



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 4

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 выполнения фильтрации синусоидальных сигналов с различными значениями параметров.

Задачи:

- 1. задать параметры синусоидальных сигнала;
- 2. выполнить фильтрацию трех синусоидальных сигналов с разными частотами, используя четыре вида фильтров (Баттерворта, Чебышева 1 рода, Чебышева 2 рода, эллиптического).

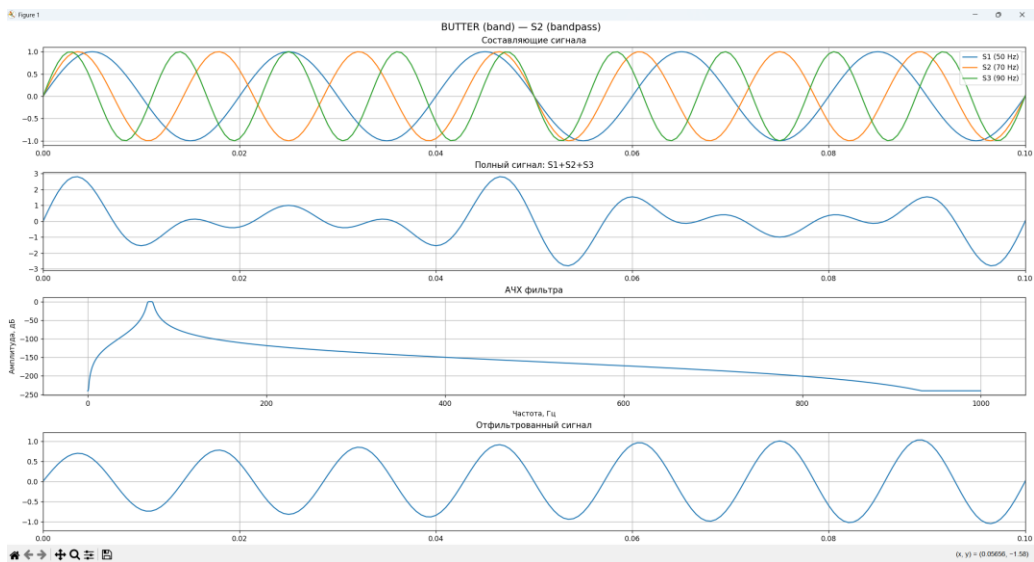
Формулировка задания (3 вариант):

