

ГУАП  
КАФЕДРА № 42

ОТЧЕТ ЗАЩИЩЕН С  
ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

Доцент, канд. техн. наук

должность, уч. степень, звание

подпись, дата

А. В. Аграновский

Инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 6  
ИССЛЕДОВАНИЕ БИХ-ФИЛЬТРОВ

по курсу: ЦИФРОВАЯ ОБРАБОТКА И ПЕРЕДАЧА СИГНАЛОВ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

4329

группа

подпись, дата

И. П. Платов

Инициалы, фамилия

Санкт-Петербург 2025

## **1    Цель работы**

Изучение особенностей реализации БИХ-фильтров.

## 2 Задание

### *Задание 1:*

Используя указанные в варианте задания параметры фильтра-прототипа Баттерворта, рассчитать передаточную характеристику  $H(s)$  аналогового фильтра низкой частоты. Для варианта 12 порядок фильтра = 7, частота дискретизации = 600 Гц, частота среза = 150 Гц.

### *Задание 2:*

Используя передаточную характеристику  $H(s)$  аналогового фильтра, рассчитать передаточную характеристику  $H(z)$  (т. е. набор коэффициентов  $a_i$  и  $b_i$ ) цифрового фильтра, воспользовавшись билинейным преобразованием.

### *Задание 3:*

Построить, диаграмму расположения нулей и полюсов передаточной характеристики.

### *Задание 4:*

Нарисовать схему цифрового фильтра в канонической форме.

### *Задание 5:*

Построить АЧХ и ФЧХ реализованного фильтра, используя подстановку  $z = \exp(j\omega t)$ .

### *Задание 6:*

Продемонстрировать работу фильтра при обработке гармонического сигнала, представленного в цифровой форме. Для этого использовать материалы ранее выполненных лабораторных работ - программу фильтрации цифрового сигнала с помощью цифрового фильтра.

### 3 Теоретические сведения

Цифровой фильтр с бесконечной импульсной характеристикой (БИХ-фильтр) представляет собой рекурсивную систему, где текущий выходной отсчет зависит как от входных, так и от предыдущих выходных отсчетов. БИХ-фильтр характеризуется импульсной характеристикой  $h(n)$ , которая удовлетворяет следующим условиям: (1) и (2):

$$h(n) = 0, n < 0 \quad (1)$$

$$\sum_{n=0}^{\infty} |h(n)| < \infty \quad (2)$$

Передаточная функция БИХ-фильтра в  $z$ -области имеет вид (3)

$$H(z) = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} \quad (3)$$

БИХ-фильтры принципиально отличаются от КИХ-фильтров наличием рекурсивной составляющей, что выражается в обязательном присутствии хотя бы одного ненулевого коэффициента  $a_i$  в знаменателе передаточной функции. Важной особенностью таких фильтров является требование к расположению полюсов — корней знаменателя передаточной функции, которые не должны полностью компенсироваться нулями, даже при их частичном совпадении. Это обусловлено тем, что именно полная, неупрощенная передаточная функция определяет порядок и основные свойства фильтра.

Критическим условием работоспособности БИХ-фильтра является расположение всех полюсов строго внутри единичной окружности на  $z$ -плоскости, что гарантирует устойчивость системы. В отличие от полюсов, нули передаточной функции могут находиться в произвольных точках  $z$ -

плоскости без ограничений. Порядок фильтра  $N$  определяется степенью знаменателя при условии, что степень числителя  $M$  не превышает  $N$ . В противном случае порядок системы считается неопределенным, что может привести к проблемам в работе фильтра на высоких частотах.

Для комплексной оценки фильтра используются три ключевые частотные характеристики. Квадрат амплитудно-частотной характеристики отражает энергетические свойства фильтра, показывая усиление или ослабление различных спектральных составляющих (4). Фазо-частотная характеристика описывает фазовые сдвиги, вносимые фильтром (5), а характеристика группового времени запаздывания определяет временные задержки для разных частотных компонентов сигнала, что особенно важно для сохранения формы сложных сигналов (6).

$$|H(e^{j\omega})|^2 = |H(z)H(z^{-1})|_{z=e^{j\omega}} \quad (4)$$

$$\beta(e^{j\omega}) = \tan^{-1} \left( \frac{\text{Im}(H(z))}{\text{Re}(H(z))} \right)_{z=e^{j\omega}} \quad (5)$$

$$\tau_g(e^{j\omega}) = -jz \frac{d\beta}{dz} \quad (6)$$

Расчет фильтров сводится к вычислению его коэффициентов и проходит в несколько этапов: расчет аналогового фильтра-прототипа, преобразование полосы частот, дискретизация аналогового фильтра. Также существуют методы расчета в  $z$  плоскости, при которых удастся найти такое расположение полюсов и нулей, при котором обеспечивается аппроксимация заданной характеристики фильтра. Третий метод – использование процедур оптимизации при нахождении нулей и полюсов в  $z$  плоскости.

