



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»

ЛАБОРАТОРНАЯ РАБОТА 5

ДИСЦИПЛИНА: «Цифровая обработка сигналов»

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

Проверил: _____ (____Чурилин О.И____)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

Целью выполнения лабораторной работы является формирование практических навыков анализа спектра дискретных сигналов с помощью дискретного преобразования Фурье (ДПФ).

Основными **задачами** выполнения лабораторной работы являются:

1. используя ДПФ построить АЧХ сигналов: заданного и отфильтрованного;
2. с помощью АЧХ проверить правильность процедуры фильтрации, при необходимости скорректировать параметры фильтра.

Вариант 17

№	Значения частот			Вид фильтра и составляющие сигнала, подлежащие фильтрации для двух видов сигналов (верхняя строка для сигнала $S_1 + S_2$, нижняя строка для $S_1 + S_2 + S_3$)			
	S_1	S_2	S_3	Баттерворта	Чебышева 1 рода	Чебышева 2 рода	Эллиптический
17	50	70	90	ФНЧ, S_1	ПФ, S_1	ФВЧ, S_2	РФ, S_1

Листинг программы:

```
# %%
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks, butter, filtfilt

fs = 1000
T = 1
t = np.linspace(0, T, fs*T, endpoint=False)
dt = 1/fs

S1 = np.sin(2*np.pi*50*t)
S2 = 0.7*np.sin(2*np.pi*70*t)
S3 = 0.5*np.sin(2*np.pi*90*t)
S123 = S1 + S2 + S3

N = len(S123)
S_fft = np.fft.fft(S123)
freqs = np.fft.fftfreq(N, d=dt)

plt.figure(figsize=(10, 5))
plt.plot(freqs[:N//2], np.abs(S_fft[:N//2]))
plt.title("Амплитудный спектр сигнала (АЧХ)")
