ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

канд. техн. наук, доцент		А.В. Аграновский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ

по курсу: ЦИФРОВАЯ ОБРАБОТКА И ПЕРЕДАЧА СИГНАЛОВ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4329		Д.С. Шаповалова
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

1. Цель работы:

Изучение особенностей дискретного преобразования Фурье.

2. Задание:

1. Разработать функцию на языке высокого уровня, вычисляющую прямое ДПФ входного сигнала (вектора) и рисующую графики действительной и мнимой частей результата преобразования. Сравнить результаты ДПФ при использовании прямоугольного окна, окон Ханна и Хемминга.
2. Разработать функцию на языке высокого уровня, формирующую из действительной и мнимой частей результата ДПФ амплитудный и фазовый спектры входного сигнала и их графики.
3. Произвести декодирование аудио-файла с записью тонального сигнала (Dual-Tone Multi-Frequency (DTMF)) сигнала в формате WAV PCM 16 bit, mono. Данный способ кодирования предполагает, что кодируемое значение представляется в виде пары различных частот (f1/f2) в соответствии с приведенными значениями на рисунке 1.1. Затем, сигнал представляется в виде отсчетов суммы двух синусоид соответствующих частот. Для декодирования сигнала необходимо произвести прямое дискретное преобразование для имеющегося набора отсчетов и определить частоты используемых для кодирования синусоид по полученному амплитудному спектру.

(f_1/f_2)	1209 Гц	1336 Гц	1477 Гц	1633 Гц
697 Гц	1	2	3	A
770 Гц	4	5	6	B
852 Гц	7	8	9	C
941 Гц	*	0	#	D

Рисунок 1.1 – Соответствие частот f1/f2 и кодируемых значений

4. Разработать функцию на языке высокого уровня, осуществляющую обратное ДПФ с целью восстановления входного сигнала.

3. Теоретические сведения:

1. Дискретное преобразование Фурье – ДПФ

Дискретное преобразование Фурье является линейным оператором, который преобразует конечную выборку отсчётов сигнала во множество комплексных спектральных коэффициентов. По сути, ДПФ выполняет разложение исходного сигнала по ортогональному базису комплексных экспонент — дискретных гармоник. Каждая гармоника имеет определённую частоту, а результатом преобразования является набор весов (амплитуд и фаз), с которыми эти гармоники участвуют в формировании исходного сигнала.

Для сигнала $x[n]$, $n = 0,1, \dots, N-1$, ДПФ определяется формулой:

$$\mathrm{X[k]} = \sum_{\mathrm{n=0}}^{\mathrm{N-1}} \mathrm{x[n]}\, \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\mathrm{N}}\mathrm{kn}},\ \mathrm{k} = 0,1, \dots, \mathrm{N}-1, \qquad (1)$$

Каждый спектральный коэффициент $X[k]$— это мера того, насколько исходный сигнал коррелирует с комплексной экспонентой частоты $\frac{k}{N}$.

Другими словами:

- для каждого k ДПФ «примеряет» на входной сигнал гармонику частоты k,
- вычисляет, насколько сильно сигнал содержит эту гармонику,
- и записывает результаты в виде комплексного числа (амплитуда + фаза).

Базис дискретных гармоник

Комплексная экспонента

$$\mathrm{E}^{-\mathrm{j}\frac{2\pi}{\mathrm{N}}\mathrm{kn}}, \qquad (2)$$

представляет собой дискретную гармоническую функцию (дискретную синусоиду и косинус одновременно), причём:

- индекс n— номер отсчёта во времени,
- индекс k— номер частоты,
- частота гармоники (в нормированных единицах) равна $\frac{k}{N}$ периода на отсчёт.

Все такие гармоники для $k = 0 \dots N-1$образуют ортогональный базис в пространстве комплексных последовательностей длины N, то есть:

- каждая гармоника не может быть представлена как комбинация других,
- их можно использовать для точного разложения любого сигнала длины N.

Таким образом ДПФ — это то же самое, что разложение произвольного вектора по ортонормированному базису.

2. Вычисление ДПФ:

1) Умножение сигнала на комплексную гармонику

Для каждой частоты kвесь сигнал перемножается с экспонентой.

Это аналогично скалярному произведению сигнала и ортогонального базисного вектора.

2) Суммирование результата

Сумма показывает степень подобия двух временных последовательностей.

3) Получение комплексного коэффициента

Полученный $X[k]$— комплексное число:

- действительная часть описывает вклад косинусной компоненты,
- мнимая часть — вклад синусной компоненты,
- модуль $|\ X[k]\ |$показывает амплитуду данной гармоники,
- аргумент $\arg X[k]$показывает фазу.

Таким образом каждый $X[k]$— это информация о присутствии в сигнале гармоники частоты k.

3. Структура комплексных чисел в спектре

Каждый спектральный коэффициент:

$$X[k] = A_k + jB_k, \tag{3}$$

имеет две интерпретации:

Алгебраическая:

$\mathrm{Re}(X[k])$ — косинусный вклад,

$\mathrm{Im}(X[k])$ — синусный вклад.

Полярная:

амплитуда:

$$|\ X[k]\ | = \sqrt{A_k^2 + B_k^2}, \tag{4}$$

фаза:

$$\varphi_k = \arctan\left(\frac{B_k}{A_k}\right), \tag{5}$$

Эти две характеристики используются при построении амплитудного и фазового спектров.

Каждый коэффициент X[k] представляет собой меру соответствия сигнала x[n] комплексной экспоненте частоты $\frac{\mathrm{k}}{\mathrm{N}}$. Действительная часть Re(X[k]) соответствует вкладу косинусоидальной составляющей, а мнимая часть Im(X[k]) – синусоидальной.

Амплитудный спектр вычисляется как модуль комплексного коэффициента:

$$|\ \mathrm{X[k]}\ | = \sqrt{\mathrm{Re(X[k])}^2 + \mathrm{Im(X[k])}^2}, \tag{6}$$

а фазовый спектр – как аргумент:

$$\varphi[k] = \arctan\left(\frac{\mathrm{Im}(X[k])}{\mathrm{Re}(X[k])}\right), \quad (7)$$

Обратное ДПФ (ОДПФ) позволяет восстановить исходный сигнал по его спектру:

$$X[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] \cdot e^{j\frac{2\pi}{N}kn}, n = 0,1,\dots,N-1, \quad (8)$$

4. Эффект растекания спектра и оконные функции

При выполнении ДПФ предполагается, что анализируемый сигнал является периодическим. Если начальные и конечные значения сигнала не совпадают, на границах периода возникают скачки, что приводит к явлению растекания спектра – появлению ложных частотных составляющих.

Для уменьшения этого эффекта используются оконные функции, которые плавно уменьшают амплитуду сигнала на краях отрезка. Наиболее распространенные (использующиеся в работе) окна:

Прямоугольное окно (отсутствие взвешивания):

$$w[n] = 1, \quad (9)$$

Обладает наилучшим частотным разрешением, но максимальным уровнем боковых лепестков.