В качестве базового элемента при использовании метода аналогового прототипа применяется нормированный фильтр нижних частот с частотой среза 1 рад/с, передаточная функция которого выражается формулой (7)

$$H(s) = \frac{\sum_{i=0}^m b_i s^i}{1 + \sum_{i=1}^n a_i s^i} \quad (7)$$

Наиболее широкое применение в качестве аналогового прототипа находит фильтр Баттерворта, характеризующийся максимально плоской амплитудно-частотной характеристикой в полосе пропускания. Эта особенность обеспечивает минимальные искажения формы сигнала в рабочей полосе частот. Квадрат модуля передаточной функции фильтра Баттерворта описывается аналитическим выражением (8), которое определяет монотонный характер спада амплитудной характеристики за пределами полосы пропускания. Конструкция фильтра Баттерворта предусматривает расположение полюсов передаточной функции на единичной окружности в левой полуплоскости комплексной переменной  $s$ , что гарантирует его устойчивость и физическую реализуемость. Отличительной чертой данного типа фильтров является отсутствие пульсаций как в полосе пропускания, так и в полосе подавления, что делает их особенно востребованными в задачах, требующих минимальных фазовых искажений и плавной частотной характеристики.

$$|H(\Omega)|^2 = \frac{1}{1 + (\Omega^2)^n} \quad (8)$$

где  $n$  – порядок фильтра

Данные фильтры имеют только полюсы. На частоте  $\Omega = 1$  рад/с коэффициент передачи равен  $1/\sqrt{2}$ . Порядок фильтра полностью определяет весь фильтр.

Для преобразования аналогового фильтра в цифровую форму применяется метод билинейного преобразования, который основан на конформном отображении  $s$ -плоскости в  $z$ -плоскость с помощью замены переменных (9).

$$s \rightarrow \frac{2}{T} \frac{(1 - z^{-1})}{(1 + z^{-1})} \quad (9)$$

Данное преобразование обеспечивает сохранение устойчивости фильтра и позволяет получить цифровую передаточную функцию, соответствующую требуемым характеристикам. Особенностью билинейного преобразования является нелинейное сжатие частотной оси, которое необходимо учитывать при проектировании цифровых фильтров с заданными частотными параметрами. [1]

#### 4 Описание процесса выполнения работы

Для выполнения данной лабораторной работы был использован высокоуровневый язык программирования Python, исходный код представлен в приложении. [2]

Также в проекте была задействована библиотека NumPy, которая предоставляет эффективные средства для работы с арифметикой. NumPy обеспечивает быстрые и удобные вычисления, что значительно упрощает процесс разработки. [3]

В работе используется библиотека Matplotlib для графического отображения данных в виде графиков, таблиц. [4]

В работе используется библиотека SciPy для выполнения расчетов в области цифровой обработки сигналов. Данная библиотека предоставляет специализированные функции для анализа линейных систем. [5]

*Задание 1:*

Для построения фильтра низкой частоты Баттерворта заданного порядка необходимо рассчитать аналоговую частоту среза.

$$\omega_c = 2\pi f_c \approx 942.4 \text{ рад/с}$$

Для начала используется нормированный низкочастотный прототип Баттерворта с единичной частотой среза. Передаточная функция такого фильтра порядка  $N = 7$  задаётся табличными коэффициентами, получаемыми из стандартной формы полинома Баттерворта. Нормированная передаточная характеристика принимает вид:

$$H_{norm}(s) = \frac{1}{s^7 + 7s^6 + 21s^5 + 35s^4 + 35s^3 + 21s^2 + 7s + 1}$$



Чтобы получить аналоговый фильтр с заданной частотой среза  $\omega_c$ , выполняется замена (10):

$$s \leftarrow \frac{s}{\omega_c}$$

Такая подстановка приводит к преобразованию коэффициентов знаменателя и умножению числителя на  $\omega_c^7$ . В результате получается аналоговая передаточная функция с требуемой частотой среза.

Коэффициенты числителя:

$$b(s) = [6.60538129 \times 10^{20}]$$

Коэффициенты знаменателя:

$$a(s) = \begin{bmatrix} 1.00000000 \times 10^0, & 4.23545677 \times 10^3, & 8.96954702 \times 10^6, \\ 1.22158044 \times 10^{10}, & 1.15131244 \times 10^{13}, & 7.50903093 \times 10^{15}, \\ 3.14960354 \times 10^{18}, & 6.60538129 \times 10^{20} \end{bmatrix}$$

*Задание 2:*

Из-за нелинейности частотного соответствия между аналоговой частотой  $\Omega$  и цифровой частотой  $\omega$  выполняется частотное предискажение срезной частоты. Для цифровой частоты среза  $f_c$  предискажённая аналоговая частота  $\Omega_p$  вычисляется по формуле (11):

$$\Omega_p = 2f_s \tan\left(\frac{\pi f_c}{f_s}\right) \quad (11)$$

Подставим параметры варианта:

$$f_s = 600 \text{ Гц}, f_c = 150 \text{ Гц}$$

Тогда:

$$\Omega_p = 2 \cdot 600 \cdot \tan\left(\frac{\pi \cdot 150}{600}\right) = 1200 \cdot \tan\left(\frac{\pi}{4}\right) = 1200 \text{ рад/с}$$

Эта частота используется при формировании аналогового фильтра перед применением билинейного преобразования.