plt.xlabel("Частота [Гц]")
plt.ylabel("Амплитуда")
plt.grid(True)
plt.show()
```

```

peaks, props = find_peaks(np.abs(S_fft[:N//2]),
height=0.1*np.max(np.abs(S_fft)))
harmonics_freqs = freqs[peaks]

print("Основные гармоники сигнала (Гц):", harmonics_freqs)

plt.figure(figsize=(12, 8))

for i, k in enumerate(peaks[:3]):
    S_fft_h = np.zeros_like(S_fft)
    S_fft_h[k] = S_fft[k]
    S_fft_h[-k] = S_fft[-k]

    S_harm = np.fft.ifft(S_fft_h).real

    plt.subplot(3,1,i+1)
    plt.plot(t, S_harm)
    plt.title(f"Гармоника {i+1}, частота = {freqs[k]:.1f} Гц")
    plt.xlabel("Время [с]")
    plt.ylabel("Амплитуда")
    plt.grid(True)

plt.tight_layout()
plt.show()

cutoff = 60
b, a = butter(4, cutoff/(fs/2), btype='low')
S123_filt = filtfilt(b, a, S123)

plt.figure(figsize=(10, 5))
plt.plot(t, S123, label="Исходный сигнал S123")
plt.plot(t, S123_filt, label="Отфильтрованный (ФНЧ, Баттерворт)")
plt.title("Фильтрация сигнала")
plt.xlabel("Время [с]")
plt.ylabel("Амплитуда")
plt.legend()
plt.grid(True)
plt.show()

# %% [markdown]
# # Сравнение фильтрованного и нефильтрованного сигнала

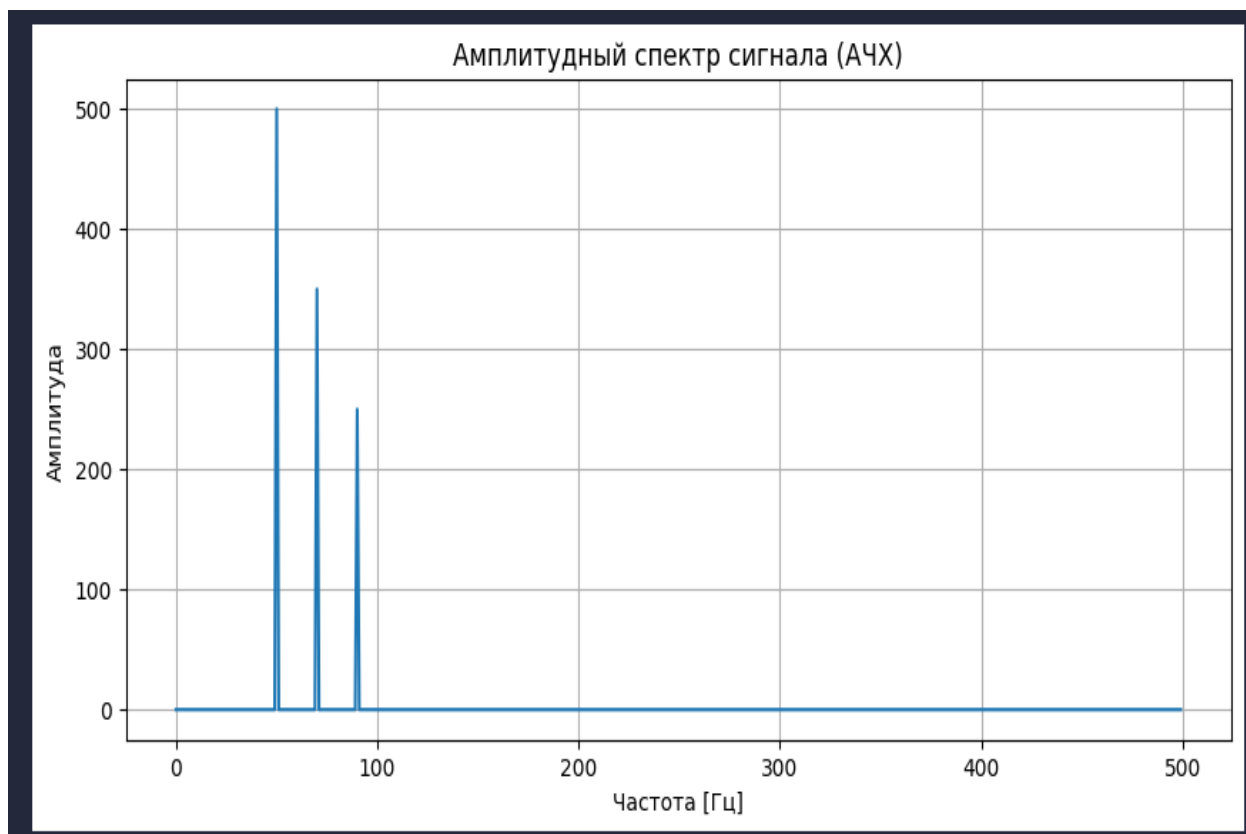
# %%
S_fft_filt = np.fft.fft(S123_filt)

plt.figure(figsize=(10, 5))
plt.plot(freqs[:N//2], np.abs(S_fft[:N//2]), label="Исходный спектр")
plt.plot(freqs[:N//2], np.abs(S_fft_filt[:N//2]), label="После фильтрации")
plt.title("Спектр исходного и отфильтрованного сигнала")
plt.xlabel("Частота [Гц]")
plt.ylabel("Амплитуда")
plt.legend()
plt.grid(True)
plt.show()