Окно Ханна:

$$w[n] = 0.5(1 - \cos(\frac{2\pi n}{N-1})), \quad (10)$$

Эффективно подавляет боковые лепестки, но ухудшает разрешение.

Окно Хэмминга:

$$w[n] = 0.53836 - 0.46164\cos(\frac{2\pi n}{N-1}), \quad (11)$$

Компромиссный вариант с хорошим подавлением боковых лепестков и приемлемым разрешением.

5. Применение ДПФ для декодирования DTMF-сигналов

DTMF (Dual-Tone Multi-Frequency) – это система кодирования символов с помощью двух тоновых частот. Каждому символу соответствует пара частот:

- одна из низкочастотной группы (697, 770, 852, 941 Гц),
- одна из высокочастотной группы (1209, 1336, 1477, 1633 Гц).

Для декодирования DTMF-сигнала:

1. Сигнал разбивается на отрезки, соответствующие длительности одного символа.
2. Для каждого отрезка вычисляется ДПФ.
3. По амплитудному спектру определяются две наиболее выраженные частоты.

4. По таблице соответствия частот и символов восстанавливается исходное значение.

6. Свойства ДПФ

Линейность:

$$ax_1[n] + bx_2[n] \leftrightarrow aX_1[k] + bX_2[k], \quad (12)$$

Сдвиг во времени:

$$x[n-m] \leftrightarrow X[k] \cdot e^{-j\frac{2\pi}{N}km}, \quad (13)$$

Симметрия спектра для вещественного сигнала:

$$X[N-k] = X^*[k], \quad (14)$$

Теорема Парсеваля:

$$\sum_{n=0}^{N-1} | x[n] |^2 = \frac{1}{N} \sum_{k=0}^{N-1} | X[k] |^2, \quad (15)$$

4. Выполнение задания:

В рамках первого этапа лабораторной работы была разработана программа на языке Python, позволяющая исследовать влияние различных оконных функций на спектральный анализ дискретного сигнала. Основной задачей данного этапа являлась реализация вычисления прямого дискретного преобразования Фурье (ДПФ) и визуализация спектральных характеристик сигнала при использовании различных типов окон.

На первом шаге был сформирован тестовый дискретный сигнал. В качестве исследуемого сигнала использовалась синусоидальная последовательность. Исходный сигнал представлен на рисунке 2.1:

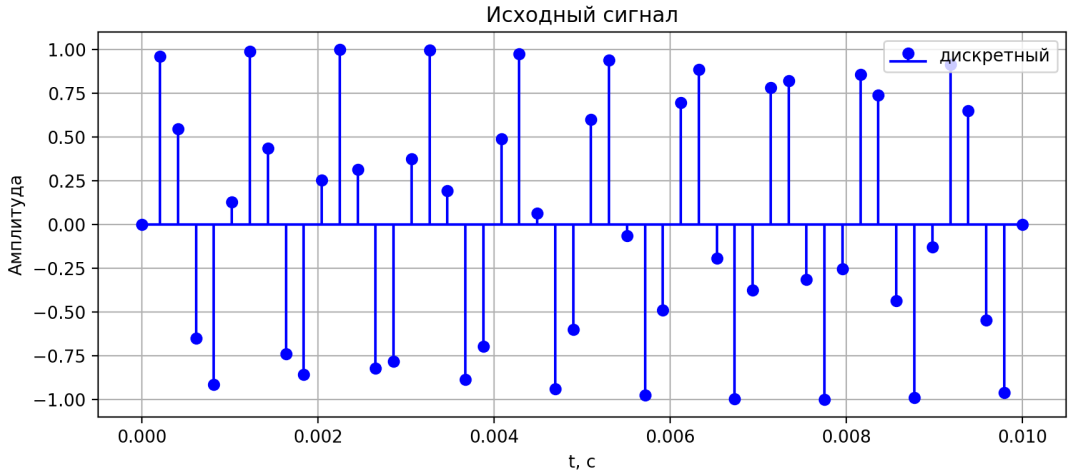

Рисунок 2.1 – Входной сигнал

Для каждого из окон выполнялись следующие шаги:

1) формирование оконной функции той же длины, что и входной сигнал;
2) перемножение окна и исследуемой последовательности;
3) вычисление ДПФ полученного сигнала;
4) вывод спектральных данных в виде действительной и мнимой частей.