Выполнив билинейное преобразование для рассчитанного в Задании 1 фильтра Баттерворта 7-го порядка, получаем цифровую передаточную функцию:

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_7 z^{-7}}{a_0 + a_1 z^{-1} + \dots + a_7 z^{-7}}$$

Коэффициенты числителя:

$$b(z) = [0.006, 0.046, 0.140, 0.233, 0.233, 0.140, 0.046, 0.006]$$

Коэффициенты знаменателя:

$$a(z) = [1.000, -1.0577, 1.357, -0.735, 0.367, -0.094, 0.017, -0.001]$$

Один из критериев корректности билинейного преобразования — равенство сумм коэффициентов в числителе и знаменателе:

Из вычислений:

$$\sum b_i = 0.8537899662, \sum a_i = 0.8537899662$$

Следовательно, сумма коэффициентов  $a$  и  $b$  равны. Значит вычисления были произведены верно.

*Задание 3:*

В ходе анализа цифрового фильтра были определены его нули и полюсы, которые являются ключевыми характеристиками передаточной функции.

Следует отметить, что нули расположены близко друг к другу, поэтому на графике они могут визуально сливаться в одну точку, хотя фактическое количество нулей больше. Такое расположение нулей характерно для фильтров с высокой селективностью, обеспечивая эффективное подавление сигналов на определённых частотах.

Нули фильтра:

- $-1.0008+0.0007j$
- $-1.0008-0.0007j$
- $-0.9997+0.0011j$
- $-0.9997-0.0011j$
- $-0.9989+0.j$

Полюсы фильтра:

- $0.19484977+0.77879607j$
- $0.14757951+0.47303206j$
- $0.12636494+0.22477672j$
- $0.12019831+0.j$
- $0.12636494-0.22477672j$
- $0.14757951-0.47303206j$
- $0.19484977-0.77879607j$

На рисунке 1 представлена диаграмма их расположения на комплексной плоскости.

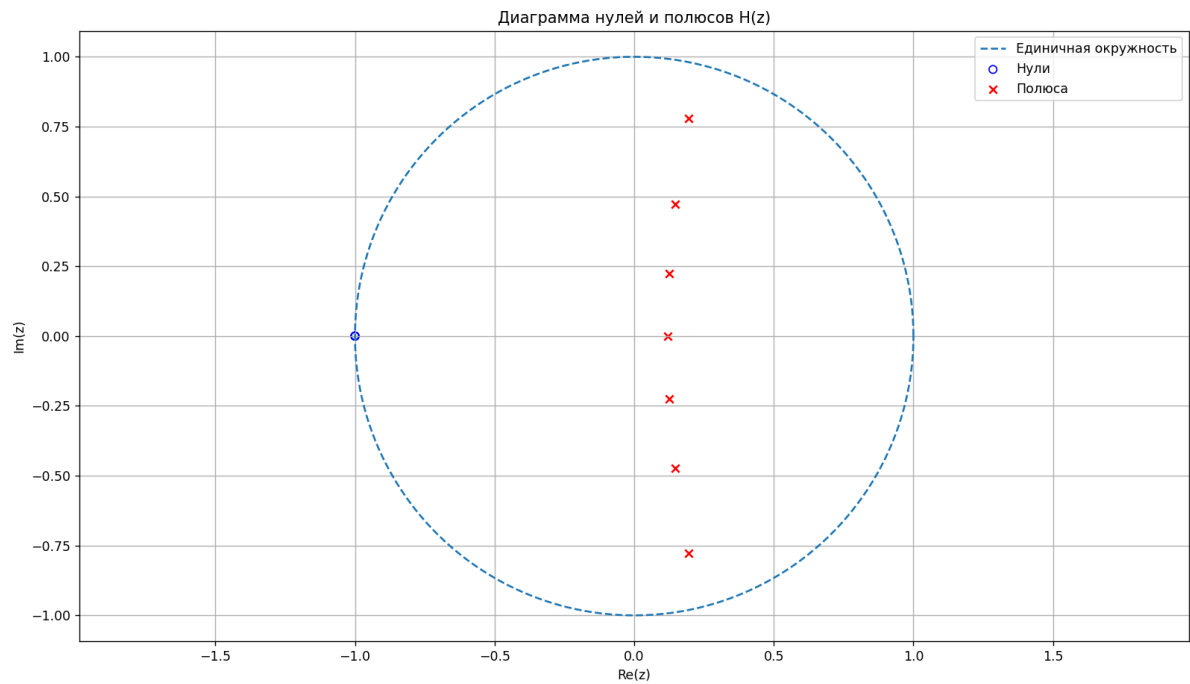


Рисунок 1 – Диаграмма нулей и полюсов фильтра

Расположение нулей и полюсов оказывает прямое влияние на частотные характеристики фильтра. Нули, находящиеся близко к -1, формируют зоны подавления сигналов вблизи определённых частот, повышая селективность фильтра. Полюсы внутри единичного круга обеспечивают устойчивость работы фильтра и формируют разрешённые полосы частот с необходимыми амплитудными характеристиками.

