

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет информационных технологий и управления  
Кафедра интеллектуальных информационных технологий  
Дисциплина «Обработка изображений в интеллектуальных системах»

**ОТЧЁТ**  
к лабораторной работе №1  
на тему  
**«БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ»**

БГУИР 6-05-0611-03 130

Выполнил студент группы 321701  
СЕМЕНЯКО Владимир Дмитриевич

---

(дата, подпись студента)

Проверил  
САЛЬНИКОВ Даниил Андреевич

---

(дата, подпись преподавателя)

Минск 2025

# 1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

- а) Изучить алгоритм быстрого преобразования Фурье (БПФ).
- б) Выполнить программную реализацию алгоритма БПФ.
- в) Подать на вход программы функцию  $\sin(x)$ , продемонстрировать правильность работы преобразования путем визуализации исходного сигнала и его спектра.

## 2 ВЫПОЛНЕНИЕ РАБОТЫ

Программа была реализована на языке Python с использованием `cmath`, `math` и `matplotlib.pyplot` для визуализации результатов.

### Листинг 1 – Код программы

```
import cmath
import math
import matplotlib.pyplot as plt

def fft(x: list[float]) -> list[complex]:
    N = len(x)
    if N <= 1:
        return [complex(val) for val in x]

    even = fft(x[0::2])
    odd = fft(x[1::2])

    result = [complex(0)] * N
    for k in range(N // 2):
        angle = -2j * cmath.pi * k / N
        t = cmath.exp(angle) * odd[k]

        result[k] = even[k] + t
        result[k + N // 2] = even[k] - t

    return result

def perform_fft_analysis(
    signal_frequency: float = 5.0,
    sampling_rate: int = 128,
    duration: int = 2
):
    num_samples = int(sampling_rate * duration)
    time_vector = [i / sampling_rate for i in range(num_samples)]

    signal = [
        math.sin(2 * math.pi * signal_frequency * t) for t in time_vector
    ]
```

```

fft_result = fft(signal)

fft_amplitude = [abs(c) / num_samples for c in fft_result]

frequencies = [
    i * sampling_rate / num_samples for i in range(num_samples)
]

plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)
plt.plot(time_vector, signal)
plt.titleИсходный(" сигнал: sin(x)")
plt.xlabelВремя(" c()")
plt.ylabelАмплитуда("")
plt.grid(True)

plt.subplot(2, 1, 2)
half_N = num_samples // 2
plt.stem(frequencies[:half_N], fft_amplitude[:half_N])
plt.titleАмплитудный(" спектр сигнала самописный( БПФ)")
plt.xlabelЧастота(" Гц()")
plt.ylabelАмплитуда("")
plt.grid(True)

plt.tight_layout()
plt.show()

```

В ходе выполнения работы на вход программы была подана функция  $\sin(x)$  с частотой 5 Гц. На Рисунке 1 представлен исходный сигнал во временной области.

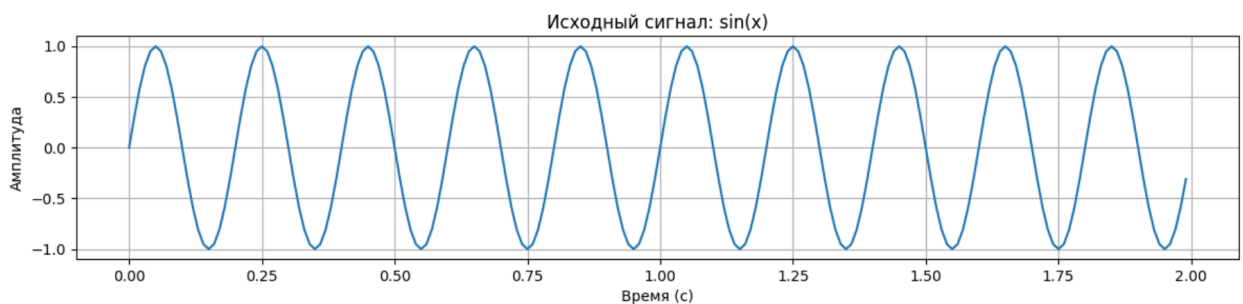


Рисунок 1 – Результат выполнения программы

После применения алгоритма БПФ был получен амплитудный спектр сигнала, показанный на Рисунке 2.

На графике спектра виден единственный ярко выраженный пик на частоте 5 Гц. Это в точности соответствует частоте исходного синусоидального сигнала, что подтверждает корректность работы реализованного преобразо-

вания. Остальные частоты имеют нулевую или близкую к нулю амплитуду, как и ожидалось для чистого синусоидального сигнала.

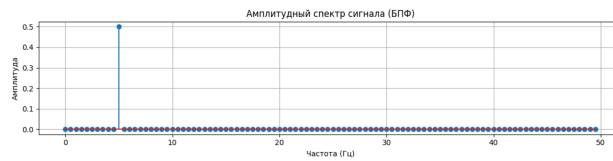


Рисунок 2 – Формула для определения ключевого пространства шифра

## ВЫВОД

В рамках данной лабораторной работы был изучен и программно реализован алгоритм быстрого преобразования Фурье. Программа успешно сгенерировала тестовый сигнал на основе функции  $\sin(x)$  и выполнила его преобразование в частотную область. Результаты анализа спектра наглядно продемонстрировали правильность работы алгоритма, так как пик на графике спектра точно совпал с исходной частотой сигнала. Это подтверждает, что БПФ является мощным инструментом для анализа частотных характеристик сигналов.