```

Результаты выполнения программы:

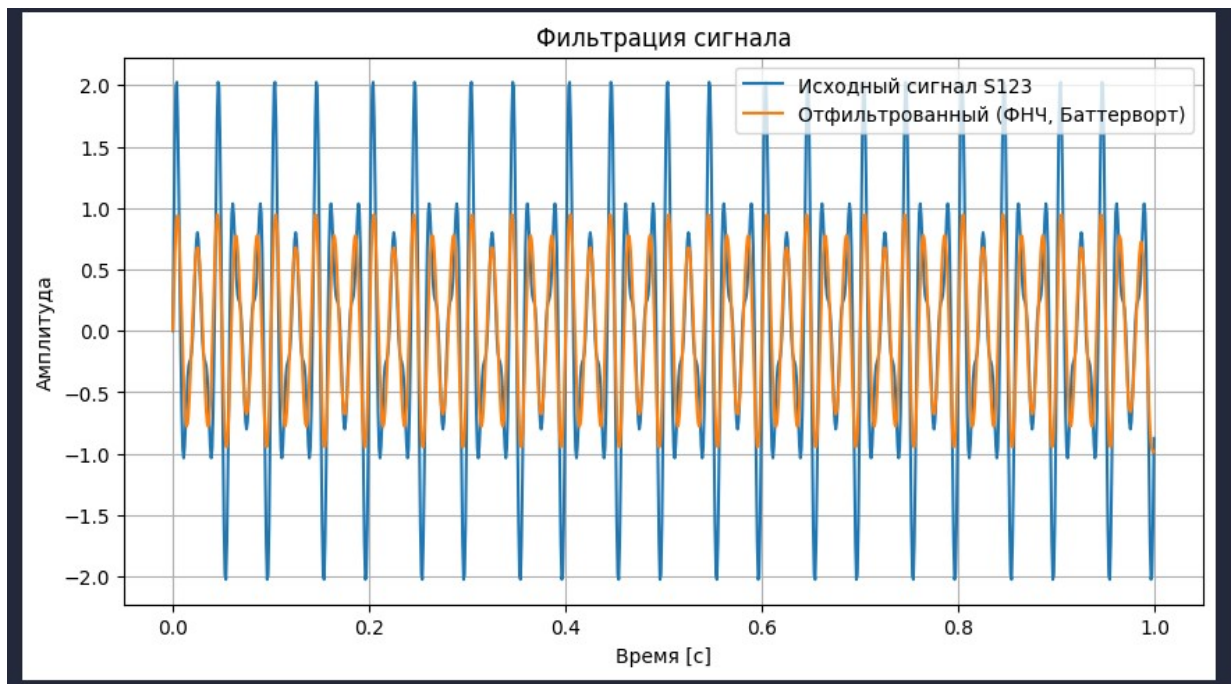
Спектр исходного сигнала $S1+S2+S3$



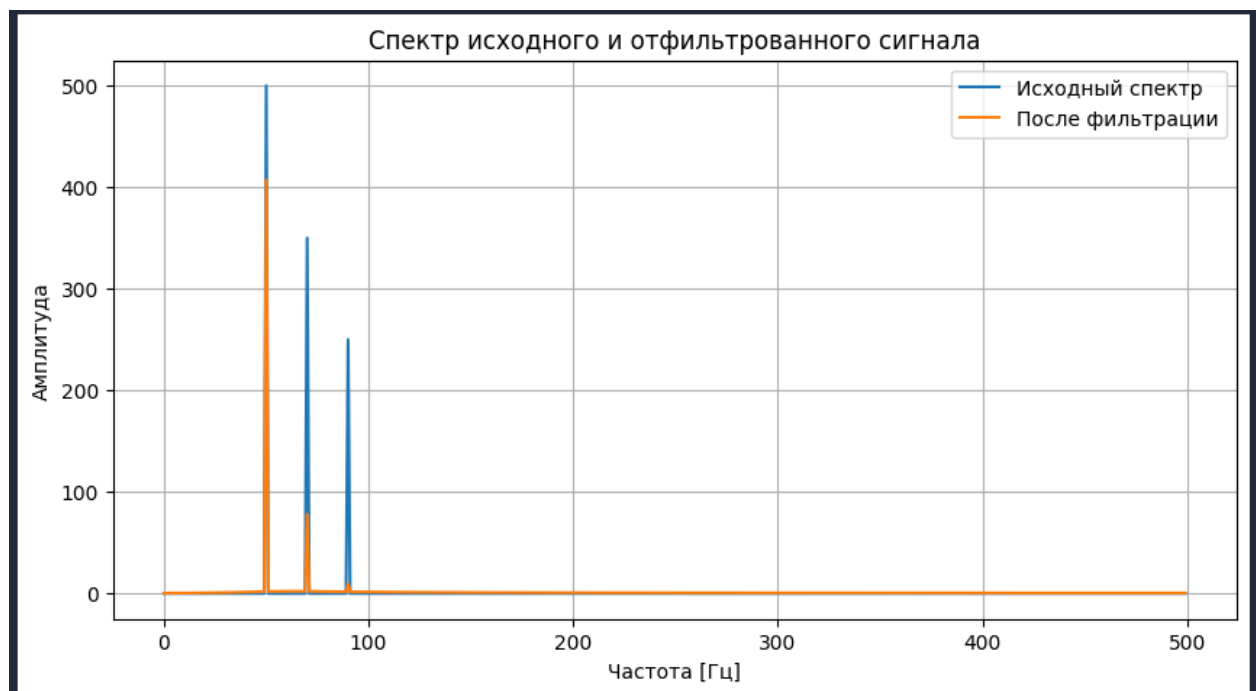
Исходные гармоники:



Сравнение исходного и отфильтрованного сигнала



Сравнение спектров сигналов до и после фильтрации фильтром butter:



Вывод: в ходе лабораторной работы я получил навыки построения аналоговых фильтров, а также базовые навыки спектрального анализа сигналов.