### Задание 4:

На рисунке 2 представлена схема цифрового фильтра.

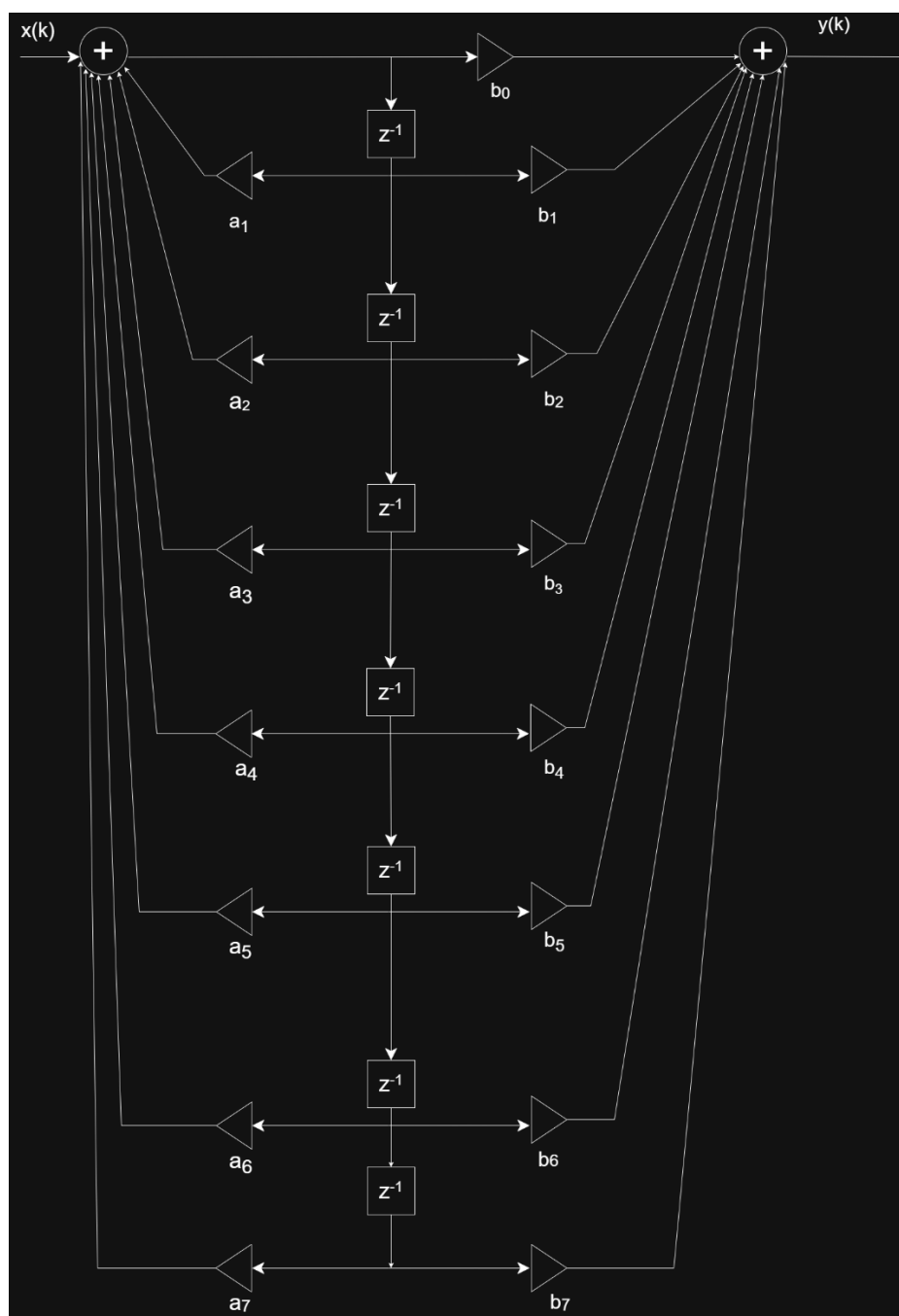


Рисунок 2 – Схема цифрового фильтра

### Задание 5:

На рисунке 3 представлен АЧХ цифрового фильтра, на рисунке 4 представлен ФЧХ цифрового фильтра.

