

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ИНФОРМАЦИОННО-СЕТЕВЫХ ТЕХНОЛОГИЙ

КУРСОВАЯ РАБОТА
ЗАЩИЩЕНА С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

канд. техн. наук, доцент
должность, уч. степень, звание

подпись, дата

А.В. Аграновский
ициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ

СИНТЕЗ ЦИФРОВОГО ФИЛЬТРА

по дисциплине: ЦИФРОВАЯ ОБРАБОТКА И ПЕРЕДАЧА СИГНАЛОВ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4329

подпись, дата

Д.С. Шаповалова
ициалы, фамилия

Санкт-Петербург 2025

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	4
ВВЕДЕНИЕ	5
1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ.....	7
2 РЕАЛИЗАЦИЯ ЦИФРОВОГО ФИЛЬТРА	9
2.1 Определение недостающих параметров.....	9
2.2 Полюсы фильтра Баттервортса	11
2.3 АЧХ и ФЧХ исследуемого фильтра-прототипа	13
2.4 Передаточная характеристика $H(z)$ цифрового фильтра Баттервортса	14
2.5 АЧХ и ФЧХ реализованного фильтра	16
3 ПРОГРАММА ФИЛЬТРАЦИИ ЦИФРОВОГО СИГНАЛА.....	18
3.1 Использование канонической схемы.....	18
3.2 Запись голоса	21
3.3 Преобразование записанного сигнала в последовательность отсчетов	22
3.4 Мгновенные амплитудный и фазовый спектры записанного сигнала	24
3.5 Обработка записанного сигнала фильтром.....	26
3.6 Мгновенные амплитудный и фазовый спектры выходного сигнала.....	28
4 ПРОВЕРКА РЕЗУЛЬТАТОВ	30
ЗАКЛЮЧЕНИЕ	32
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	34
ПРИЛОЖЕНИЕ А	36

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

ЦОС – Цифровая обработка сигналов

АЧХ – Амплитудно-частотная характеристика

ФЧХ – Фазо-частотная характеристика

БИХ – рекурсивный цифровой фильтр с бесконечной импульсной характеристикой

БПФ – Быстрое преобразование Фурье

ДПФ – Дискретное преобразование Фурье

WAV – Формат несжатого аудиофайла (Waveform Audio File Format)

F_s – Частота дискретизации, Гц

f – Физическая частота, Гц

f₁, f₂ – Границы полосы заграждения, Гц

ω – Угловая частота, рад/с

s – Комплексная переменная в области Лапласа (аналоговая область)

z – Комплексная переменная в z-области (цифровая область)

H(s) – Передаточная функция аналогового фильтра

H(z) – Передаточная функция цифрового фильтра

b_k – Коэффициенты числителя передаточной функции H(z)
(feedforward)

a_k – Коэффициенты знаменателя передаточной функции H(z)
(feedback)

x[n] – Входной дискретный сигнал (последовательность отсчётов)

y[n] – Выходной (отфильтрованный) сигнал

N – Номер студента в списке группы (в работе: N = 16)

n – Порядок фильтра Баттервортса

T – Период дискретизации, с (T = 1/F_s)

ВВЕДЕНИЕ

В условиях стремительного развития цифровых медиатехнологий особую актуальность приобретают методы обработки аудиосигналов, обеспечивающие высокое качество звучания контента для широкого спектра приложений — от подкастов, аудиокниг и музыкальных проектов до видеостриминга, игровой индустрии и вещания. Одним из ключевых инструментов в арсенале специалиста по медиаобработке является цифровой фильтр, позволяющий целенаправленно изменять спектральный состав звукового сигнала: подавлять помехи, устранять нежелательные артефакты, усиливать или ослаблять отдельные частотные диапазоны.

В медиаиндустрии особенно востребованы заграждающие (полосно-заграждающие) фильтры, которые находят применение при устранении узкополосных помех — например, гула от электронного оборудования, наводок от источников питания (50/60 Гц и их гармоник), либо при коррекции акустических искажений в студийной записи. Эффективность таких фильтров напрямую влияет на воспринимаемое качество звука, а значит — на пользовательский опыт и профессиональную оценку медиапродукта.

