|  |  |
| --- | --- |
|  | **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 технологи»

**ЛАБОРАТОРНАЯ РАБОТА**

**«Фильтрация синусоидальных сигналов»**

**по дисциплине: «Цифровая обработка сигналов»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил: студент группы ИУК4-72Б | |  |  | И. А. Петроченков | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Проверил(-а): | |  |  | О. И. Чурилин | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | | | |
|  | - Балльная оценка:  - Оценка: | | | |

Калуга, 2025

**Целью** выполнения лабораторной работы является формирование практических навыков выполнения фильтрации синусоидальных сигналов с различными значениями параметров.

**Основными задачами** выполнения лабораторной работы являются:

1. Задать параметры синусоидальных сигналов;
2. Выполнить фильтрацию трех синусоидальных сигналов с разными частотами, используя четыре вида фильтров (Баттерворта, Чебышева 1 рода, Чебышева 2 рода, эллиптического).

**Вариант задания:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Значения частот** | | | **Вид фильтра и составляющие сигнала, подлежащие фильтрации для двух видов сигнала (верхняя строка для сигнала , нижняя строка для )** | | | |
|  |  |  | **Баттерворта** | **Чебышева 1 рода** | **Чебышева 2 рода** | **Эллиптический** |
| 1 | 25 | 40 | 60 | ФВЧ, | РФ, | ФНЧ, | ПФ, |
|  |  |  | ПФ, | ФНЧ, | РФ, | ФВЧ, |