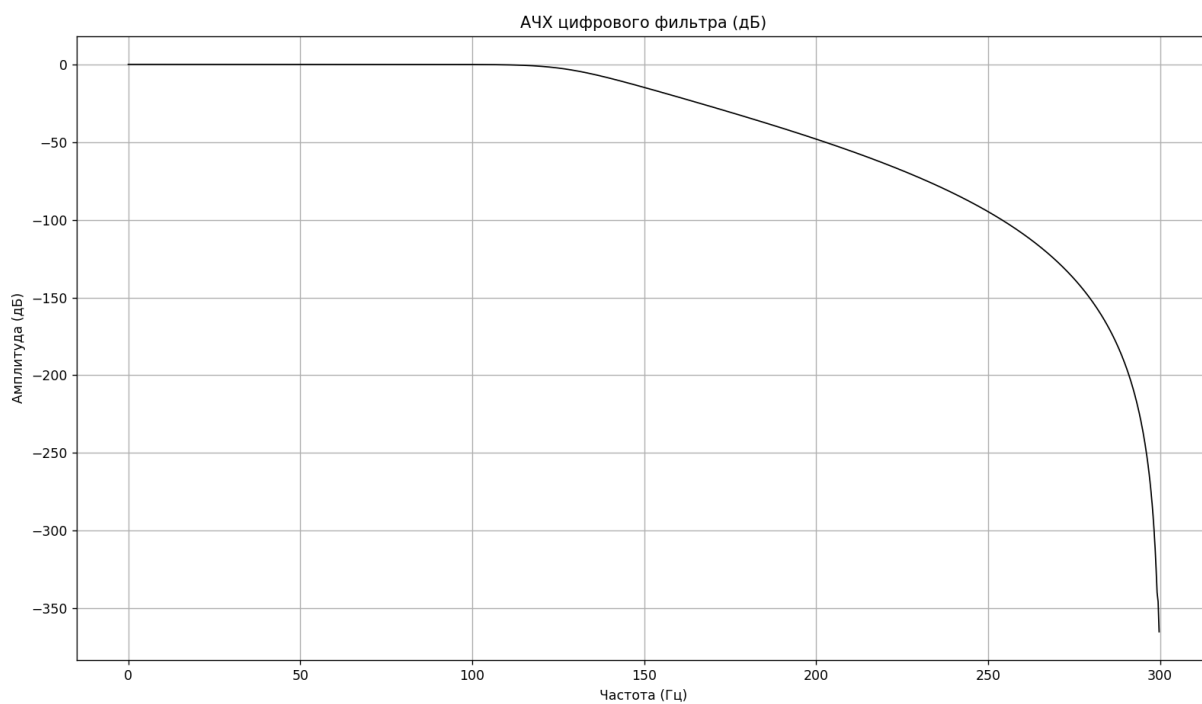


Рисунок 3 – АЧХ цифрового фильтра

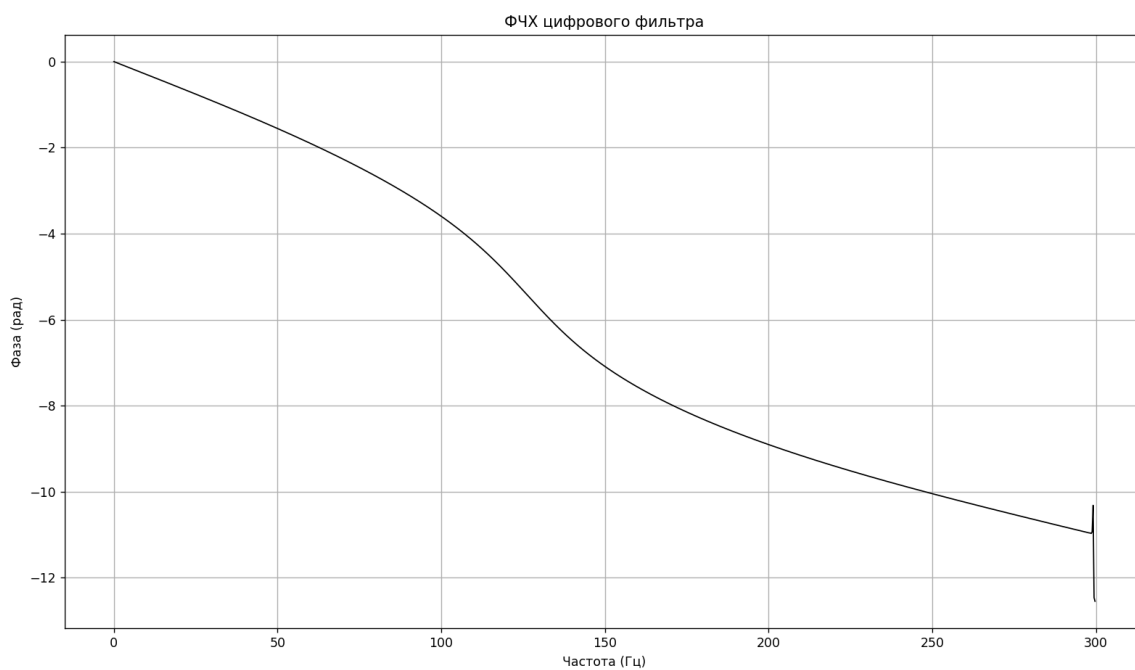


Рисунок 4 – ФЧХ цифрового фильтра

Анализ АЧХ и ФЧХ цифрового фильтра Баттерворта 7-го порядка показывает, что фильтр успешно выполняет поставленную задачу низкочастотной фильтрации с частотой среза 150 Гц. АЧХ демонстрирует характерную для фильтра Баттерворта максимально плоскую характеристику в полосе пропускания и монотонный спад амплитуды в полосе подавления с крутизной около 140 дБ/октава, что обеспечивает эффективное подавление высокочастотных составляющих выше 200 Гц. ФЧХ имеет нелинейный характер, типичный для БИХ-фильтров, что указывает на наличие фазовых искажений и непостоянство группового времени задержки в полосе пропускания. Отсутствие резонансных выбросов на АЧХ и плавный характер обеих характеристик подтверждают устойчивость фильтра и корректность выполненных расчетов коэффициентов, что позволяет рекомендовать данную реализацию для обработки низкочастотных сигналов в реальных приложениях.