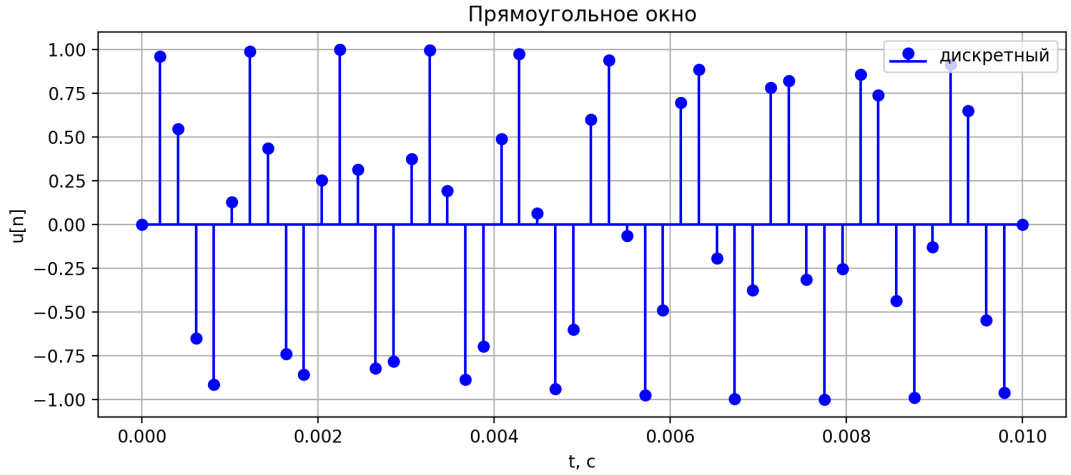

Рисунок 2.2.1 – Сигнал после применения прямоугольного окна

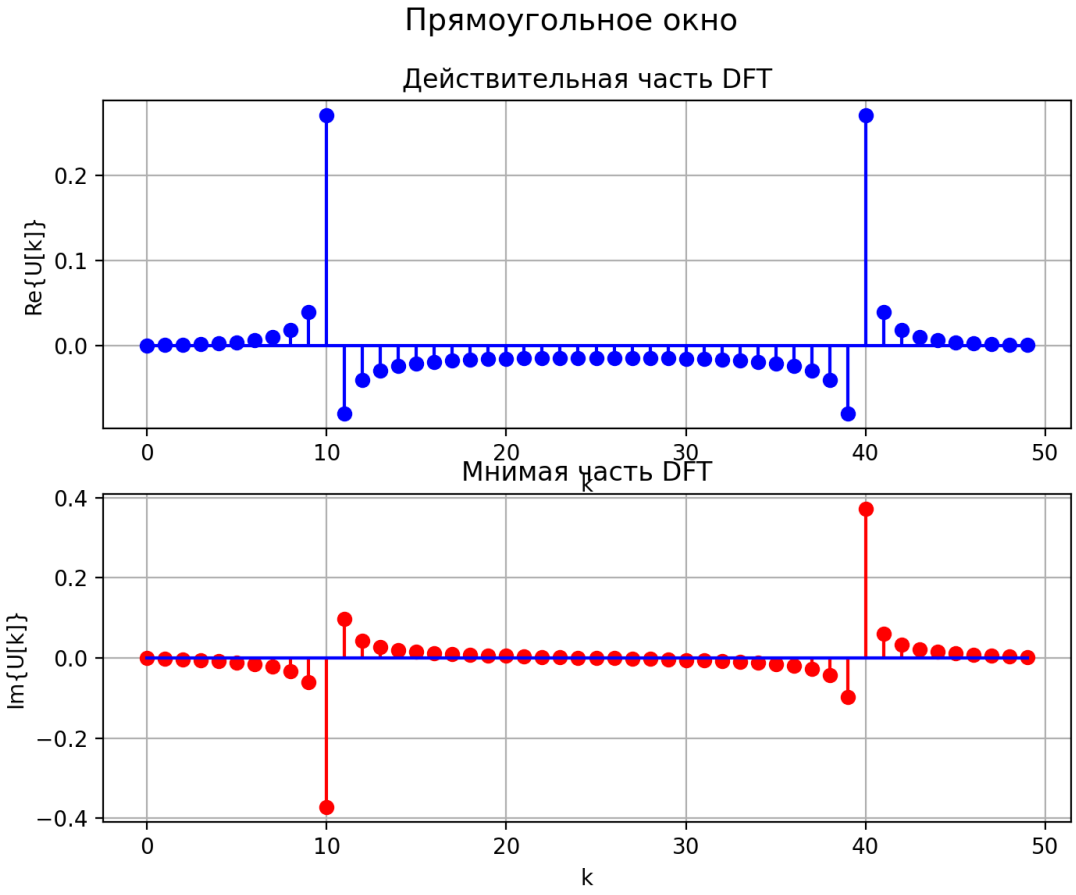

Рисунок 2.2.2 – Результаты ДПФ с использованием прямоугольного окна

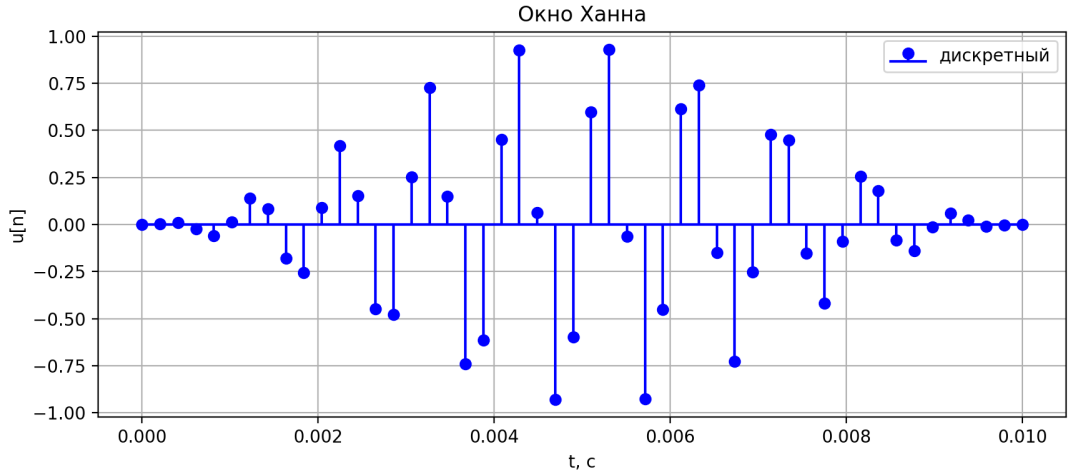

Рисунок 2.3.1 - Сигнал после применения окна Ханна

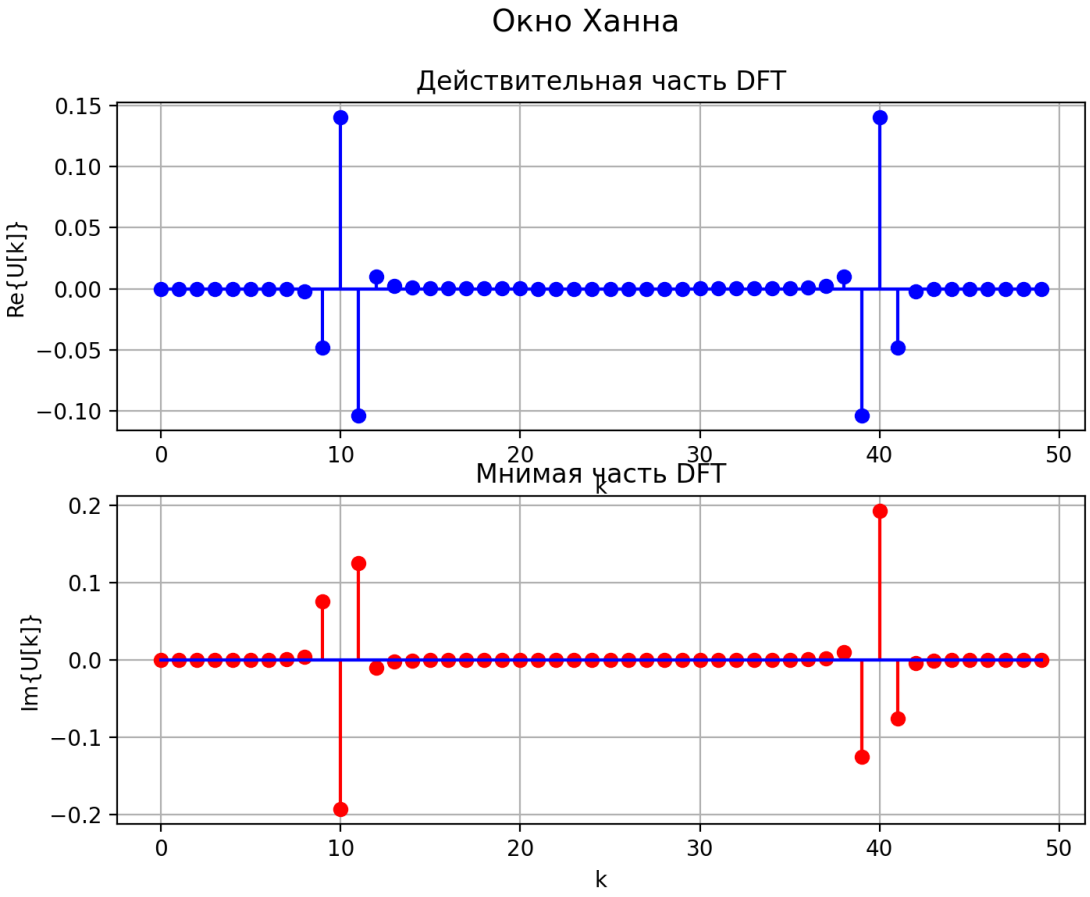

Рисунок 2.3.2 – Результаты ДПФ при использовании окна Ханна

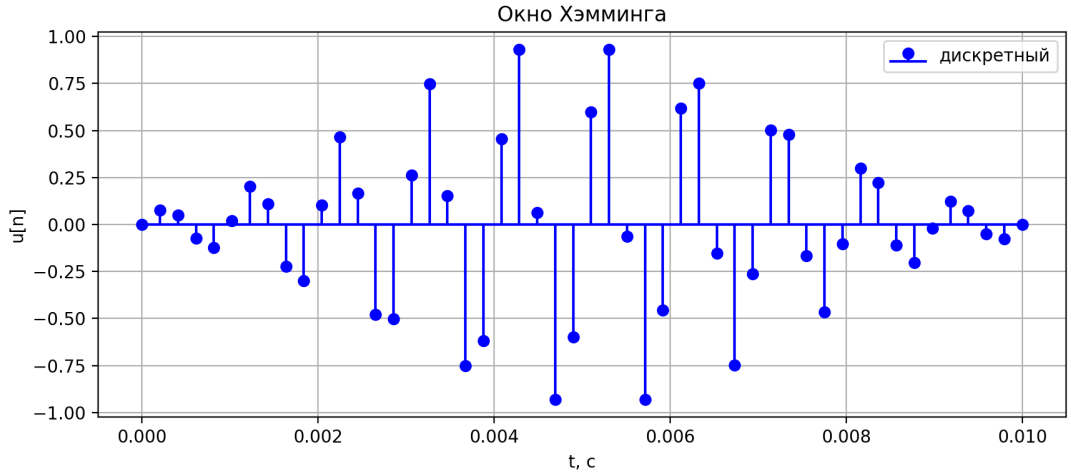

Рисунок 2.4.1 – Сигнал после использования окна Хемминга

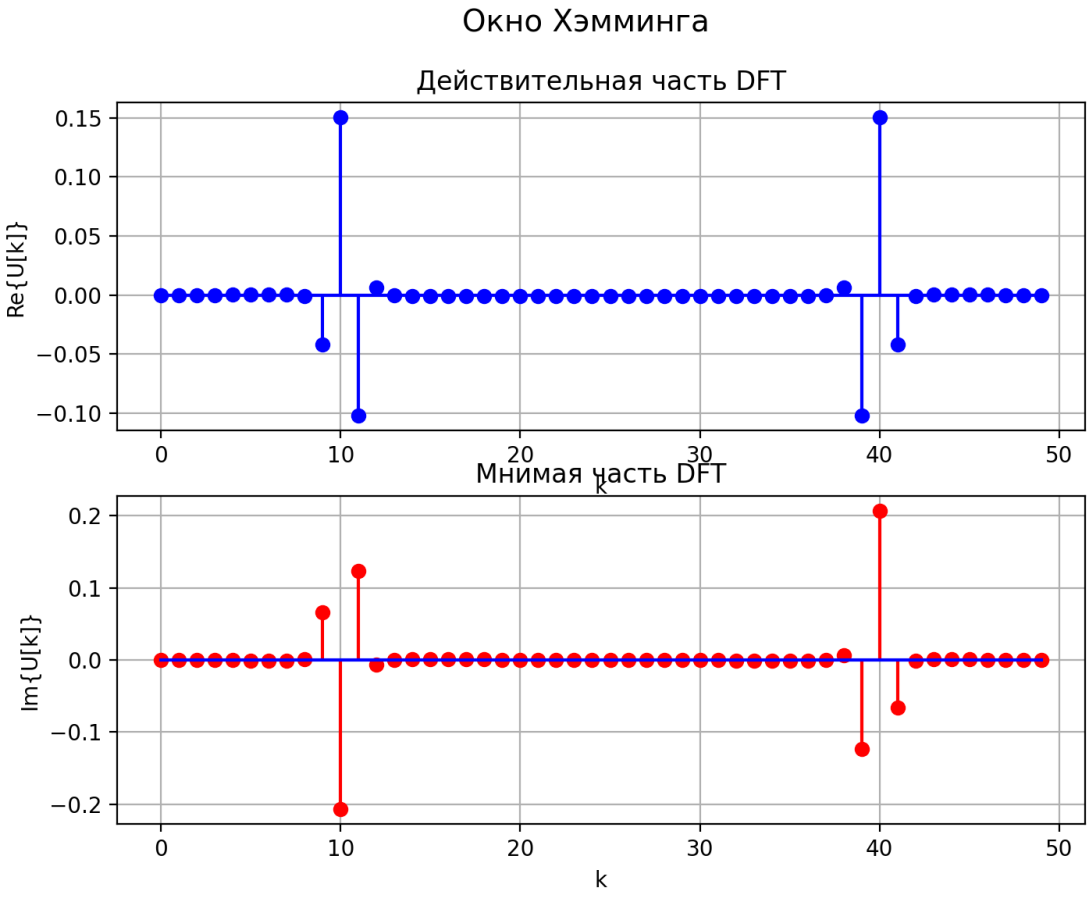

Рисунок 2.4.2 – Результаты ДПФ при использовании окна Хемминга

Сравнение ДПФ при использовании разных вариантов окон:

Прямоугольное окно обеспечивает максимальное частотное разрешение (узкий главный лепесток), но обладает высоким уровнем боковых лепестков и выраженным эффектом Гиббса, что приводит к появлению ложных спектральных компонентов.

Окно Ханна существенно снижает уровень боковых лепестков и подавляет эффект Гиббса за счет косинусного взвешивания. Недостатком является уширение главного лепестка, ухудшающее разделение близких частот.

Окно Хэмминга демонстрирует наилучшее подавление первого бокового лепестка при сохранении приемлемой ширины главного лепестка, обеспечивая оптимальный компромисс между частотной избирательностью и уровнем искажений.

На втором этапе лабораторной работы была разработана программная часть, направленная на получение амплитудного и фазового спектров сигнала на основе результатов, полученных после вычисления прямого дискретного преобразования Фурье (ДПФ). Если на первом этапе исследовались действительные и мнимые составляющие спектра, то на данном этапе выполнялось преобразование этих компонент в более интерпретируемые спектральные характеристики – амплитуду и фазовый сдвиг каждой гармоники.