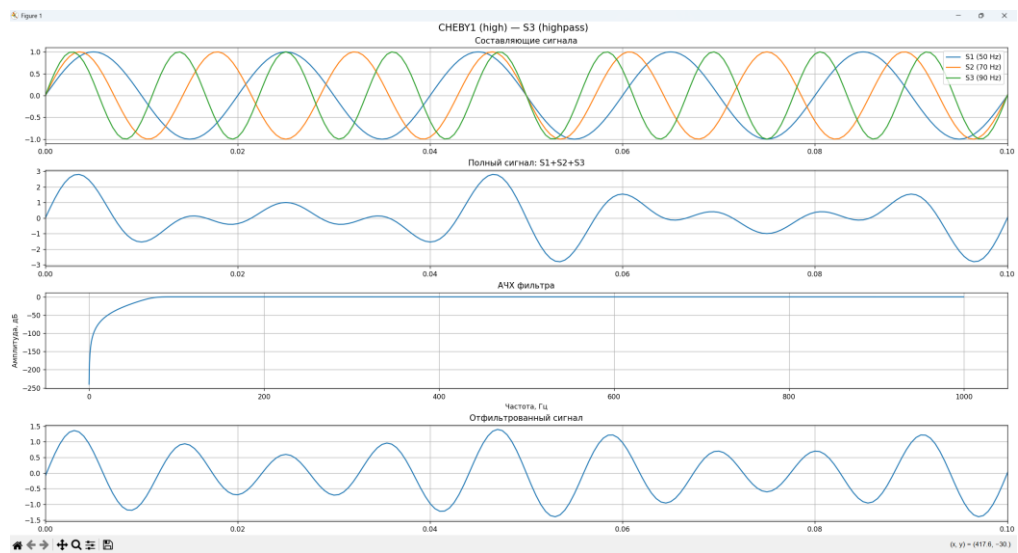
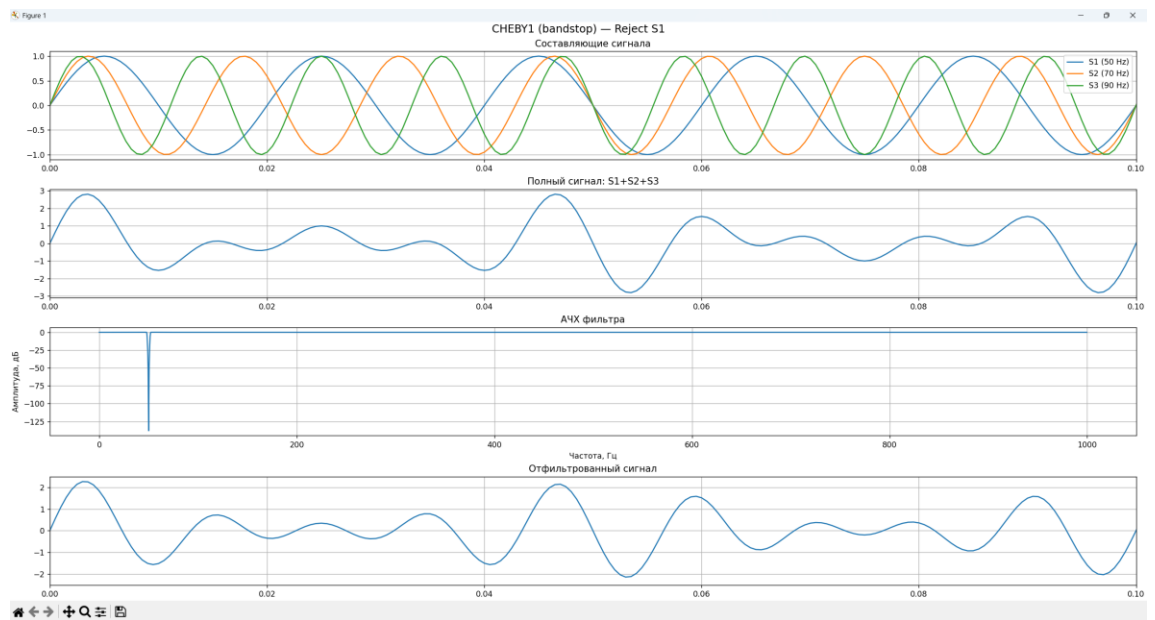
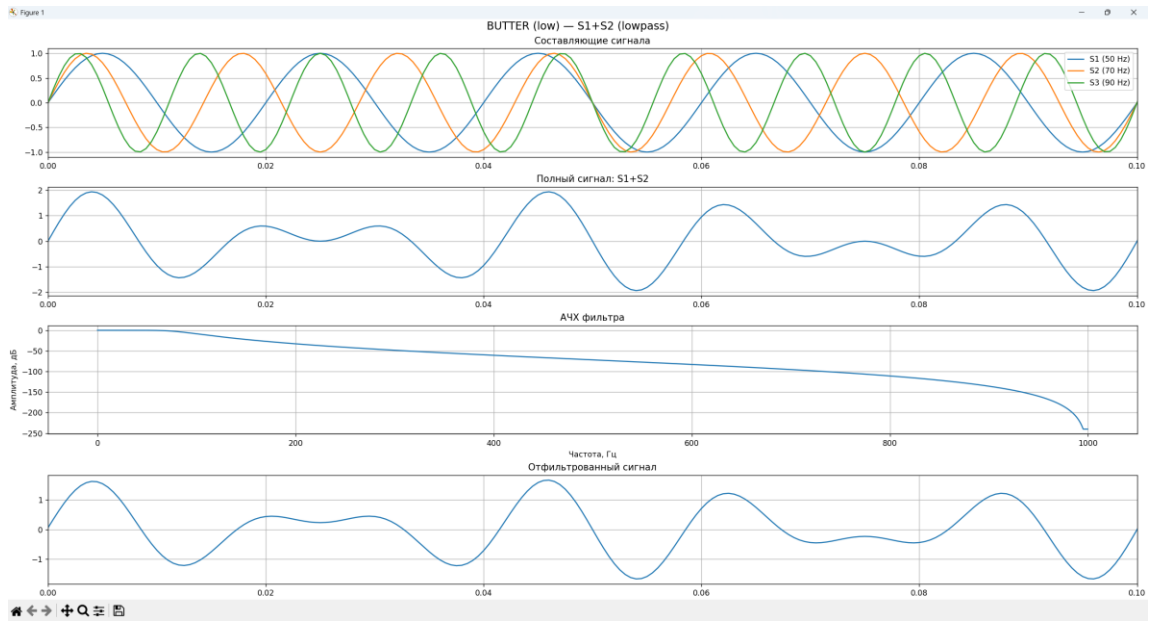
Для параметров фильтров следует принимать приблизительно следующие значения:

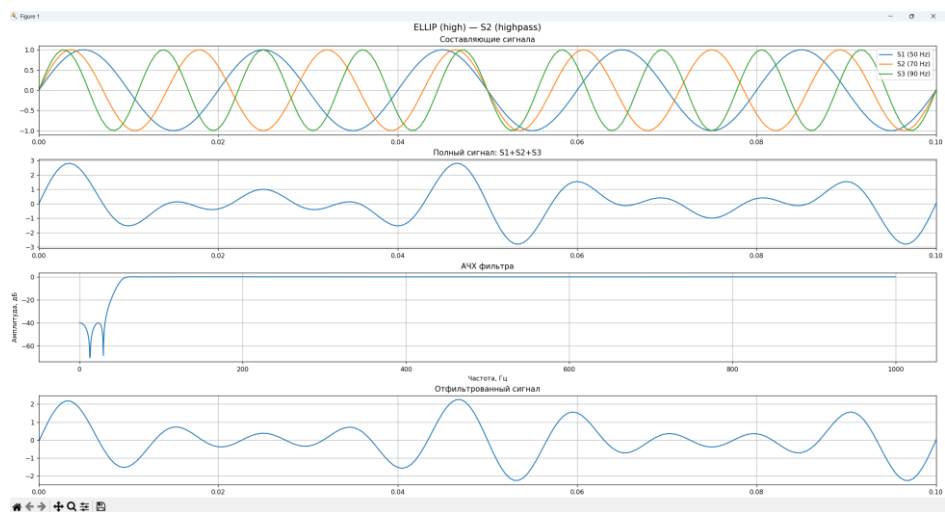
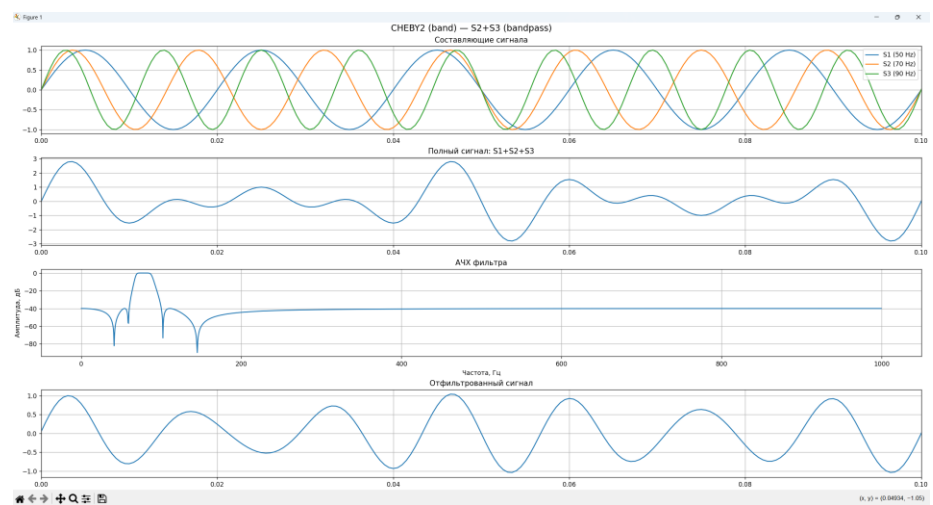
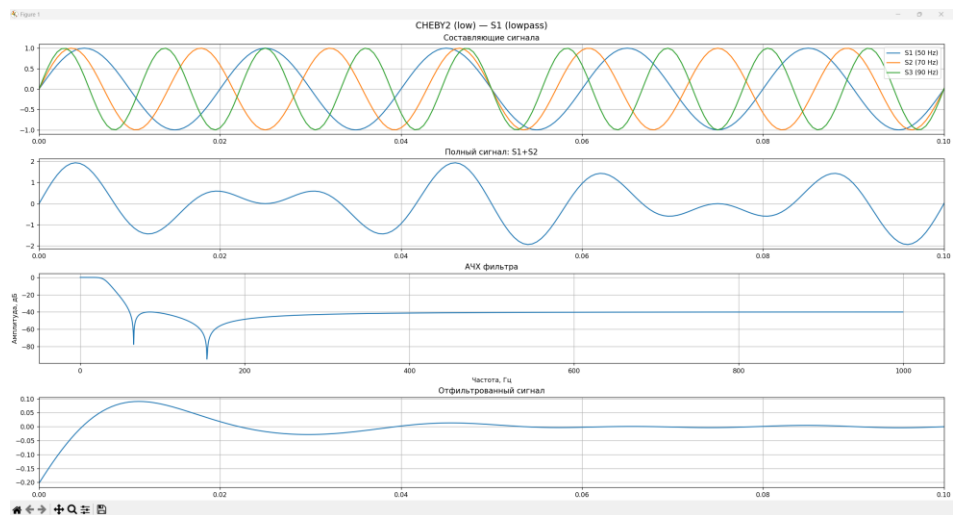
- 1. n (порядок фильтра) ≥ 4 ;
- 2. R_p (уровень пульсаций в полосе пропускания) $\leq 0,1$;
- 3. R_s (уровень пульсаций в полосе задерживания) ≥ 40 .