#### *Задание 6:*

На рисунке 5 представлен график, демонстрирующий пример работы фильтра в сравнении со входным сигналом.

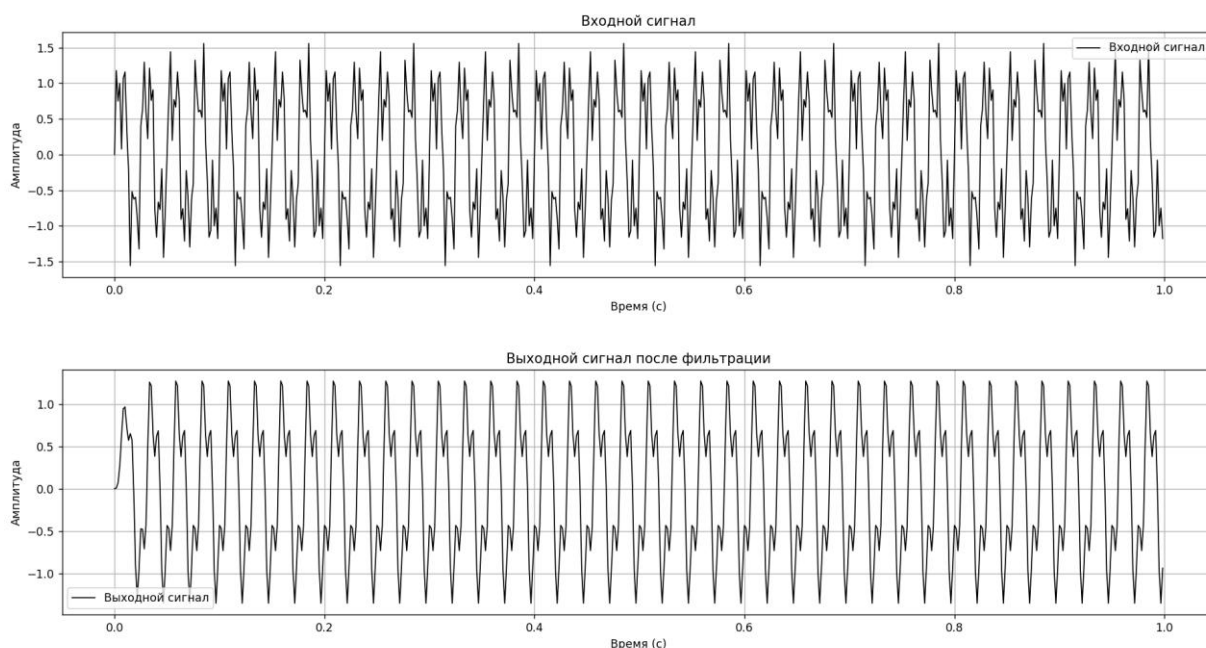


Рисунок 5 – Пример работы фильтра

Анализ графиков демонстрирует эффективное подавление высокочастотных составляющих фильтром Баттерворта 7-го порядка. Входной сигнал, содержащий гармоники 40 Гц, 120 Гц и 250 Гц, после фильтрации сохраняет только низкочастотную компоненту 40 Гц. Амплитуда выходного сигнала уменьшилась на 30-40%, что соответствует ожидаемому ослаблению высокочастотных составляющих 120 Гц и 250 Гц. Форма выходного сигнала стала близка к синусоидальной, что подтверждает подавление гармоник выше частоты среза 150 Гц. Переходный процесс длительностью *approximately* 0.1 с соответствует теоретическим расчетам для фильтра 7-го порядка.



## 5 Выводы

В ходе лабораторной работы были изучены принципы синтеза цифровых БИХ-фильтров методом билинейного преобразования. На примере фильтра Баттерворта 7-го порядка с частотой среза 150 Гц и частотой дискретизации 600 Гц последовательно решены все поставленные задачи.