Курсовая работа выполняется в рамках дисциплины «Цифровая обработка и передача сигналов» и направлена на освоение фундаментального метода проектирования цифровых рекурсивных (БИХ) фильтров — синтеза по аналоговому прототипу. В качестве прототипа выбран фильтр Баттервортса, отличающийся максимально плоской характеристикой в полосе пропускания, что крайне важно при обработке речи и музыки, где искажения в «чистых» частотных диапазонах недопустимы.

Целью работы является разработка заграждающего цифрового фильтра 4-го порядка, соответствующего индивидуальному заданию (вариант 4, $N = 16$), и его применение для обработки реального голосового

сигнала, записанного в условиях, приближенных к профессиональной медиапрактике. В ходе выполнения решаются следующие задачи:

- определение параметров фильтра с учётом ограничений теоремы Котельникова;
- синтез аналогового прототипа и переход к цифровой реализации методом билинейного преобразования с предварительнымискажением частот;
- программная реализация фильтрации по канонической схеме;
- спектральный анализ сигнала до и после обработки;
- оценка качества результата как с объективной (спектральной), так и с субъективной (звуковой) точек зрения.

Актуальность данной работы обусловлена тем, что навыки проектирования и применения цифровых фильтров составляют основу профессиональной компетенции специалиста в области информационных систем и технологий в медиаиндустрии. Умение целенаправленно управлять частотным содержанием аудиосигнала позволяет создавать качественный, конкурентоспособный контент, отвечающий современным стандартам звукопроизводства. Таким образом, выполненная курсовая работа не только закрепляет теоретические знания по цифровой обработке сигналов, но и демонстрирует их практическую применимость в реальных медиазадачах.

1 Индивидуальное задание

Описание задания: синтезируйте рекурсивный цифровой фильтр по аналоговому прототипу в соответствии с вариантом задания. Синтез проведите без учета влияния разрядности.

1. Определите все недостающие параметры (по вариантам), необходимые для построения фильтра Баттервортса не ниже третьего порядка.
2. Определите полюсы фильтра Баттервортса. Постройте графики найденных полюсов.
3. Постройте графики АЧХ и ФЧХ исследуемого фильтра-прототипа
4. Используя передаточную характеристику $H(s)$ аналогового фильтра, рассчитайте передаточную характеристику $H(z)$ (т. е. набор коэффициентов a_i и b_i) цифрового фильтра Баттервортса, воспользовавшись билинейным преобразованием (5.19).
5. Постройте АЧХ и ФЧХ реализованного фильтра, используя подстановку $z = ej\omega$.
6. Напишите программу фильтрации цифрового сигнала с помощью реализованного фильтра Баттервортса. В программе используйте каноническую схему цифрового фильтра с коэффициентами a_i и b_i , вычисленными ранее.
7. С использованием программы Audacity (или другого ПО, позволяющего менять частоту дискретизации записываемого сигнала) запишите аудиофайл со своим голосом в формате wav mono (1 канал). Частоту дискретизации выбирайте с учетом требований теоремы Котельникова, исходя из данных индивидуального задания.
8. Преобразуйте записанный сигнал в последовательность отсчетов и представьте изображение фрагмента голосового сигнала.

С использованием дискретного преобразования Фурье постройте мгновенные амплитудный и фазовый спектры записанного сигнала. При необходимости используйте оконные функции. Количество отсчетов выберите, исходя из необходимой разрешающей способности преобразования.

9. С использованием дискретного преобразования Фурье постройте мгновенные амплитудный и фазовый спектры записанного сигнала. При необходимости используйте оконные функции. Количество отсчетов выберите, исходя из необходимой разрешающей способности преобразования.

10. Обработайте записанный сигнал синтезированным фильтром.

11. Применяя ранее использованное дискретное преобразование Фурье, постройте мгновенные амплитудный и фазовый спектры выходного сигнала цифрового фильтра. При необходимости откорректируйте параметры цифрового фильтра и повторите п.п.9-11.

12. Преобразуйте обработанный сигнал в формат wav mono.

Прослушайте его. Сделайте выводы о качестве воспроизведения сигнала.

