобразования Фурье (ДПФ).

В результате был сгенерирован аналоговый сигнал, после чего он был дискретизирован. Для полученного дискретного сигнала был построен спектр, представленный в виде набора дискретных отсчетов. Было определено, что спектр показывает наличие в исходном сигнале множества различных частот.

Были применены фильтры линейного сглаживания, а также дискретные фильтры, соответствующие численному дифференцированию 1-го порядка и численному интегрированию, произведенному по методам прямоугольников, трапеций и парабол (метод Симпсона).

В результате по спектру было определено, что рассматриваемый фильтр, соответствующий дифференцированию 1-го порядка, имеет полосу пропускания в области средних частот и уменьшает амплитуду низких и высоких частот. Фильтры, соответствующие численному интегрированию по формуле трапеций и по формуле прямоугольников значительно усиливают низкие частоты, а средние и высокие – подавляют. У формулы трапеций понижение средних и высоких частот больше, чем у формулы прямоугольников. В отличии от формул трапеций и прямоугольников, формула Симпсона значительно усиливает низкие и высокие частоты, а средние – подавляет.

Приложение А

исходный код программы

import math

import numpy as np

import matplotlib.pyplot as plt

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

N = 31

T = 1

analog\_t\_values = np.arange(0, N + 0.01, 0.01)

t\_values = np.arange(0, N + T, T)

w\_k = np.arange(0, np.pi + 0.1 \* np.pi, 0.1 \* np.pi)

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

A\_k = A\_k / sum(A\_k)

fi\_k = np.random.random(size=11) / 2

print(w\_k)

print(A\_k)

print(fi\_k)

# [0. 0.31415927 0.62831853 0.9424778 1.25663706 1.57079633 1.88495559 2.19911486 2.51327412 2.82743339 3.14159265]

# [0.02040816 0.12244898 0.10204082 0.18367347 0.16326531 0.08163265 0.02040816 0.04081633 0.02040816 0.14285714 0.10204082]

# [0.16838626 0.18390124 0.48714459 0.42935826 0.15484798 0.47822058 0.08505703 0.4713078 0.0025991 0.06435555 0.11157867]

def s(t):

r = 0

for i in range(11):

r += A\_k[i] \* math.cos(w\_k[i] \* t + fi\_k[i])

return r

plt.figure(figsize=(10, 6))

plt.plot(analog\_t\_values, [s(t) for t in analog\_t\_values])

plt.ylabel("s(t)")

plt.xlabel("t")

plt.grid()

plt.show()

plt.figure(figsize=(10, 6))

plt.stem(t\_values, [s(t) for t in t\_values])

plt.ylabel(r"$x\_k$")

plt.xlabel(r"$t\_k$")

plt.grid()

plt.show()

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)

f\_values = fftfreq(N + 1, T)

X\_values = np.abs(dft([s(t) for t in t\_values]))

plt.figure(figsize=(10, 6))

plt.stem(f\_values, X\_values)

plt.ylabel("X")

plt.xlabel("f")

plt.grid()

plt.show()

x\_values = [s(t) for t in t\_values]

x\_values\_5 = np.convolve(x\_values, np.ones(5), 'same') / 5

x\_values\_9 = np.convolve(x\_values, np.ones(9), 'same') / 9

plt.figure(figsize=(10, 6))

plt.stem(t\_values, x\_values, label="Исходный сигнал")

plt.stem(t\_values, x\_values\_5, linefmt="C1--", markerfmt="C1o", label="Отфильтрованный сигнал")

plt.ylabel(r"$x\_k$")

plt.xlabel(r"$t\_k$")

plt.legend()

plt.grid()

plt.show()

plt.figure(figsize=(10, 6))

plt.stem(t\_values, x\_values, label="Исходный сигнал")

plt.stem(t\_values, x\_values\_9, linefmt="C1--", markerfmt="C1o", label="Отфильтрованный сигнал")

plt.ylabel(r"$x\_k$")

plt.xlabel(r"$t\_k$")

plt.legend()

plt.grid()

plt.show()

f\_values = fftfreq(N + 1, T)