**Порядок выполнения работы:**

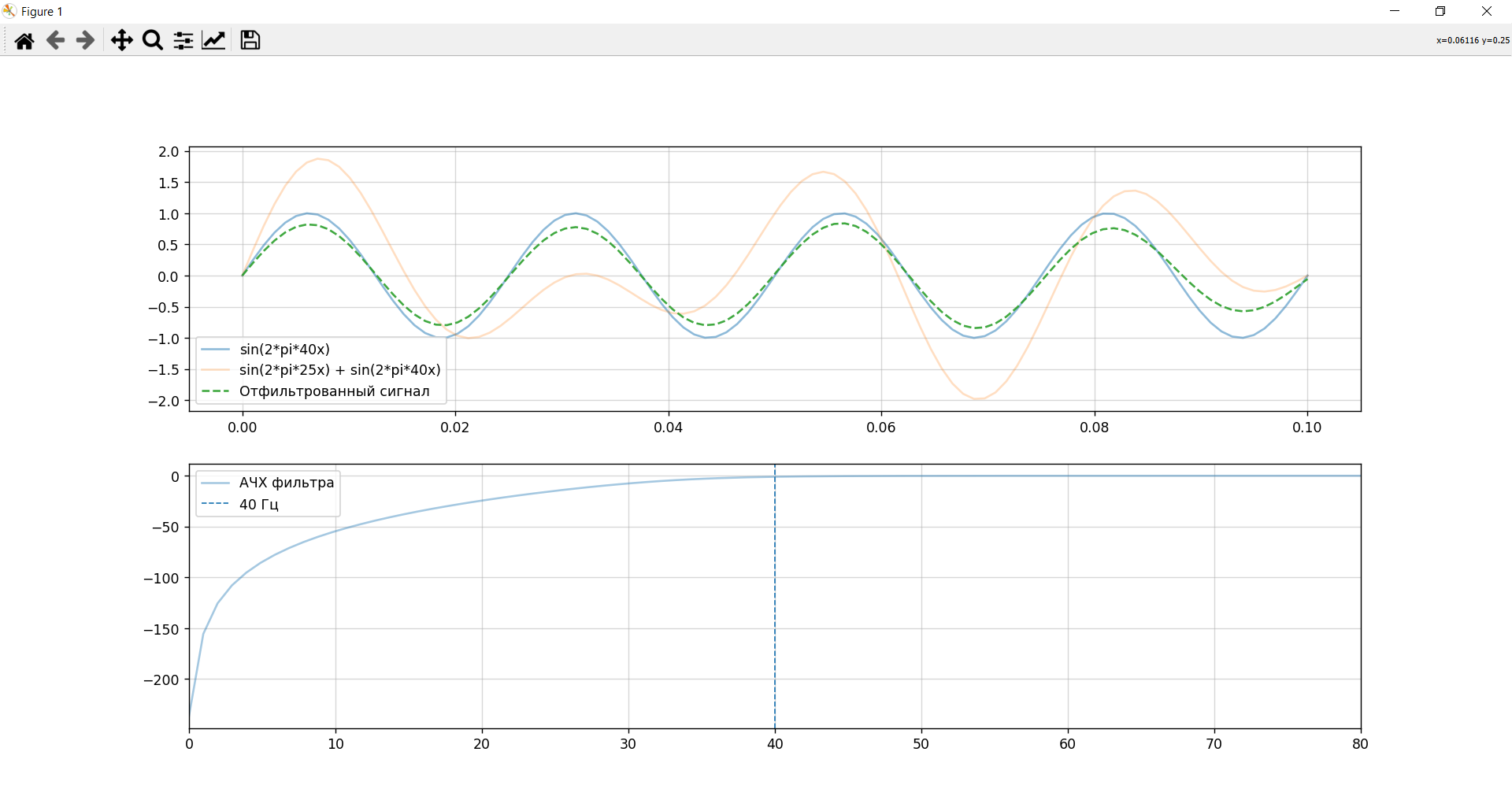


Рис. 1 Фильтрация сигнала с помощью фильтра Баттерворта

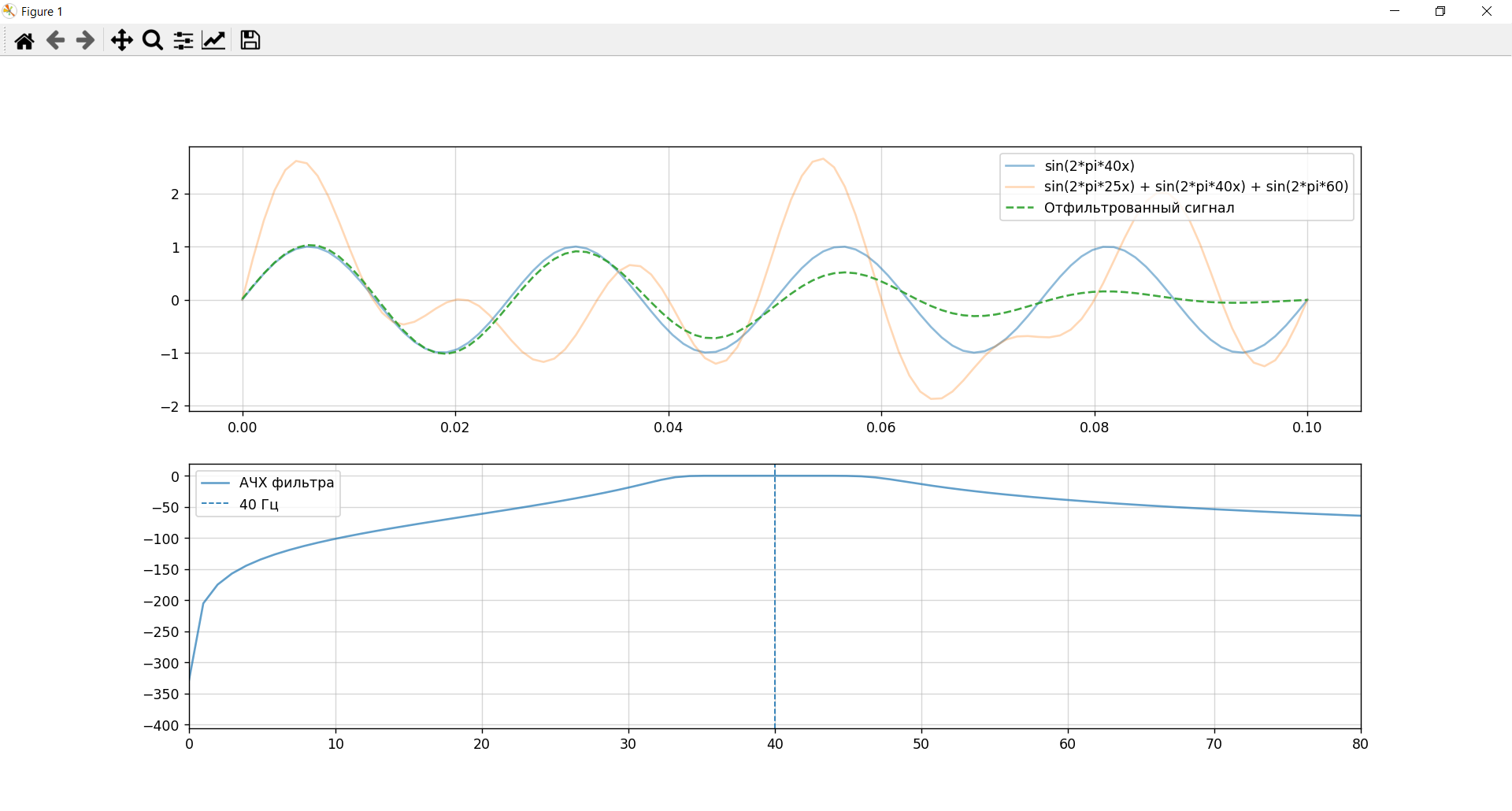


Рис. 2 Фильтрация сигнала с помощью фильтра Баттерворта

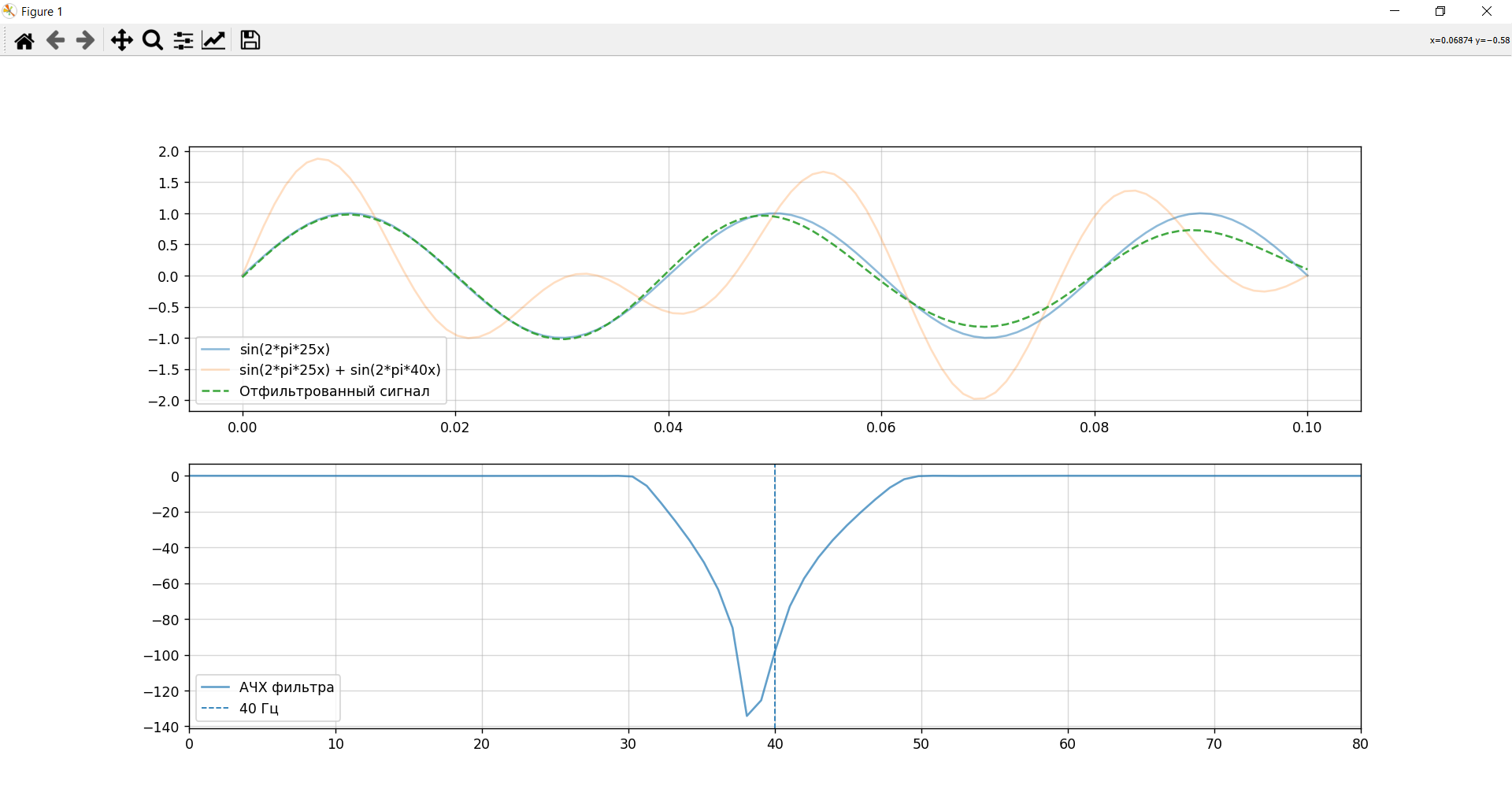


Рис. 3 Фильтрация сигнала с помощью фильтра Чебышева 1 рода

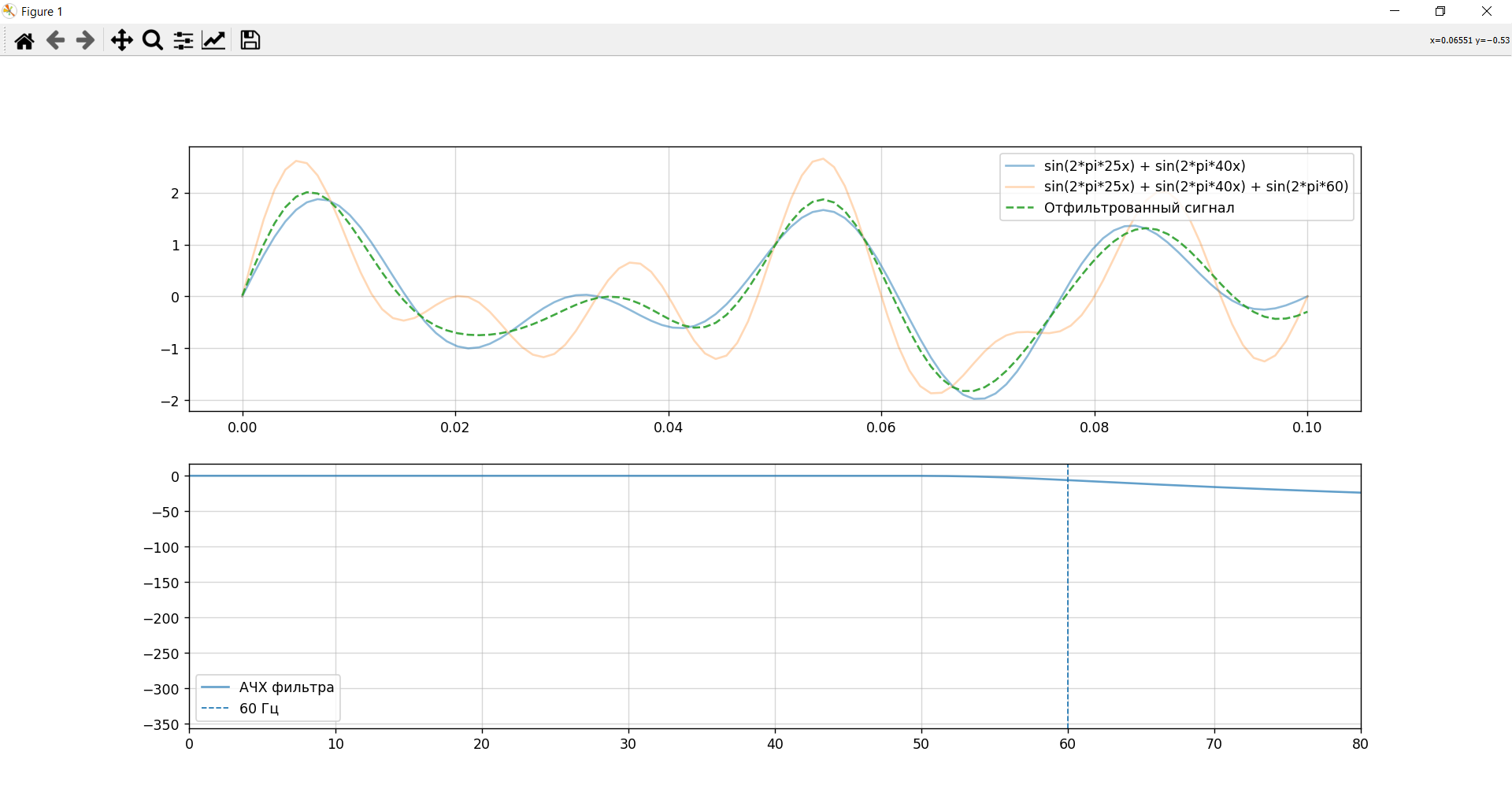


Рис. 4 Фильтрация сигнала с помощью фильтра Чебышева 1 рода

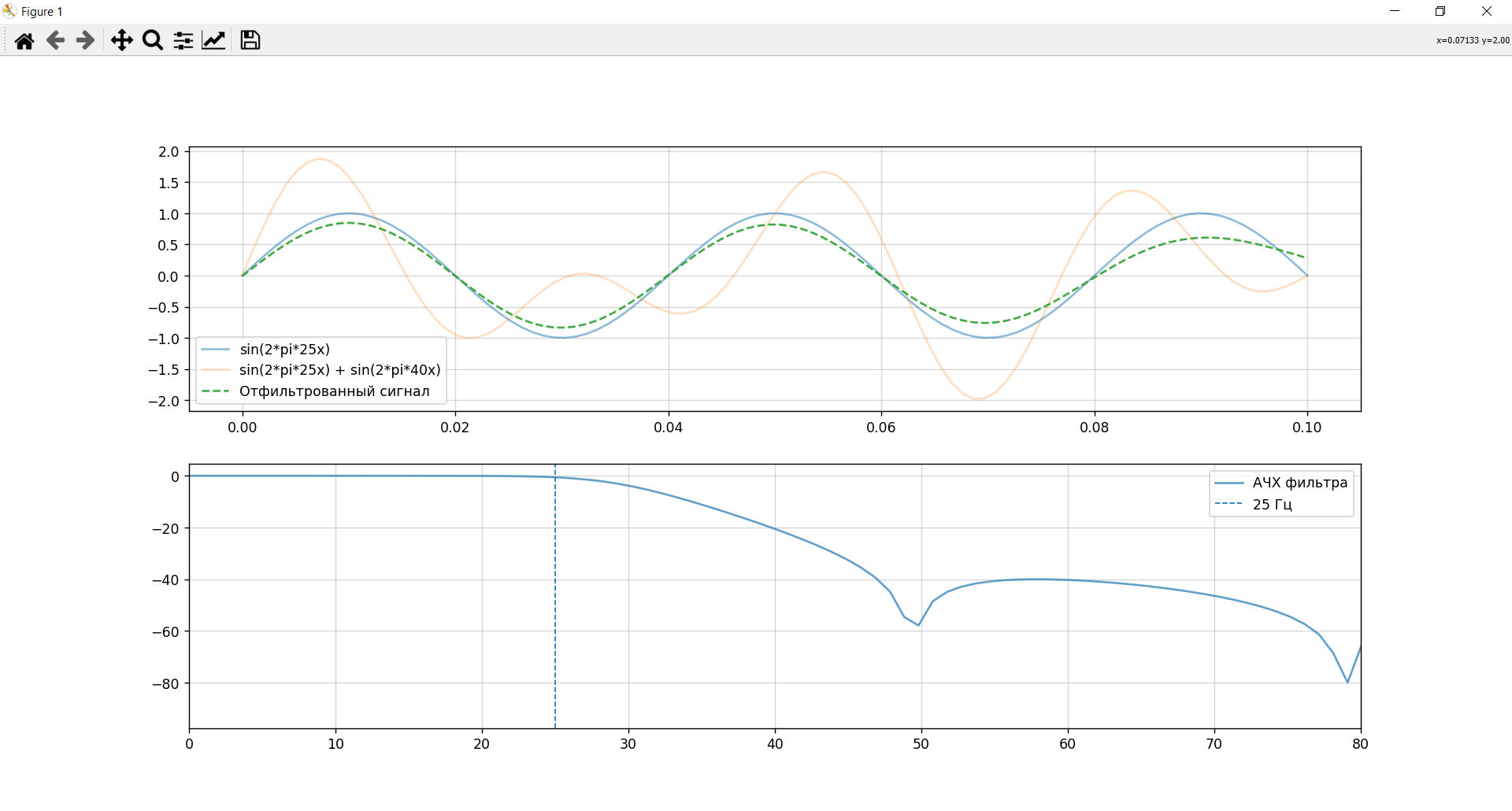


Рис. 5 Фильтрация сигнала с помощью фильтра Чебышева 2 рода

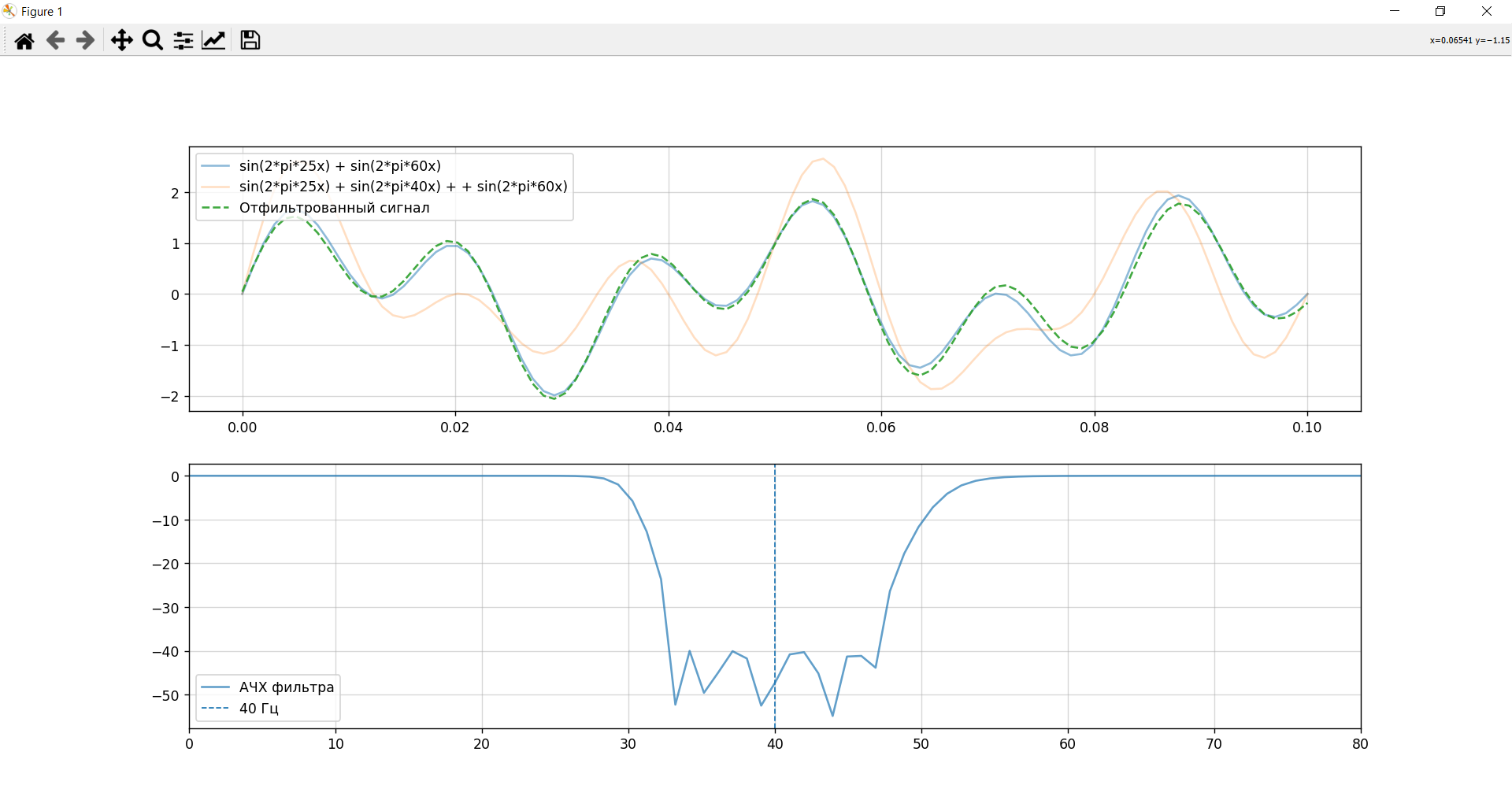