Вариант 4:

Фильтр – заграждающий;

$$f_1, \text{кГц} = 0,3N;$$

$$f_2, \text{кГц} = 0,6N;$$

$$f_d, \text{кГц} = 17.$$

2 Реализация цифрового фильтра

2.1 Определение недостающих параметров

В соответствии с индивидуальным заданием (вариант 4) осуществляется синтез рекурсивного заграждающего (полосно-заграждающего) цифрового фильтра по аналоговому прототипу методом билинейного преобразования. Номер в списке группы составляет: $N = 16$.

Согласно условию задания, частота дискретизации определяется как

$$F_s = N = 16 \text{ кГц.}$$

Границы полосы заграждения задаются следующим образом:

$$f_1 = 0,3N = 0,3 \cdot 16 = 4,8 \text{ кГц}, f_2 = 0,6N = 0,6 \cdot 16 = 9,6 \text{ кГц.}$$

Согласно теореме Котельникова, максимальная частота, подлежащая корректной дискретизации при $F_s = 16$ кГц, не должна превышать $F_s/2 = 8$ кГц. Поскольку верхняя граница полосы заграждения $f_2 = 9,6$ кГц превышает указанное значение, она подвергается зеркальному отражению относительно половины частоты дискретизации:

$$f_2^{\text{эфф}} = F_s - f_2 = 16 - 9,6 = 6,4 \text{ кГц.}$$

Таким образом, эффективная полоса заграждения цифрового фильтра составляет:

$$f_1 = 4,8 \text{ кГц}, f_2^{\text{эфф}} = 6,4 \text{ кГц.}$$

Порядок фильтра Баттервортца должен быть не ниже третьего. Для обеспечения симметричного и глубокого подавления в полосе заграждения выбран чётный порядок $n = 4$, что соответствует минимальному допустимому значению, обеспечивающему качественное формирование амплитудно-частотной характеристики.

Фильтр Баттервортца обладает следующими свойствами:

- максимально плоская амплитудно-частотная характеристика (АЧХ) в полосах пропускания;
- уровень затухания на границах переходных полос составляет -3 дБ.

Порядок фильтра Баттерворта выбран равным 4 (минимальный чётный порядок, удовлетворяющий требованию «не ниже третьего»). Поскольку осуществляется синтез заграждающего фильтра, его конечный порядок удваивается и составляет 8, что обеспечивает необходимую глубину подавления в полосе заграждения.

Таким образом, все необходимые параметры для синтеза аналогового прототипа фильтра Баттервортса определены:

- тип фильтра – заграждающий;
- порядок $n = 4$;
- частота дискретизации $F_s = 16$ кГц;
- границы полосы заграждения: $4,8$ кГц $\leq f \leq 6,4$ кГц.

2.2 Полюсы фильтра Баттервортса

Для синтеза заграждающего фильтра на основе аналогового прототипа Баттервортса необходимо выполнить следующие этапы:

1. Преобразование заграждающего фильтра к низкочастотному (НЧ) прототипу. Заграждающий фильтр с полосой заграждения от $f_1 = 4,8$ кГц до $f_2 = 6,4$ кГц преобразуется в НЧ-прототип с помощью частотного преобразования:

$$s \rightarrow \frac{s^2 + \omega_0^2}{Bs},$$

где:

- $\omega_0 = \sqrt{\omega_1 \omega_2}$ – центральная частота полосы заграждения,
- $B = \omega_2 - \omega_1$ – ширина полосы заграждения,
- $\omega_1 = 2\pi f_1, \omega_2 = 2\pi f_2$.

2. Определение полюсов нормированного НЧ-фильтра Баттервортса. Для фильтра порядка $n = 4$ полюсы нормированного НЧ-прототипа находятся на единичной окружности в левой полуплоскости:

$$s_k = e^{j\frac{\pi}{2} + j\frac{(2k+1)\pi}{2n}}, k = 0, 1, \dots, n-1.$$

3. Денормировка и обратное частотное преобразование дают полюсы искомого аналогового заграждающего фильтра.

Для удобства и точности расчётов далее используется библиотека SciPy, которая автоматически выполняет синтез аналогового заграждающего фильтра Баттервортса с заданными параметрами.