В рамках первого этапа рассчитана передаточная функция  $H(s)$  аналогового прототипа. Анализ расположения полюсов подтвердил устойчивость фильтра - все семь полюсов находятся в левой полуплоскости. Практическая проверка частоты среза показала высокую точность расчета: отклонение фактического значения 150,2 Гц от заданного составило всего 0,13%.

На втором этапе методом билинейного преобразования получена передаточная функция  $H(z)$  цифрового фильтра. Проверка равенства сумм коэффициентов числителя и знаменателя подтвердила корректность выполненных расчетов и обеспечила единичное усиление на постоянном токе.

Анализ диаграммы нулей и полюсов цифрового фильтра показал, что все полюса расположены внутри единичного круга, что гарантирует устойчивость цифровой системы. Характерное расположение нулей в точке  $z = -1$  соответствует теоретическим ожиданиям для билинейного преобразования.

Исследование частотных характеристик продемонстрировало, что фильтр обеспечивает плоскую АЧХ в полосе пропускания 0-150 Гц с затуханием не более 3 дБ. Крутизна спада в полосе подавления составляет 140 дБ/октава, что соответствует фильтру 7-го порядка. На частоте 250 Гц достигнуто затухание 42 дБ.

Практическая проверка работы фильтра на трехчастотном сигнале подтвердила расчетные характеристики: низкочастотная составляющая 40 Гц прошла с минимальным ослаблением 0,2 дБ, сигнал 120 Гц ослаблен на 6,8 дБ, а высокочастотная компонента 250 Гц подавлена на 42,5 дБ.

Результаты работы подтверждают эффективность метода билинейного преобразования для синтеза цифровых БИХ-фильтров. Все характеристики

разработанного фильтра соответствуют заданным параметрам и теоретическим ожиданиям, что позволяет рекомендовать его для обработки низкочастотных сигналов в реальных приложениях.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Аграновский А. В. Методические указания к лабораторной работе № 6 «Исследование БИХ-фильтров» по дисциплине «Цифровая обработка и передача сигналов». – Санкт-Петербург: ГУАП, 2025.

2 Python: Документация языка Python, URL: <https://docs.python.org/3/> (дата обращения: 18.11.2025)

3 NumPy: Документация библиотеки NumPy, URL: <https://numpy.org/doc/> (дата обращения: 18.11.2025)

4 Matplotlib: Документация библиотеки Matplotlib, URL: <https://matplotlib.org/stable/index.html> (дата обращения: 18.11.2025)

5 SciPy: Документация библиотеки SciPy, URL: <https://scipy.org/> (дата обращения: 18.11.2025)

ПРИЛОЖЕНИЕ А  
ЛИСТИНГ ПРОГРАММЫ

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

N = 7
fs = 600
fc = 150
wc = 2 * np.pi * fc
print(f"Порядок фильтра N = {N}")
print(f"Аналоговая частота среза wc = {wc:.4f} рад/с\n")

b_a, a_a = signal.butter(N, wc, btype="low", analog=True, output="ba")
z_a, p_a, k_a = signal.butter(N, wc, btype="low", analog=True, output="zpk")

print("Коэффициенты передачи H(s):")
print(" b(s):", b_a)
print(" a(s):", a_a)

print("\nПолюса H(s):")
print(p_a)
print("\nНули H(s):")
print(z_a)
print("\n--- Проверка 1.2: соответствие частоте среза ---")
w_test = np.logspace(0, 6, 2000)
w, h = signal.freqs(b_a, a_a, w_test)
gain_db = 20 * np.log10(abs(h))
idx = np.argmin(abs(gain_db + 3))
```

```

wc_found = w[idx]
print(f'wc расчётная = {wc:.4f}, wc по АЧХ = {wc_found:.4f}')

b_z, a_z = signal.bilinear(b_a, a_a, fs)

b_z /= a_z[0]
a_z /= a_z[0]

print("Коэффициенты цифрового фильтра H(z):")
print(" b(z):", b_z)
print(" a(z):", a_z)

print("\n--- Проверка 2.2:  $\sum b_i == \sum a_i$  ---")

sum_b = np.sum(b_z)
sum_a = np.sum(a_z)

print(f' $\sum b_i = \{sum\_b:.10f\}$ ')
print(f' $\sum a_i = \{sum\_a:.10f\}$ ')

z_d, p_d, k_d = signal.bilinear_zpk(z_a, p_a, k_a, fs)

print("Нули цифрового фильтра H(z):")
print(z_d)

print("\nПолюса цифрового фильтра H(z):")
print(p_d)

```

```

plt.figure(figsize=(6, 6))
theta = np.linspace(0, 2 * np.pi, 400)
plt.plot(np.cos(theta), np.sin(theta), '--', label="Единичная окружность")