X\_values\_5 = np.abs(dft(x\_values\_5))

plt.figure(figsize=(10, 6))

plt.stem(f\_values, X\_values, label="Для исходного сигнала")

plt.stem(f\_values, X\_values\_5, linefmt="C1--", markerfmt="C1o", label="Для отфильтрованного сигнала")

plt.ylabel("X")

plt.xlabel("f")

plt.legend()

plt.grid()

plt.show()

f\_values = fftfreq(N + 1, T)

X\_values\_9 = np.abs(dft(x\_values\_9))

plt.figure(figsize=(10, 6))

plt.stem(f\_values, X\_values, label="Для исходного сигнала")

plt.stem(f\_values, X\_values\_9, linefmt="C1--", markerfmt="C1o", label="Для отфильтрованного сигнала")

plt.ylabel("X")

plt.xlabel("f")

plt.legend()

plt.grid()

plt.show()

def h\_5(f):

return (1 + 2 \* math.cos(2 \* math.pi \* f) + 2 \* math.cos(4 \* math.pi \* f)) / 5

def h\_9(f):

return (1 + 2 \* math.cos(2 \* math.pi \* f) + 2 \* math.cos(4 \* math.pi \* f) + 2 \* math.cos(6 \* math.pi \* f) + 2 \* math.cos(8 \* math.pi \* f)) / 9

f\_values\_for\_h = np.arange(0, 0.5 + 0.01, 0.01)

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

plt.plot(f\_values\_for\_h, [h\_5(f) for f in f\_values\_for\_h], label="По 5-ти точкам")

plt.plot(f\_values\_for\_h, [h\_9(f) for f in f\_values\_for\_h], label="По 9-ти точкам")

plt.ylabel("H(f)")

plt.xlabel("f")

plt.legend()

plt.grid()

plt.show()

def h\_d\_1(f):

return (1j \* math.sin(2 \* math.pi \* f)).imag

f\_values\_for\_h = np.arange(0, 0.5 + 0.01, 0.01)

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

plt.plot(f\_values\_for\_h, [h\_d\_1(f) for f in f\_values\_for\_h], label="Дифференцирование 1-го порядка")

plt.ylabel("H(f)")

plt.xlabel("f")

plt.legend()

plt.grid()

plt.show()

x\_values = [s(t) for t in t\_values]

x\_values\_d1 = np.convolve(x\_values, np.array([-1, 0, 1]), 'same') / 2

plt.figure(figsize=(10, 6))

plt.stem(t\_values, x\_values, label="Исходный сигнал")

plt.stem(t\_values, x\_values\_d1, linefmt="C1--", markerfmt="C1o", label="Отфильтрованный сигнал")

plt.ylabel(r"$x\_k$")

plt.xlabel(r"$t\_k$")

plt.legend()

plt.grid()

plt.show()

f\_values = fftfreq(N + 1, T)

X\_values\_d1 = np.abs(dft(x\_values\_d1))

plt.figure(figsize=(10, 6))

plt.stem(f\_values, X\_values, label="Для исходного сигнала")

plt.stem(f\_values, X\_values\_d1, linefmt="C1--", markerfmt="C1o", label="Для отфильтрованного сигнала")

plt.ylabel("X")

plt.xlabel("f")

plt.legend()

plt.grid()

plt.show()

def h\_rect(f):

return (1 / (2j\*np.sin(math.pi\*f))).imag

def h\_trapezoid(f):

return (np.cos(math.pi\*f) / (2j\*math.sin(math.pi\*f))).imag

def h\_simpson(f):

return ((np.cos(2\*math.pi\*f)+2)/ (3j\*math.sin(2\*math.pi\*f))).imag

f\_values\_for\_h = np.arange(0.01, 0.5 + 0.01, 0.01)