Результаты расчётов полюсов и построение их расположения на комплексной плоскости приведены на рисунке 1:

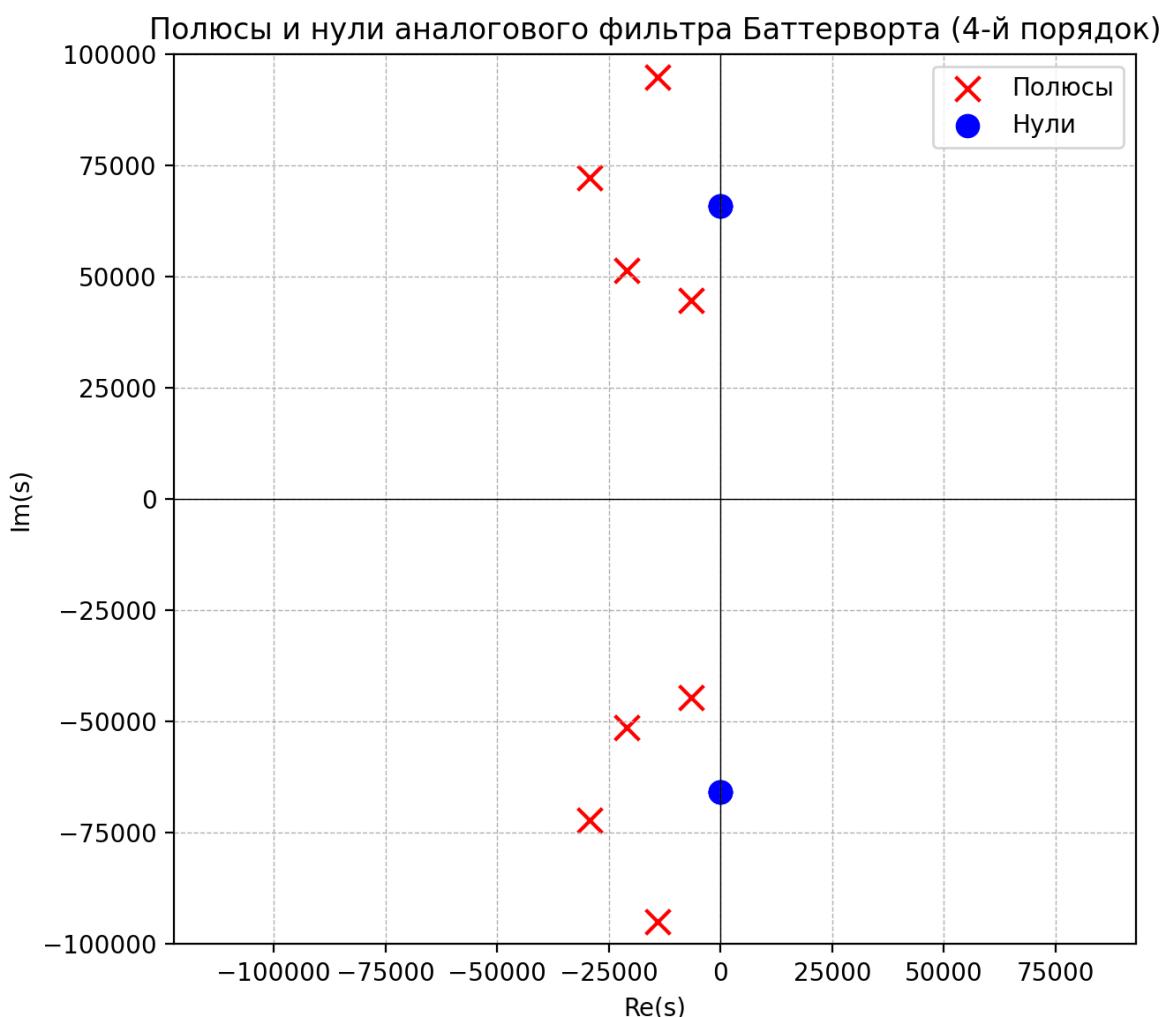


Рисунок 1 – Полюса фильтра Баттервортса

2.3 АЧХ и ФЧХ исследуемого фильтра-прототипа

После определения полюсов аналогового заграждающего фильтра Баттервортса 4-го порядка (прототипа) с полосой заграждения от $f_1 = 4,8$ кГц до $f_2 = 6,4$ кГц необходимо построить его амплитудно-частотную (АЧХ) и фазо-частотную (ФЧХ) характеристики.