Рис. 6 Фильтрация сигнала с помощью фильтра Чебышева 2 рода

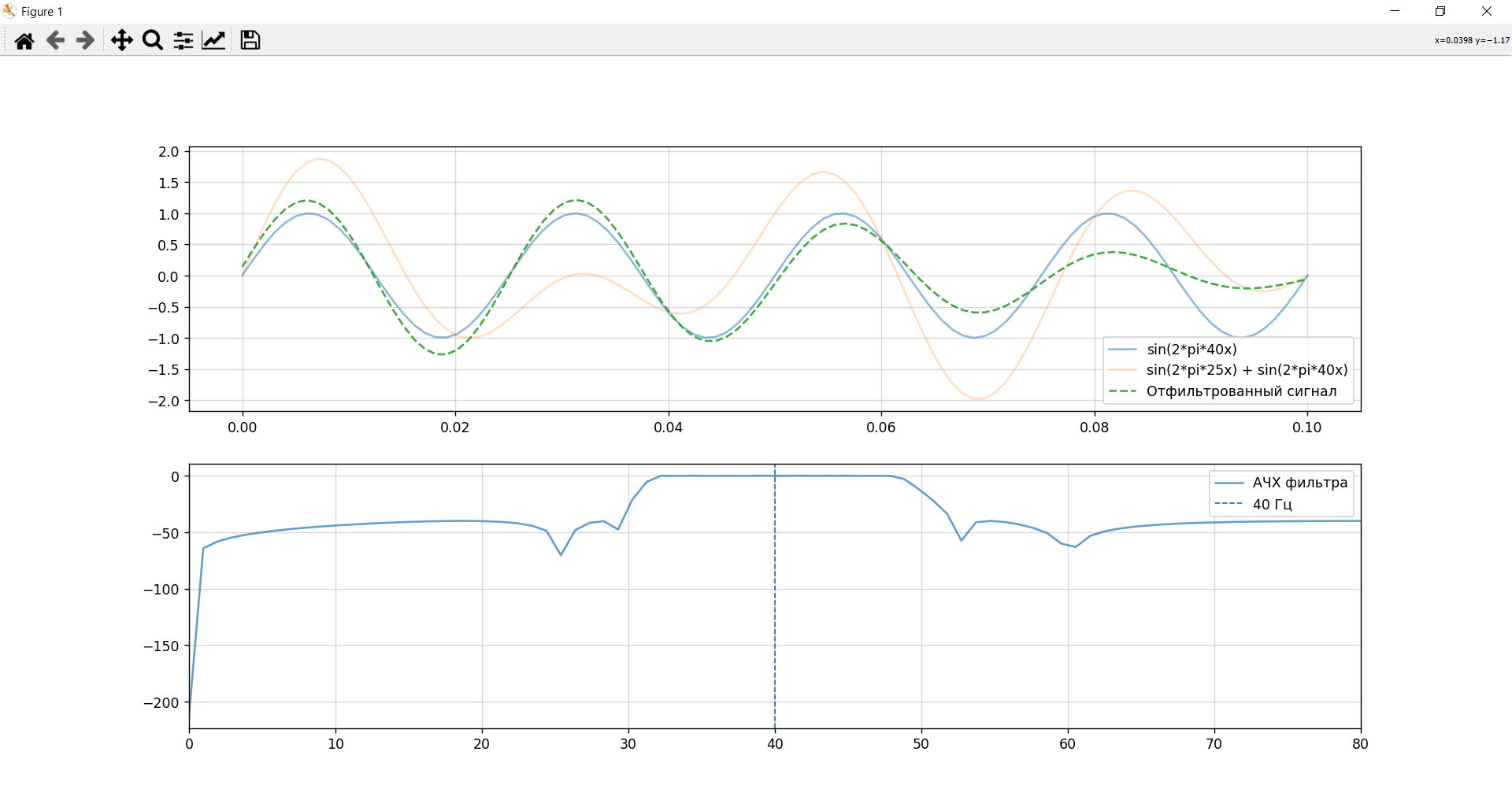


Рис. 7 Фильтрация сигнала с помощью фильтра эллиптического

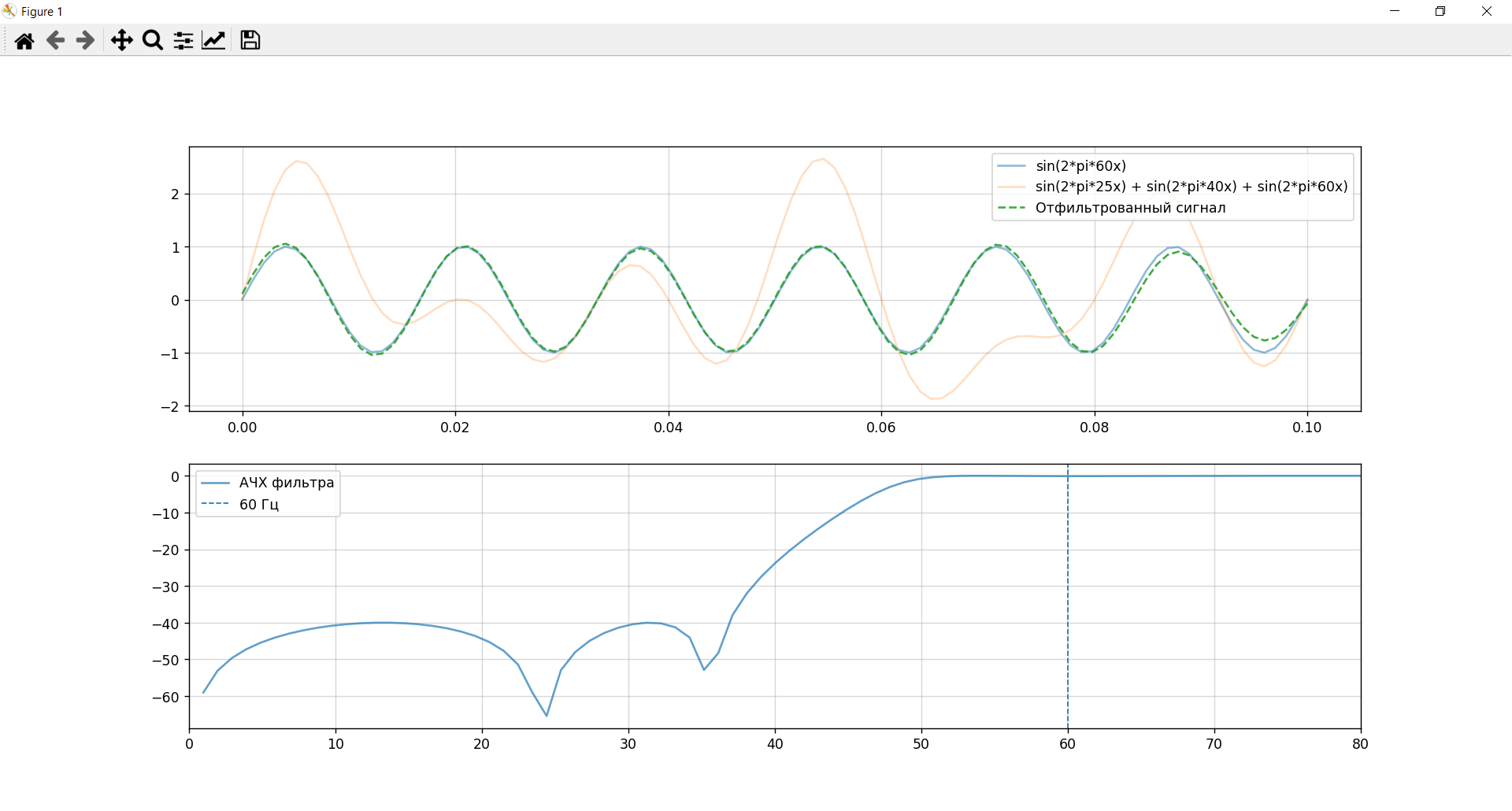


Рис. 8 Фильтрация сигнала с помощью фильтра эллиптического

**Листинг кода:**

**lr4\_butterword1.py**

import numpy as np

import scipy.signal as sgl

import matplotlib.pyplot as plt

#butter, cheb1, cheb2, ellip

#ФВЧ - s2, РФ - s1, ФНЧ - s1, ПФ - s2 //s1+s2

#ПФ - s2, ФНЧ - s1 + s2, РФ - s1 + s3, ФВЧ - s3 // s1+s2+s3

fs = 1000