Основная цель этапа состояла в формировании функции, которая принимает результаты ДПФ в виде действительной и мнимой частей спектра, вычисляет из них амплитудный и фазовый спектры, а затем визуализирует полученные данные.

В результате были получены амплитудный и фазовый спектры для каждого вида окна, графики спектров представлены на рисунках 3.1-3.3:

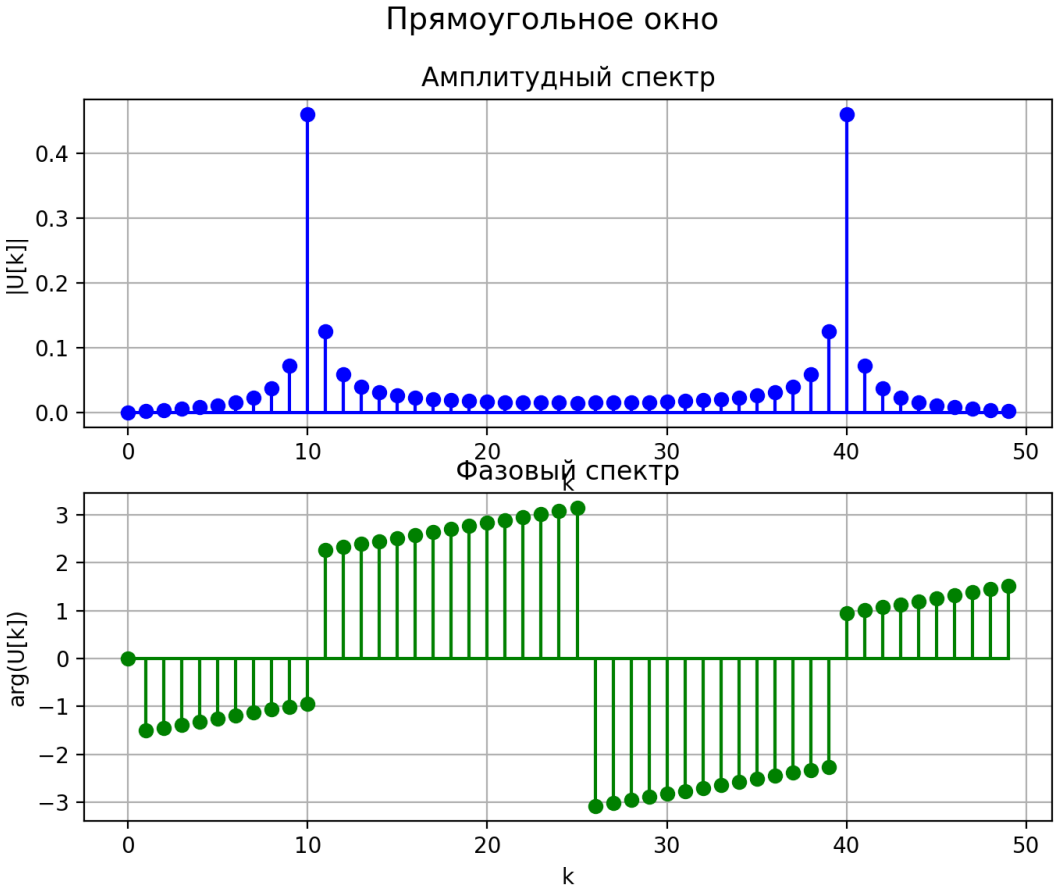

Рисунок 3.1 – Амплитудный и фазовый спектры для прямоугольного окна

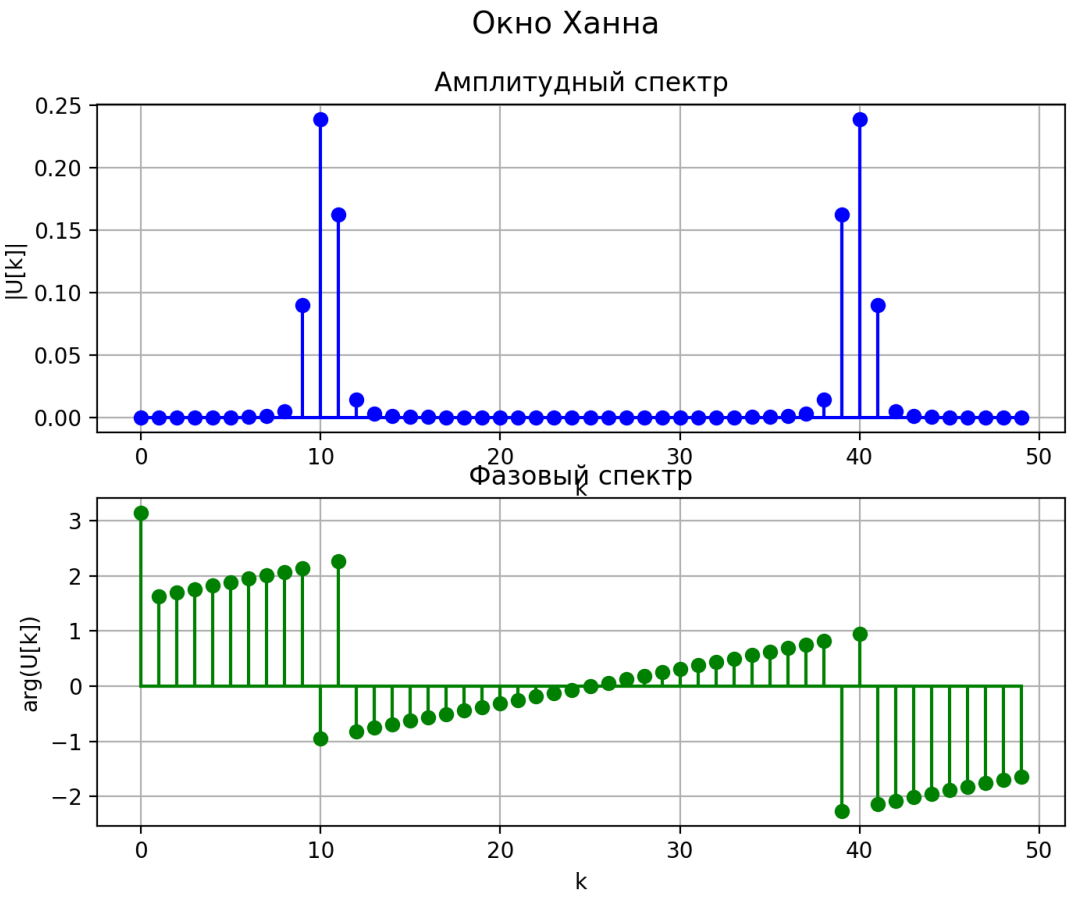

Рисунок 3.2 – Амплитудный и фазовый спектры для окна Ханна

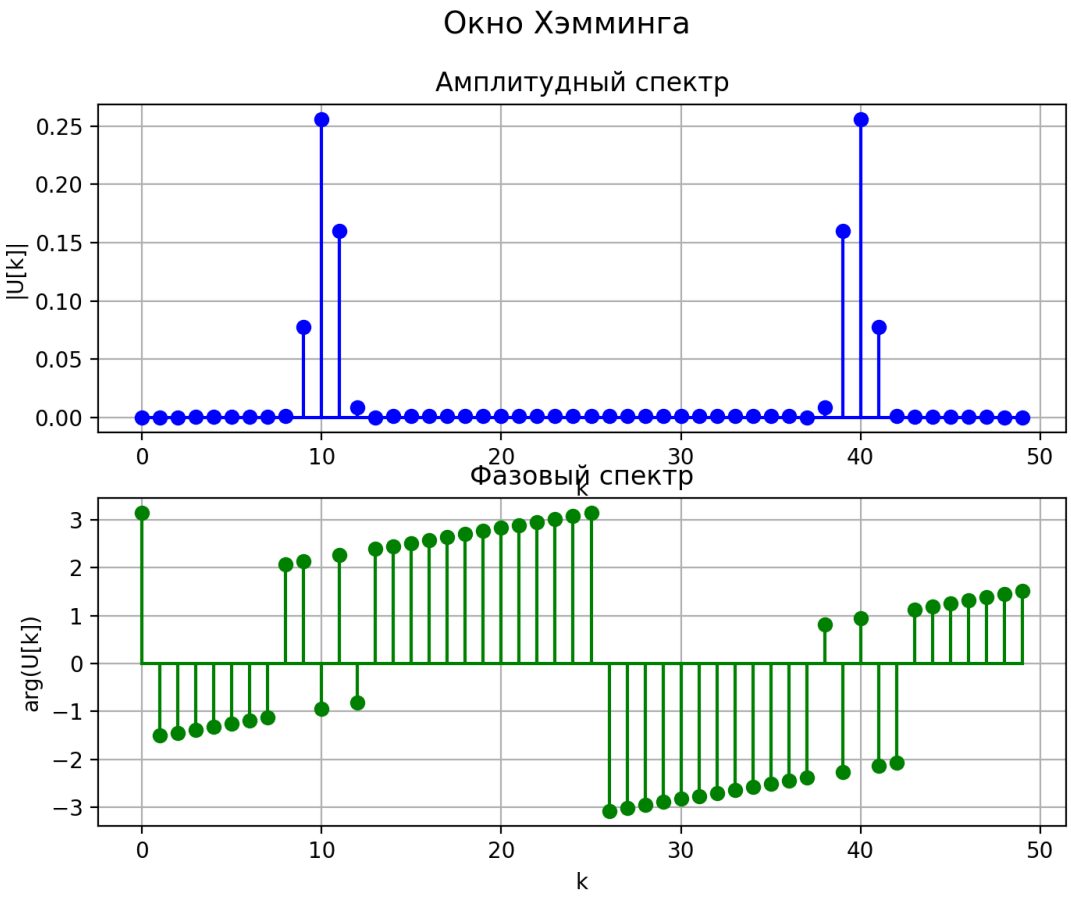

Рисунок 3.3 – Амплитудный и фазовый спектры для окна Хемминга

Разработанная функция является логическим продолжением первого этапа, переводя комплексное спектральное представление в физически интерпретируемые характеристики.

Третий этап лабораторной работы был направлен на исследование и практическую реализацию технологий кодирования и декодирования тональных сигналов стандарта DTMF (Dual-Tone Multi-Frequency), применяемых в телефонных системах набора номера. На данном этапе были разработаны функции, обеспечивающие:

1. генерацию аудиофайла с корректной DTMF-последовательностью;
2. чтение, анализ и декодирование WAV-файла;
3. выделение частотных компонентов каждого тона через прямое ДПФ;
4. определение символов на основе таблицы стандартных DTMF-частот;
5. визуализацию временных форм и спектров для каждого тона.

Работа этапа базируется на свойствах DTMF-сигналов, каждый из которых формируется как сумма двух синусоид фиксированных частот: одна из *low*-группы (697–941 Гц), другая из *high*-группы (1209–1633 Гц). Использовалась таблица приведённая в задании (рисунок 1.1)

Был сгенерирован аудиофайл по последовательности символов: “3566678889776563566678889776556356668999*00*9*6678#89*6687767****99999*****09876”.

Для наглядности процесса декодирования были визуализированы временные формы и спектры для каждого тона. Для примера приведены таковые для первых двух символов на рисунках 4.1.1 – 4.2.2:

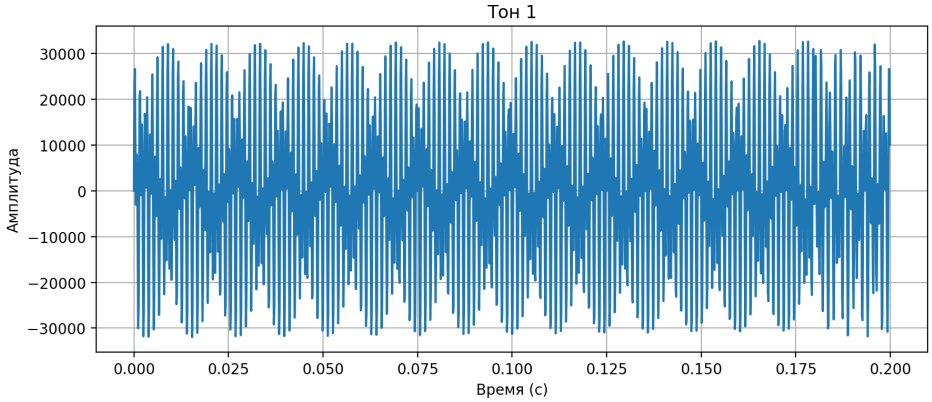

Рисунок 4.1.1 – Графическое представление тона 1

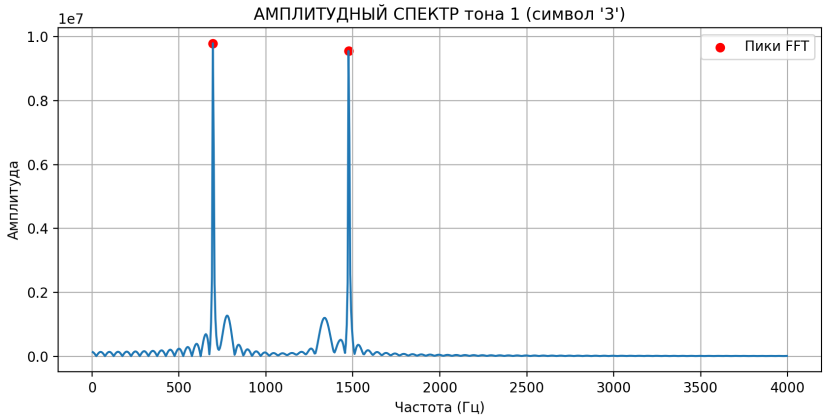

Рисунок 4.1.2 – Амплитудный спектр тона 1

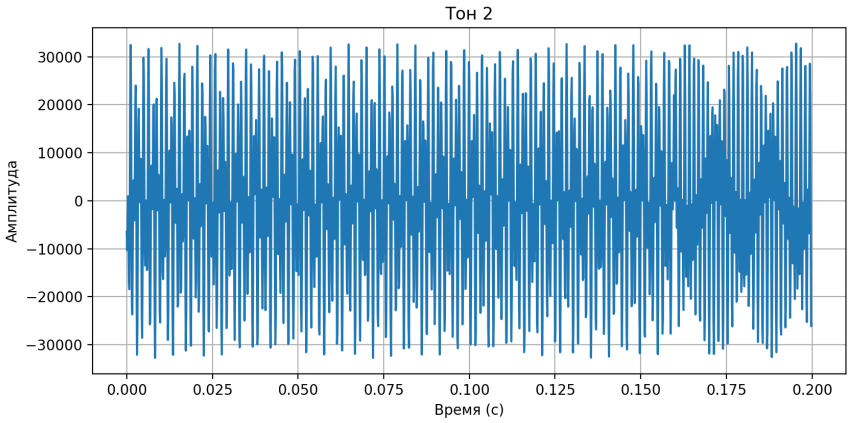

Рисунок 4.2.1 – Графическое представление тона 2

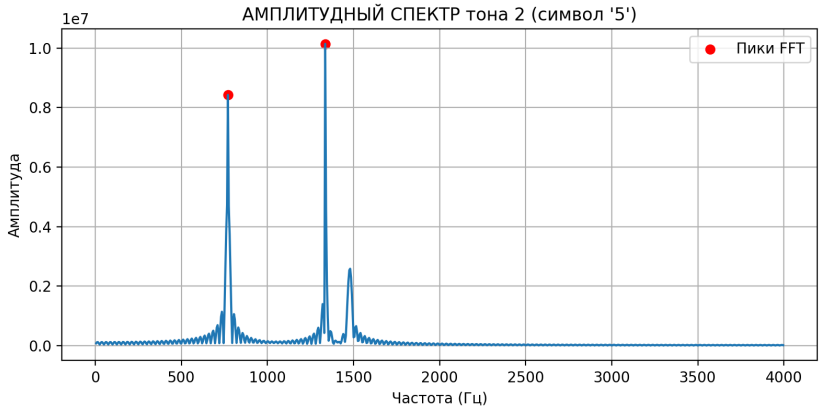

Рисунок 4.2.2 – Амплитудный спектр тона 2

После применения алгоритма декодирования ДПФ была получена исходная последовательность.

В качестве четвёртого этапа выполнения лабораторной работы была разработана функция, осуществляющая обратное ДПФ с целью восстановления исходного сигнала.

Результат восстановления и сравнение с входным сигналом представлен на рисунке 5:

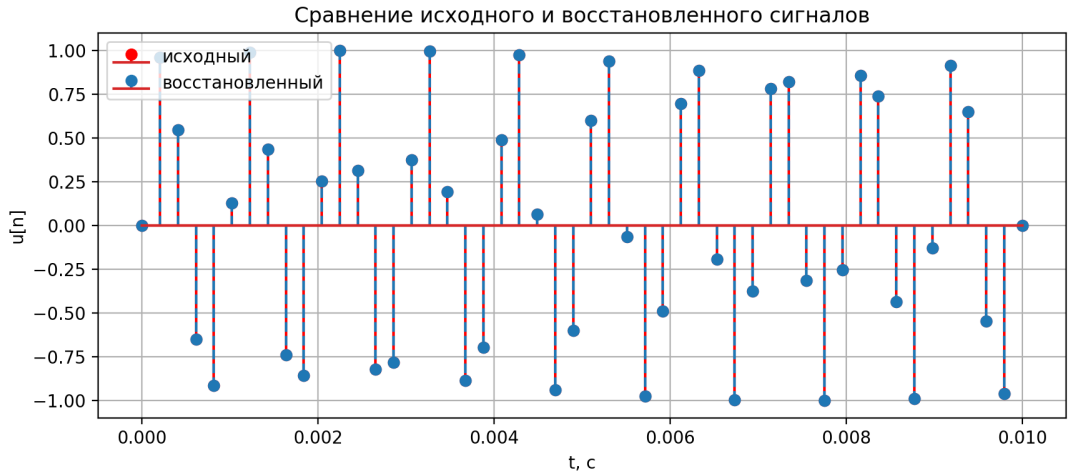

Рисунок 5 – Сравнение исходного и сигнала после обратного ДПФ

5. Вывод:

В ходе лабораторной работы были успешно изучены и практически реализованы ключевые аспекты дискретного преобразования Фурье. Разработанные функции позволяют вычислять прямое и обратное ДПФ, строить спектральные характеристики (действительную, мнимую части, амплитудный и фазовый спектры), а также применять оконные функции для снижения эффекта растекания спектра.

Практическое сравнение оконных функций подтвердило теоретические положения: прямоугольное окно обеспечивает наилучшее разрешение, но вызывает значительное растекание; окна Ханна и Хэмминга эффективно подавляют боковые лепестки, улучшая точность спектрального анализа ценой некоторого ухудшения разрешения.

Реализованный алгоритм декодирования DTMF-сигналов продемонстрировал высокую эффективность и точность, успешно определяя символы по двум доминирующим частотам в спектре.

Проверка обратного ДПФ подтвердила возможность точного восстановления исходного сигнала из его частотного представления с малой погрешностью, что доказывает корректность реализации и соответствие теоретическим основам.

Таким образом, полученные результаты наглядно иллюстрируют основные свойства и возможности дискретного преобразования Фурье в области цифровой обработки сигналов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аграновский А. В. Методические указания к лабораторной работе № 5 «Дискретное преобразование Фурье» по дисциплине «Цифровая обработка и передача сигналов». – Санкт-Петербург: ГУАП, 2025.
2. Библиотека NumPy в Python – URL: https://numpy.org/doc/2.3/user/index.html#user (дата обращения 09.11.2025)
3. Matplotlib Development Team. Matplotlib: Visualization with Python – URL: https://matplotlib.org/stable/index.html (дата обращения: 09.11.2025).
4. Дискретное преобразование Фурье — Википедия – URL: https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B5%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%A4%D1%83%D1%80%D1%8C%D0%B5 (дата обращения: 09.11.2025)