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

plt.plot(f\_values\_for\_h, [h\_rect(f) for f in f\_values\_for\_h], label="Формула прямоугольников")

plt.plot(f\_values\_for\_h, [h\_trapezoid(f) for f in f\_values\_for\_h], label="Формула трапеций")

plt.plot(f\_values\_for\_h, [h\_simpson(f) for f in f\_values\_for\_h], label="Формула Симпсона")

plt.ylabel("H(f)")

plt.xlabel("f")

plt.ylim((-4, 1))

plt.legend()

plt.grid()

plt.show()

def rectangle():

x\_values = [s(t) for t in t\_values]

y\_values = [0 for \_ in x\_values]

for i in range(0, len(y\_values) - 1):

y\_values[i + 1] = y\_values[i] + s(t\_values[i] + T/2)

return y\_values

def trapezoid():

x\_values = [s(t) for t in t\_values]

y\_values = [0 for \_ in x\_values]

for i in range(0, len(y\_values) - 1):

y\_values[i + 1] = y\_values[i] + (x\_values[i] + x\_values[i + 1]) / 2

return y\_values

def simpson():

x\_values = [s(t) for t in t\_values]

y\_values = [0 for \_ in x\_values]

for i in range(1, len(y\_values)-2):

y\_values[i + 1] = y\_values[i - 1] + (x\_values[i-1] + 4\*x\_values[i] + x\_values[i+1]) / 3

return y\_values

x\_values = [s(t) for t in t\_values]

x\_values\_rect = rectangle()

x\_values\_trap = trapezoid()

x\_values\_simpson = simpson()

plt.figure(figsize=(10, 6))

plt.stem(t\_values, x\_values, label="Исходный сигнал")

plt.stem(t\_values, x\_values\_rect, linefmt="C1--", markerfmt="C1o", label="Отфильтрованный сигнал")

plt.ylabel(r"$x\_k$")

plt.xlabel(r"$t\_k$")

plt.legend()

plt.grid()

plt.show()

plt.figure(figsize=(10, 6))

plt.stem(t\_values, x\_values, label="Исходный сигнал")

plt.stem(t\_values, x\_values\_trap, linefmt="C1--", markerfmt="C1o", label="Отфильтрованный сигнал")

plt.ylabel(r"$x\_k$")

plt.xlabel(r"$t\_k$")

plt.legend()

plt.grid()

plt.show()

plt.figure(figsize=(10, 6))

plt.stem(t\_values, x\_values, label="Исходный сигнал")

plt.stem(t\_values, x\_values\_simpson, linefmt="C1--", markerfmt="C1o", label="Отфильтрованный сигнал")

plt.ylabel(r"$x\_k$")

plt.xlabel(r"$t\_k$")

plt.legend()

plt.grid()

plt.show()

f\_values = fftfreq(N + 1, T)

X\_values\_d1 = np.abs(dft(x\_values\_rect))

plt.figure(figsize=(10, 6))

plt.stem(f\_values, X\_values, label="Для исходного сигнала")

plt.stem(f\_values, X\_values\_d1, linefmt="C1--", markerfmt="C1o", label="Для отфильтрованного сигнала")

plt.ylabel("X")

plt.xlabel("f")

plt.legend()

plt.grid()

plt.show()

f\_values = fftfreq(N + 1, T)

X\_values\_d1 = np.abs(dft(x\_values\_trap))

plt.figure(figsize=(10, 6))

plt.stem(f\_values, X\_values, label="Для исходного сигнала")

plt.stem(f\_values, X\_values\_d1, linefmt="C1--", markerfmt="C1o", label="Для отфильтрованного сигнала")

plt.ylabel("X")

plt.xlabel("f")

plt.legend()

plt.grid()

plt.show()

f\_values = fftfreq(N + 1, T)

X\_values\_d1 = np.abs(dft(x\_values\_simpson))

plt.figure(figsize=(10, 6))

plt.stem(f\_values, X\_values, label="Для исходного сигнала")

plt.stem(f\_values, X\_values\_d1, linefmt="C1--", markerfmt="C1o", label="Для отфильтрованного сигнала")

plt.ylabel("X")

plt.xlabel("f")

plt.legend()

plt.grid()

plt.show()