T = 0.1

s1 = 25

s2 = 40

s3 = 60

delta = 5

n = 5

Rp = 0.1

Rs = 40

x = np.linspace(0, T, int(T\*fs))

signal1 = np.sin(2 \* np.pi \* s1 \* x)

signal2 = np.sin(2 \* np.pi \* s2 \* x)

signal3 = np.sin(2 \* np.pi \* s3 \* x)

noisy = signal1 + signal2

b, a = sgl.butter(n, s2 - delta, 'highpass', fs=fs)

filtered = sgl.filtfilt(b, a, noisy)

w, h = sgl.freqz(b, a, fs=fs)

h = np.abs(h)

plt.figure()

plt.subplot(2, 1, 1)

#plt.plot(x, signal1, alpha=0.5, label=f'sin({s1}x)')

plt.plot(x, signal2, alpha=0.5, label=f'sin(2\*pi\*{s2}x)')

plt.plot(x, noisy, alpha=0.25, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x)')

plt.plot(x, filtered, '--', alpha=0.9, label=f'Отфильтрованный сигнал')

plt.grid(True, alpha=0.5)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(w, 20 \* np.log10(np.abs(h)), alpha=0.4, label=f'АЧХ фильтра')

plt.axvline(x=s2, linestyle='--', linewidth=1, label=f'{s2} Гц')

plt.legend()

plt.grid(True, alpha=0.5)

plt.xlim(0, 2 \* s2)

plt.show()

**lr4\_butterword2.py**

import numpy as np

import scipy.signal as sgl

import matplotlib.pyplot as plt

#butter, cheb1, cheb2, ellip

#ФВЧ - s2, РФ - s1, ФНЧ - s1, ПФ - s2 //s1+s2

#ПФ - s2, ФНЧ - s1 + s2, РФ - s1 + s3, ФВЧ - s3 // s1+s2+s3

fs = 1000

T = 0.1

s1 = 25

s2 = 40

s3 = 60

delta = 7

n = 5

Rp = 0.1

Rs = 40

x = np.linspace(0, T, int(T\*fs))

signal1 = np.sin(2 \* np.pi \* s1 \* x)

signal2 = np.sin(2 \* np.pi \* s2 \* x)

signal3 = np.sin(2 \* np.pi \* s3 \* x)

noisy = signal1 + signal2 + signal3

b, a = sgl.butter(n, [s2 - delta, s2 + delta], 'bandpass', fs=fs)

filtered = sgl.filtfilt(b, a, noisy)

w, h = sgl.freqz(b, a, fs=fs)

h = np.abs(h)

plt.figure()

plt.subplot(2, 1, 1)