ПРИЛОЖЕНИЕ А

Листинг Программы

```
import numpy as np
import matplotlib.pyplot as plt
import wave
import struct


# Функции DFT и IDFT


def dft(x):
    """
    Прямое Дискретное Преобразование Фурье (DFT) через матричное умножение.
    Возвращает комплексный спектр.
    """
    x = np.asarray(x, dtype=complex)
    N = len(x)
    n = np.arange(N)
    k = n.reshape((N, 1))
    W = np.exp(-2j * np.pi * k * n / N)
    return np.dot(W, x) / N

def idft(X):
    """
    Обратное Дискретное Преобразование Фурье (IDFT) через матричное
умножение.
    """
    X = np.asarray(X, dtype=complex)
    N = len(X)
    k = np.arange(N)
    n = k.reshape((N, 1))
    W = np.exp(2j * np.pi * k * n / N)
    return np.dot(W, X)


# Генерация сигнала

fs = 5000
t = np.linspace(0, 0.01, fs // 100)
u = np.sin(2 * np.pi * 1000 * t)

plt.figure(figsize=(10,4))
plt.title("Исходный сигнал")
plt.stem(t, u, linefmt='b-', markerfmt='bo', basefmt='b', label="дискретный")
plt.xlabel("t, c")
plt.ylabel("Амплитуда")
plt.grid()
plt.legend()
plt.show()


# Формирование окон

N = len(u)
k = np.arange(N)

# Прямоугольное окно
w_rect = np.ones(N)
rect = u * w_rect

# Окно Ханна
```

```
w_hann = 0.5 * (1 - np.cos(2 * np.pi * k / (N - 1)))
hann = u * w_hann

# Окно Хэмминга
w_hamming = 0.53836 - 0.46164 * np.cos(2 * np.pi * k / (N - 1))
hamming = u * w_hamming

windows = {
    "Прямоугольное окно": rect,
    "Окно Ханна": hann,
    "Окно Хэмминга": hamming
}


# Обработка каждого окна

for name, sig in windows.items():
    U = dft(sig)
    Re = np.real(U)
    Im = np.imag(U)
    Amp = np.abs(U)
    Phase = np.angle(U)
    k = np.arange(len(U))

    # Сигнал
    plt.figure(figsize=(10, 4))
    plt.title(f"{name}")
    plt.stem(t, sig, linefmt='b-', markerfmt='bo', basefmt='b',
label="дискретный")
    plt.xlabel("t, c")
    plt.ylabel("u[n]")
    plt.grid()
    plt.legend()
    plt.show()

    # Действительная и мнимая части
    plt.figure(figsize=(8,6))
    plt.suptitle(f"{name}", fontsize=14)

    plt.subplot(2, 1, 1)
    plt.title("Действительная часть DFT")
    plt.stem(k, Re, linefmt='b-', markerfmt='bo', basefmt='b')
    plt.xlabel("k")
    plt.ylabel("Re{U[k]}")
    plt.grid()

    plt.subplot(2, 1, 2)
    plt.title("Мнимая часть DFT")
    plt.stem(k, Im, linefmt='r-', markerfmt='ro', basefmt='b')
    plt.xlabel("k")
    plt.ylabel("Im{U[k]}")
    plt.grid()
    plt.show()

    # Амплитудный и фазовый спектры
    plt.figure(figsize=(8,6))
    plt.suptitle(f"{name}", fontsize=14)

    plt.subplot(2, 1, 1)
    plt.title("Амплитудный спектр")
    plt.stem(k, Amp, linefmt='b-', markerfmt='bo', basefmt='b')
    plt.xlabel("k")
    plt.ylabel("|U[k]|")
```

```
    plt.grid()

    plt.subplot(2, 1, 2)
    plt.title("Фазовый спектр")
    plt.stem(k, Phase, linefmt='g-', markerfmt='go', basefmt='g')
    plt.xlabel("k")
    plt.ylabel("arg(U[k])")
    plt.grid()
    plt.show()


# Восстановление сигнала

X_rect = dft(u)
x_recovered = idft(X_rect)

plt.figure(figsize=(10,4))
plt.title("Сравнение исходного и восстановленного сигналов")
plt.stem(t, u, linefmt='b-', markerfmt='bo', label="исходный")
plt.stem(t, np.real(x_recovered), linefmt='--', label="восстановленный")
plt.xlabel("t, c")
plt.ylabel("u[n]")
plt.legend()
plt.grid()
plt.show()


# Частоты DTMF

low_freqs = [697, 770, 852, 941]
high_freqs = [1209, 1336, 1477, 1633]

dtmf_map = {
    (697, 1209): '1', (697, 1336): '2', (697, 1477): '3', (697, 1633): 'A',
    (770, 1209): '4', (770, 1336): '5', (770, 1477): '6', (770, 1633): 'B',
    (852, 1209): '7', (852, 1336): '8', (852, 1477): '9', (852, 1633): 'C',
    (941, 1209): '*', (941, 1336): '0', (941, 1477): '#', (941, 1633): 'D'
}

# Декодирование DTMF
def decode_dtmf(filename):
    # Чтение WAV файла
    with wave.open(filename, 'rb') as wav:
        fs = wav.getframerate()
        frames = wav.readframes(wav.getnframes())
        samples = np.array(struct.unpack('<' + 'h' * wav.getnframes(),
frames))

    frame_len = int(0.05 * fs)
    decoded = ""
    prev_symbol = None

    for start in range(0, len(samples), frame_len):
        frame = samples[start:start+frame_len]
        if len(frame) < frame_len:
            break
        if np.max(np.abs(frame)) < 500:
            prev_symbol = None
            continue

        X = np.abs(dft(frame))
        freqs = np.fft.fftfreq(len(frame), 1/fs)
```

```
        # Низкий диапазон
        low_mask = (freqs >= 600) & (freqs <= 1000)
        low_X = X[low_mask]
        low_freqs_axis = freqs[low_mask]
        lf_detected = low_freqs_axis[np.argmax(low_X)]

        # Высокий диапазон
        high_mask = (freqs >= 1200) & (freqs <= 1700)
        high_X = X[high_mask]
        high_freqs_axis = freqs[high_mask]
        hf_detected = high_freqs_axis[np.argmax(high_X)]

        # Ближайшие стандартные частоты
        lf = min(low_freqs, key=lambda f: abs(f-lf_detected))
        hf = min(high_freqs, key=lambda f: abs(f-hf_detected))
        symbol = dtmf_map.get((lf,hf), None)

        if symbol is not None and symbol != prev_symbol:
            decoded += symbol
            # Визуализация сигнала
            t_frame = np.arange(len(frame))/fs
            plt.figure(figsize=(6,3))
            plt.plot(t_frame, frame)
            plt.title(f"Сигнал для символа '{symbol}'")
            plt.xlabel("Время, с")
            plt.ylabel("Амплитуда")
            plt.grid()
            plt.show()

        prev_symbol = symbol

    return decoded


# Пример использования DTMF

decoded_msg = decode_dtmf("dtmf_test.wav")
print("Расшифрованное сообщение:", decoded_msg)
```