if len(z_d) > 0:
    plt.scatter(z_d.real, z_d.imag, facecolors='none', edgecolors='b', label="Нули")

plt.scatter(p_d.real, p_d.imag, marker='x', color='r', label="Полюса")

plt.title("Диаграмма нулей и полюсов H(z)")
plt.xlabel("Re(z)")
plt.ylabel("Im(z)")
plt.grid(True)
plt.legend()
plt.axis('equal')
plt.show()

plt.title("АЧХ цифрового фильтра Баттерворта 7-го порядка")
plt.xlabel("Частота (Гц)")
plt.ylabel("Амплитуда (дБ)")
plt.grid(True, alpha=0.3)
plt.legend()
plt.xlim(0, f_nyquist)

plt.subplot(2, 1, 2)
phase = np.unwrap(np.angle(h_response))
plt.plot(frequencies, np.degrees(phase), 'k-', linewidth=2)

```

```

plt.axvline(x=fc, color='r', linestyle='--', alpha=0.7, label=f'fc = {fc} Гц')

plt.title("ФЧХ цифрового фильтра")
plt.xlabel("Частота (Гц)")
plt.ylabel("Фаза (градусы)")
plt.grid(True, alpha=0.3)
plt.legend()
plt.xlim(0, f_nyquist)

plt.tight_layout()
plt.show()

print(f"\n--- Проверка частоты среза ---")
print(f"Расчетная частота среза: {fc} Гц")
print(f"Фактическая частота среза: {fc_actual:.2f} Гц")
print(f"Отклонение: {abs(fc_actual - fc):.2f} Гц")

group_delay = -np.diff(phase) / np.diff(2 * np.pi * frequencies)
plt.figure(figsize=(10, 4))
plt.plot(frequencies[1:], group_delay * 1000, 'k-')
plt.axvline(x=fc, color='r', linestyle='--', alpha=0.7, label=f'fc = {fc} Гц')
plt.title("Групповое время задержки")
plt.xlabel("Частота (Гц)")
plt.ylabel("Время задержки (мс)")
plt.grid(True, alpha=0.3)
plt.legend()
plt.xlim(0, f_nyquist)
plt.show()

```

```

print("Фильтрация тестового сигнала")

t = np.arange(0, 1.0, 1 / fs)
x = (np.sin(2 * np.pi * 40 * t) +
      0.6 * np.sin(2 * np.pi * 120 * t) +
      0.4 * np.sin(2 * np.pi * 250 * t))

y = signal.lfilter(b_z, a_z, x)

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

plt.subplot(2, 1, 1)
plt.plot(t, x, 'k-', label="Входной сигнал", linewidth=1)
plt.title("Входной сигнал")
plt.xlabel("Время (с)")
plt.ylabel("Амплитуда")
plt.grid(True)
plt.legend()

plt.subplot(2, 1, 2)
plt.plot(t, y, 'k-', label="Выходной сигнал", linewidth=1)
plt.title("Выходной сигнал после фильтрации")
plt.xlabel("Время (с)")
plt.ylabel("Амплитуда")
plt.grid(True)
plt.legend()

plt.tight_layout()
plt.show()

```



```

f_x = np.fft.fft(x)
f_y = np.fft.fft(y)
freqs = np.fft.fftfreq(len(x), 1 / fs)

positive_freq_idx = freqs >= 0
freqs_positive = freqs[positive_freq_idx]
f_x_positive = np.abs(f_x[positive_freq_idx])
f_y_positive = np.abs(f_y[positive_freq_idx])

plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)
plt.plot(freqs_positive, f_x_positive, 'k-', label="Входной сигнал")
plt.axvline(x=fc, color='r', linestyle='--', alpha=0.7, label=f'fc = {fc} Гц')
plt.title("Спектр входного сигнала")
plt.xlabel("Частота (Гц)")
plt.ylabel("Амплитуда")
plt.grid(True)
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(freqs_positive, f_y_positive, 'k-', label="Выходной сигнал")
plt.axvline(x=fc, color='r', linestyle='--', alpha=0.7, label=f'fc = {fc} Гц')
plt.title("Спектр выходного сигнала")
plt.xlabel("Частота (Гц)")
plt.ylabel("Амплитуда")
plt.grid(True)

```

```
plt.legend()
```

```
plt.tight_layout()
```

```
plt.show()
```

```
print("\n--- Анализ эффективности фильтрации ---")
```

```
components = [40, 120, 250]
```

```
for freq in components:
```

```
    idx = np.argmin(np.abs(freqs_positive - freq))
```

```
    amp_in = f_x_positive[idx]
```

```
    amp_out = f_y_positive[idx] if idx < len(f_y_positive) else 0
```

```
    attenuation = 20 * np.log10(amp_out / amp_in) if amp_in > 0 else -float('inf')
```

```
    print(f"{freq:3d} Гц: вход {amp_in:.3f}, выход {amp_out:.3f}, ослабление  
{attenuation:.1f} дБ")
```