#plt.plot(x, signal1, '-', alpha=0.5, label=f'sin({s1}x)')

plt.plot(x, signal2, '-', alpha=0.5, label=f'sin(2\*pi\*{s2}x)')

plt.plot(x, noisy, '-', alpha=0.3, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x) + sin(2\*pi\*{s3})')

plt.plot(x, filtered, '--', alpha=0.9, label=f'Отфильтрованный сигнал')

plt.grid(True, alpha=0.5)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(w, 20 \* np.log10(np.abs(h)), alpha=0.7, label=f'АЧХ фильтра')

plt.axvline(x=s2, linestyle='--', linewidth=1, label=f'{s2} Гц')

plt.legend()

plt.grid(True, alpha=0.5)

plt.xlim(0, 2 \* s2)

plt.show()

**lr4\_cheb11.py**

import numpy as np

import scipy.signal as sgl

import matplotlib.pyplot as plt

#butter, cheb1, cheb2, ellip

#ФВЧ - s2, РФ - s1, ФНЧ - s1, ПФ - s2 //s1+s2

#ПФ - s2, ФНЧ - s1 + s2, РФ - s1 + s3, ФВЧ - s3 // s1+s2+s3

fs = 1000

T = 0.1

s1 = 25

s2 = 40

s3 = 60

delta = 10

n = 5

Rp = 0.1

Rs = 40

x = np.linspace(0, T, int(T\*fs))

signal1 = np.sin(2 \* np.pi \* s1 \* x)

signal2 = np.sin(2 \* np.pi \* s2 \* x)

signal3 = np.sin(2 \* np.pi \* s3 \* x)

noisy = signal1 + signal2

b, a = sgl.cheby1(n, Rp, [s2 - delta, s2 + delta], 'bandstop', fs=fs)

filtered = sgl.filtfilt(b, a, noisy)

w, h = sgl.freqz(b, a, fs=fs)

h = np.abs(h)

plt.figure()

plt.subplot(2, 1, 1)

#plt.plot(x, signal1, alpha=0.5, label=f'sin({s1}x)')

plt.plot(x, signal1, alpha=0.5, label=f'sin(2\*pi\*{s1}x)')

plt.plot(x, noisy, alpha=0.25, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x)')

plt.plot(x, filtered, '--', alpha=0.9, label=f'Отфильтрованный сигнал')

plt.grid(True, alpha=0.5)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(w, 20 \* np.log10(np.abs(h)), alpha=0.7, label=f'АЧХ фильтра')

plt.axvline(x=s2, linestyle='--', linewidth=1, label=f'{s2} Гц')

plt.legend()

plt.grid(True, alpha=0.5)

plt.xlim(0, 2 \* s2)

plt.show()

**lr4\_cheb12.py**

import numpy as np

import scipy.signal as sgl

import matplotlib.pyplot as plt

#butter, cheb1, cheb2, ellip

#ФВЧ - s2, РФ - s1, ФНЧ - s1, ПФ - s2 //s1+s2

#ПФ - s2, ФНЧ - s1 + s2, РФ - s1 + s3, ФВЧ - s3 // s1+s2+s3

fs = 1000

T = 0.1

s1 = 25

s2 = 40

s3 = 60

delta = 10

n = 5

Rp = 0.1

Rs = 40

x = np.linspace(0, T, int(T\*fs))

signal1 = np.sin(2 \* np.pi \* s1 \* x)

signal2 = np.sin(2 \* np.pi \* s2 \* x)

signal3 = np.sin(2 \* np.pi \* s3 \* x)

noisy = signal1 + signal2 + signal3

b, a = sgl.cheby1(n, Rp, s3 - delta, 'lowpass', fs=fs)

filtered = sgl.filtfilt(b, a, noisy)

w, h = sgl.freqz(b, a, fs=fs)

h = np.abs(h)

plt.figure()

plt.subplot(2, 1, 1)

#plt.plot(x, signal1, '-', alpha=0.5, label=f'sin({s1}x)')

#plt.plot(x, signal2, '-', alpha=0.5, label=f'sin({s2}x)')

plt.plot(x, signal1 + signal2, '-', alpha=0.5, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x)')

plt.plot(x, noisy, '-', alpha=0.3, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x) + sin(2\*pi\*{s3})')

plt.plot(x, filtered, '--', alpha=0.9, label=f'Отфильтрованный сигнал')

plt.grid(True, alpha=0.5)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(w, 20 \* np.log10(np.abs(h)), alpha=0.7, label=f'АЧХ фильтра')

plt.axvline(x=s3, linestyle='--', linewidth=1, label=f'{s3} Гц')

plt.legend()

plt.grid(True, alpha=0.5)

plt.xlim(0, 2 \* s2)

plt.show()

**lr4\_cheb21.py**

import numpy as np

import scipy.signal as sgl

import matplotlib.pyplot as plt

#butter, cheb1, cheb2, ellip

#ФВЧ - s2, РФ - s1, ФНЧ - s1, ПФ - s2 //s1+s2

#ПФ - s2, ФНЧ - s1 + s2, РФ - s1 + s3, ФВЧ - s3 // s1+s2+s3

fs = 1000

T = 0.1

s1 = 25

s2 = 40

s3 = 60

delta = 22

n = 5

Rp = 0.1

Rs = 40

x = np.linspace(0, T, int(T\*fs))

signal1 = np.sin(2 \* np.pi \* s1 \* x)

signal2 = np.sin(2 \* np.pi \* s2 \* x)

signal3 = np.sin(2 \* np.pi \* s3 \* x)

noisy = signal1 + signal2

b, a = sgl.cheby2(n, Rs, s1 + delta, 'lowpass', fs=fs)

filtered = sgl.filtfilt(b, a, noisy)

w, h = sgl.freqz(b, a, fs=fs)

h = np.abs(h)

plt.figure()

plt.subplot(2, 1, 1)

#plt.plot(x, signal1, alpha=0.5, label=f'sin(2\*pi\*{s1}x)')

plt.plot(x, signal1, alpha=0.5, label=f'sin(2\*pi\*{s1}x)')

plt.plot(x, noisy, alpha=0.25, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x)')

plt.plot(x, filtered, '--', alpha=0.9, label=f'Отфильтрованный сигнал')

plt.grid(True, alpha=0.5)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(w, 20 \* np.log10(np.abs(h)), alpha=0.7, label=f'АЧХ фильтра')

plt.axvline(x=s1, linestyle='--', linewidth=1, label=f'{s1} Гц')

plt.legend()

plt.grid(True, alpha=0.5)

plt.xlim(0, 2 \* s2)

plt.show()

