



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного автономного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ДОМАШНЯЯ РАБОТА 1

«РАЗЛОЖЕНИЕ СИГНАЛОВ»

ДИСЦИПЛИНА: «Цифровая обработка сигналов»

Выполнил: студент гр. ИУК4-72Б _____ (____ Губин Е.В.____)
(Подпись) (Ф.И.О.)

Проверил: _____ (____ Чурилин О.И.____)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

Цель: формирование практических навыков разложения сигналов с использованием дискретного преобразования Фурье (ДПФ).

Задачи:

- 1) с помощью ДПФ построить АЧХ гармонического сигнала;
- 2) из спектра сигнала определить частоты основных гармоник сигнала и осуществить фильтрацию этих гармоник с помощью фильтров любого типа, подобрав соответствующие параметры фильтров;
- 3) в спектральной плоскости отобразить составляющую сигнала;
- 4) над каждой выделенной составляющей сигнала произвести обратное ДПФ;
- 5) построить графики полученных сигналов.

Вариант №3

Ход выполнения лабораторной работы:

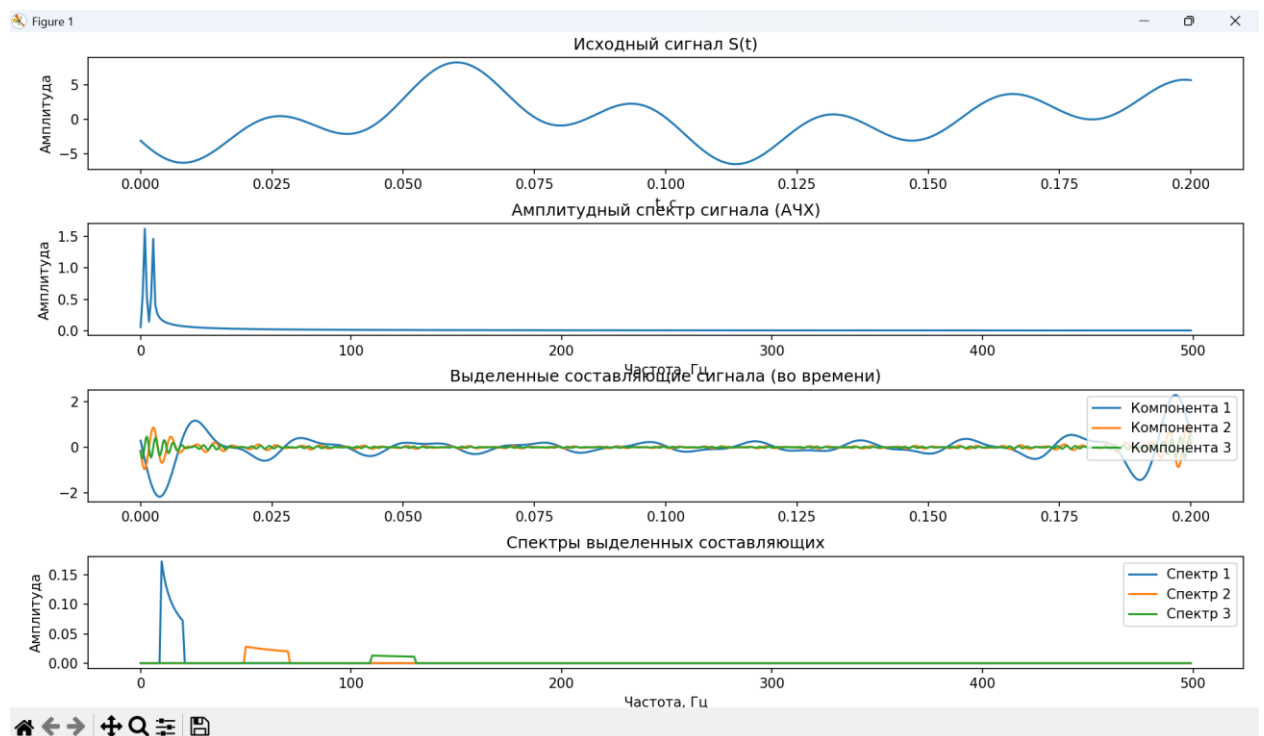


Рисунок 1 Результат выполнения

Листинг программы:

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

A1, O1, F1 = 1.5, 60, np.deg2rad(-80)
A2, O2, F2 = 3.0, 15, np.deg2rad(30)
A3, O3, F3 = 0.7, 90, np.deg2rad(120)
A4, O4, F4 = 4.0, 120, np.deg2rad(0)

fs = 1000
t = np.linspace(0, 0.2, fs)\
```

```

S1 = A1 * np.sin(O1 * t + F1)
S2 = A2 * np.sin(O2 * t + F2)
S3 = A3 * np.sin(O3 * t + F3)
S4 = A4 * np.sin(O4 * t + F4)

S = S1 * (S2 + S3 + S4)

N = len(S)
S_fft = np.fft.fft(S)
freq = np.fft.fftfreq(N, d=1/fs)
amp = np.abs(S_fft) / N

def bandpass_filter(fft_data, freq, f_low, f_high):
    filtered = np.copy(fft_data)
    filtered[(np.abs(freq) < f_low) | (np.abs(freq) > f_high)] = 0
    return filtered

bands = [(10, 20), (50, 70), (110, 130)]
filtered_components = []

for f_low, f_high in bands:
    filtered_fft = bandpass_filter(S_fft, freq, f_low, f_high)
    component = np.fft.ifft(filtered_fft).real
    filtered_components.append(component)

plt.figure(figsize=(12, 10))

plt.subplot(4, 1, 1)
plt.plot(t, S)
plt.title("Исходный сигнал S(t)")
plt.xlabel("t, с")
plt.ylabel("Амплитуда")

plt.subplot(4, 1, 2)
plt.plot(freq[:N//2], amp[:N//2])
plt.title("Амплитудный спектр сигнала (АЧХ)")
plt.xlabel("Частота, Гц")
plt.ylabel("Амплитуда")

for i, comp in enumerate(filtered_components):
    plt.subplot(4, 1, 3)
    plt.plot(t, comp, label=f'Компонента {i+1}')
plt.title("Выделенные составляющие сигнала (во времени)")
plt.legend()

plt.subplot(4, 1, 4)
for i, comp in enumerate(filtered_components):
    comp_fft = np.abs(np.fft.fft(comp)) / N
    plt.plot(freq[:N//2], comp_fft[:N//2], label=f'Спектр {i+1}')
plt.title("Спектры выделенных составляющих")
plt.xlabel("Частота, Гц")
plt.ylabel("Амплитуда")
plt.legend()

plt.tight_layout()
plt.show()

```

Вывод: в ходе выполнения домашней работы было произведено разложение сигнала на спектры и ДПФ.