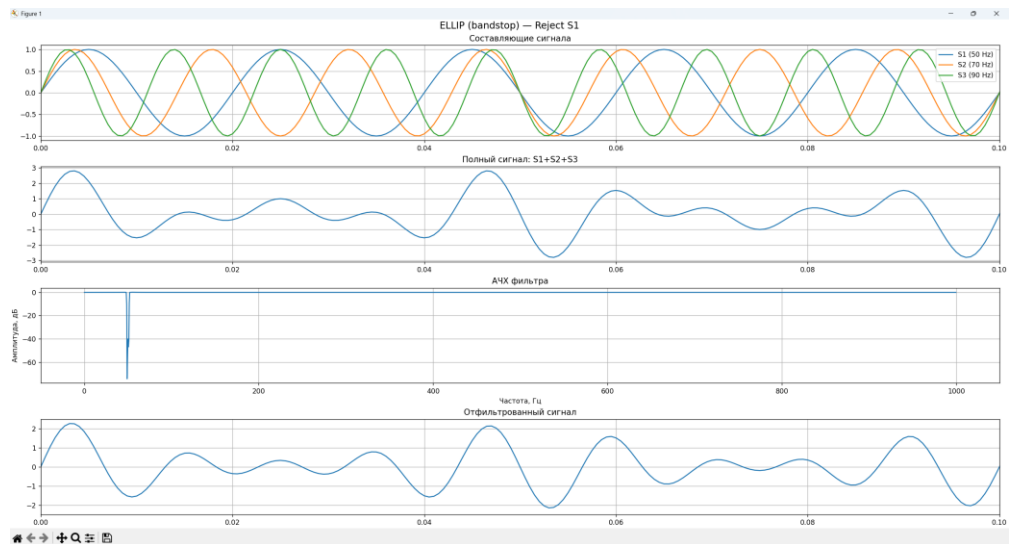
№	Значения частот			Вид фильтра и составляющие сигнала, подлежащие фильтрации для двух видов сигналов (верхняя строка для сигнала $S_1 + S_2$, нижняя строка для $S_1 + S_2 + S_3$)			
	S_1	S_2	S_3	Баттерворт а	Чебышева 1 рода	Чебышева 2 рода	Эллиптический
3	50	70	90	ПФ, S_2	РФ, S_1	ФНЧ, S_1	ФВЧ, S_2
				ФНЧ, $S_1 + S_2$	ФВЧ, S_3	ПФ, $S_2 + S_3$	РФ, S_1

Результаты выполнения:









Листинг программы:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

fs = 2000
T = 1.0
t = np.linspace(0, T, int(fs*T), endpoint=False)

f1, f2, f3 = 50.0, 70.0, 90.0

s1 = np.sin(2*np.pi*f1*t)
s2 = np.sin(2*np.pi*f2*t)
s3 = np.sin(2*np.pi*f3*t)
sig_s12 = s1 + s2
sig_s123 = s1 + s2 + s3

n = 4
rp = 0.1
rs = 40.0
nyq = fs / 2.0

def design_filter(filt, fcut, btype):
    if isinstance(fcut, (list, tuple, np.ndarray)):
        Wn = [f/nyq for f in fcut]
    else:
        Wn = fcut/nyq
    if filt == 'butter':
        sos = signal.butter(n, Wn, btype=btype, output='sos')
    elif filt == 'cheby1':
        sos = signal.cheby1(n, rp, Wn, btype=btype, output='sos')
    elif filt == 'cheby2':
        sos = signal.cheby2(n, rs, Wn, btype=btype, output='sos')
    elif filt == 'ellip':
        sos = signal.ellip(n, rp, rs, Wn, btype=btype, output='sos')
    else:
        raise ValueError('Unknown filter type')
    return sos

def plot_response(sos, ax):
    w, h = signal.sosfreqz(sos, worN=2000, fs=fs)
    ax.plot(w, 20*np.log10(np.maximum(np.abs(h), 1e-12)))
    ax.set_title('АЧХ фильтра')
```

```

ax.set_xlabel('Частота, Гц')
ax.set_ylabel('Амплитуда, дБ')
ax.grid(True)

def apply_and_plot(filt, input_signal, input_label, fcut, btype,
title_label):
    sos = design_filter(filt, fcut, btype)
    y = signal.sosfiltfilt(sos, input_signal)

    fig, axs = plt.subplots(4, 1, figsize=(10, 10), constrained_layout=True)

    axs[0].plot(t, s1, label='S1 (50 Hz)')
    axs[0].plot(t, s2, label='S2 (70 Hz)')
    axs[0].plot(t, s3, label='S3 (90 Hz)')
    axs[0].set_xlim(0, 0.1)
    axs[0].set_title('Составляющие сигнала')
    axs[0].legend()
    axs[0].grid(True)

    axs[1].plot(t, input_signal)
    axs[1].set_xlim(0, 0.1)
    axs[1].set_title(f'Полный сигнал: {input_label}')
    axs[1].grid(True)

    plot_response(sos, axs[2])

    axs[3].plot(t, y)
    axs[3].set_xlim(0, 0.1)
    axs[3].set_title('Отфильтрованный сигнал')
    axs[3].grid(True)

    fig.suptitle(f'{filt.upper()} ({btype}) - {title_label}', fontsize=14)
    plt.show()

bw = 6.0

bp_s2 = (f2 - bw/2, f2 + bw/2)
bp_s3 = (f3 - bw/2, f3 + bw/2)
bp_s2s3 = (f2 - 10.0, f3 + 10.0)

lp_s12_cut = (f2 + f3) / 2.0
lp_s1_cut = (f1 + f2) / 2.0
hp_cut_60 = (f1 + f2) / 2.0
hp_cut_80 = (f2 + f3) / 2.0

tasks = [
    ('butter', sig_s123, 'S1+S2+S3', bp_s2, 'band', 'S2 (bandpass)'),
    ('butter', sig_s12, 'S1+S2', lp_s12_cut, 'low', 'S1+S2 (lowpass)'),
    ('cheby1', sig_s123, 'S1+S2+S3', (f1-2, f1+2), 'bandstop', 'Reject S1'),
    ('cheby1', sig_s123, 'S1+S2+S3', bp_s3[0], 'high', 'S3 (highpass)'),
    ('cheby2', sig_s12, 'S1+S2', lp_s1_cut, 'low', 'S1 (lowpass)'),
    ('cheby2', sig_s123, 'S1+S2+S3', bp_s2s3, 'band', 'S2+S3 (bandpass)'),
    ('ellip', sig_s123, 'S1+S2+S3', hp_cut_60, 'high', 'S2 (highpass)'),
    ('ellip', sig_s123, 'S1+S2+S3', (f1-2, f1+2), 'bandstop', 'Reject S1'),
]

for filt, sig_in, sig_label, fcut, btype, title in tasks:
    apply_and_plot(filt, sig_in, sig_label, fcut, btype, title)

print("Готово. Все фильтры построены.")

```

Вывод: в ходе выполнения лабораторной работы были получены навыки по фильтрации синусоидальных сигналов.