**lr4\_cheb22.py**

import numpy as np

import scipy.signal as sgl

import matplotlib.pyplot as plt

#butter, cheb1, cheb2, ellip

#ФВЧ - s2, РФ - s1, ФНЧ - s1, ПФ - s2 //s1+s2

#ПФ - s2, ФНЧ - s1 + s2, РФ - s1 + s3, ФВЧ - s3 // s1+s2+s3

fs = 1000

T = 0.1

s1 = 25

s2 = 40

s3 = 60

delta = 7

n = 5

Rp = 0.1

Rs = 40

x = np.linspace(0, T, int(T\*fs))

signal1 = np.sin(2 \* np.pi \* s1 \* x)

signal2 = np.sin(2 \* np.pi \* s2 \* x)

signal3 = np.sin(2 \* np.pi \* s3 \* x)

noisy = signal1 + signal2 + signal3

b, a = sgl.cheby2(n, Rs, [s2 - delta, s2 + delta], 'bandstop', fs=fs)

filtered = sgl.filtfilt(b, a, noisy)

w, h = sgl.freqz(b, a, fs=fs)

h = np.abs(h)

plt.figure()

plt.subplot(2, 1, 1)

#plt.plot(x, signal1, alpha=0.5, label=f'sin({s1}x)')

plt.plot(x, signal1 + signal3, alpha=0.5, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s3}x)')

plt.plot(x, noisy, alpha=0.25, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x) + + sin(2\*pi\*{s3}x)')

plt.plot(x, filtered, '--', alpha=0.9, label=f'Отфильтрованный сигнал')

plt.grid(True, alpha=0.5)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(w, 20 \* np.log10(np.abs(h)), alpha=0.7, label=f'АЧХ фильтра')

plt.axvline(x=s2, linestyle='--', linewidth=1, label=f'{s2} Гц')

plt.legend()

plt.grid(True, alpha=0.5)

plt.xlim(0, 2 \* s2)

plt.show()

**lr4\_ellip1.py**

import numpy as np

import scipy.signal as sgl

import matplotlib.pyplot as plt

#butter, cheb1, cheb2, ellip

#ФВЧ - s2, РФ - s1, ФНЧ - s1, ПФ - s2 //s1+s2

#ПФ - s2, ФНЧ - s1 + s2, РФ - s1 + s3, ФВЧ - s3 // s1+s2+s3

fs = 1000

T = 0.1

s1 = 25

s2 = 40

s3 = 60

delta = 8

n = 5

Rp = 0.1

Rs = 40

x = np.linspace(0, T, int(T\*fs))

signal1 = np.sin(2 \* np.pi \* s1 \* x)

signal2 = np.sin(2 \* np.pi \* s2 \* x)

signal3 = np.sin(2 \* np.pi \* s3 \* x)

noisy = signal1 + signal2

b, a = sgl.ellip(n, Rp, Rs, [s2 - delta, s2 + delta], 'bandpass', fs=fs)

filtered = sgl.filtfilt(b, a, noisy)

w, h = sgl.freqz(b, a, fs=fs)

h = np.abs(h)

plt.figure()

plt.subplot(2, 1, 1)

#plt.plot(x, signal1, alpha=0.5, label=f'sin({s1}x)')

plt.plot(x, signal2, alpha=0.5, label=f'sin(2\*pi\*{s2}x)')

plt.plot(x, noisy, alpha=0.25, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x)')

plt.plot(x, filtered, '--',alpha=0.9, label=f'Отфильтрованный сигнал')

plt.grid(True, alpha=0.5)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(w, 20 \* np.log10(np.abs(h)), alpha=0.7, label=f'АЧХ фильтра')

plt.axvline(x=s2, linestyle='--', linewidth=1, label=f'{s2} Гц')

plt.legend()

plt.grid(True, alpha=0.5)

plt.xlim(0, 2 \* s2)

plt.show()

**lr4\_ellip2.py**

import numpy as np

import scipy.signal as sgl

import matplotlib.pyplot as plt

#butter, cheb1, cheb2, ellip

#ФВЧ - s2, РФ - s1, ФНЧ - s1, ПФ - s2 //s1+s2

#ПФ - s2, ФНЧ - s1 + s2, РФ - s1 + s3, ФВЧ - s3 // s1+s2+s3

fs = 1000

T = 0.1

s1 = 25

s2 = 40

s3 = 60

delta = 8

n = 5

Rp = 0.1

Rs = 40

x = np.linspace(0, T, int(T\*fs))

signal1 = np.sin(2 \* np.pi \* s1 \* x)

signal2 = np.sin(2 \* np.pi \* s2 \* x)

signal3 = np.sin(2 \* np.pi \* s3 \* x)

noisy = signal1 + signal2 + signal3

b, a = sgl.ellip(n, Rp, Rs, s3 - delta, 'highpass', fs=fs)

filtered = sgl.filtfilt(b, a, noisy)

w, h = sgl.freqz(b, a, fs=fs)

h = np.abs(h)

plt.figure()

plt.subplot(2, 1, 1)

plt.plot(x, signal3, alpha=0.5, label=f'sin(2\*pi\*{s3}x)')

plt.plot(x, noisy, alpha=0.25, label=f'sin(2\*pi\*{s1}x) + sin(2\*pi\*{s2}x) + sin(2\*pi\*{s3}x)')

plt.plot(x, filtered, '--', alpha=0.9, label=f'Отфильтрованный сигнал')

plt.grid(True, alpha=0.5)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(w, 20 \* np.log10(np.abs(h)), alpha=0.7, label=f'АЧХ фильтра')

plt.axvline(x=s3, linestyle='--', linewidth=1, label=f'{s3} Гц')

plt.legend()

plt.grid(True, alpha=0.5)

plt.xlim(0, 2 \* s2)

plt.show()

**Вывод:** в ходе выполнения лабораторной работы были сформированы практические навыки разложения сигналов различного вида в ряд Фурье и моделирование сигналов различной формы с заданными параметрами.