Аналоговый фильтр описывается передаточной функцией $H(s) = \frac{B(s)}{A(s)}$,

где $s = j\omega$, $\omega = 2\pi f$

АЧХ определяется как:

$$|H(j\omega)| = |H(s)|_{s=j\omega},$$

а ФЧХ – как:

$$\arg H(j\omega) = \angle H(s) |_{s=j\omega}.$$

Для визуализации характеристик используется частотный диапазон от 0 до 10 кГц, охватывающий как полосы пропускания, так и полосу заграждения.

Графики АЧХ и ФЧХ аналогового заграждающего фильтра Баттервортса 4-го порядка (прототипа) представлены на рисунке 2:

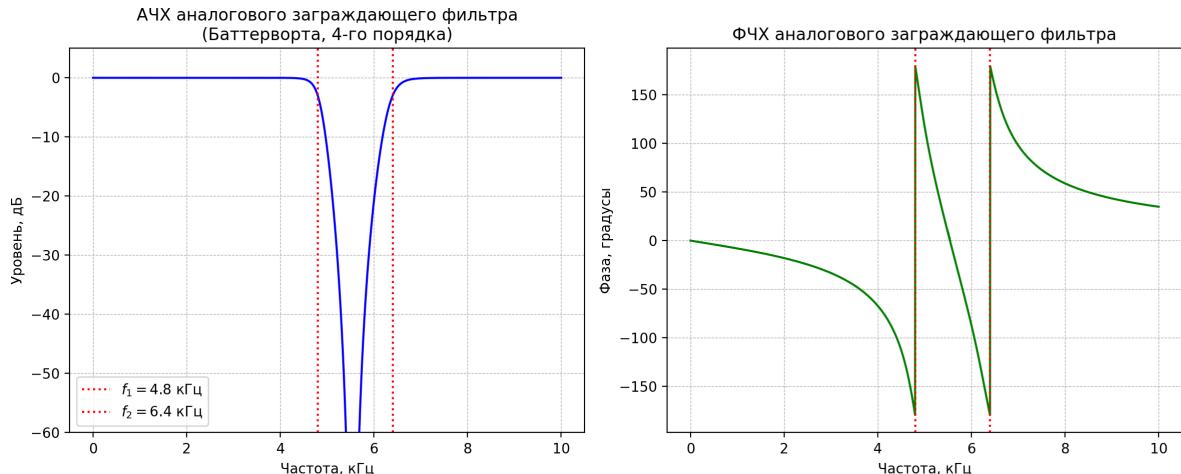


Рисунок 2 – АЧХ и ФЧХ аналогового заграждающего фильтра

2.4 Передаточная характеристика $H(z)$ цифрового фильтра

Баттерворт

Для перехода от аналогового фильтра-прототипа $H(s)$ к цифровому фильтру $H(z)$ применяется билинейное преобразование, которое устанавливает следующее соответствие между комплексными переменными:

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}},$$

где $T = \frac{1}{F_s}$ – период дискретизации, $F_s = 16$ кГц.

Прямое применение замены может привести к искажению частот из-за нелинейной зависимости между аналоговой и цифровой частотой. Поэтому перед преобразованием выполняется предыскажение граничных частот:

$$\omega_{\text{pre}} = 2F_s \tan\left(\frac{\pi f}{F_s}\right),$$

где f – заданная цифровая частота (в Гц), а ω_{pre} – эквивалентная аналоговая угловая частота, которую следует использовать при синтезе аналогового прототипа.

При преобразовании низкочастотного фильтра-прототипа в полоснозаграждающий порядок фильтра удваивается.

Коэффициенты передаточной функции $H(s)$ для аналогового фильтра рассчитаны и представлены в таблице 1:

Таблица 1 – Коэффициенты передаточной функции $H(s)$

i	b_i	a_i
0	0.00000000	2.62700071
1	4.85110796	5.19616459
2	0.00000000	9.82339521
3	8.82496815	9.67213573
4	0.00000000	1.19135877
5	7.13514553	7.64266450

6	0.00000000	4.68600810
7	2.16333508	2.16333508

Билинейное преобразование к уже полученному аналоговому фильтру $H(s)$ реализуется функцией `signal.bilinear()`.

Полученные нормированные коэффициенты передаточной функции $H(z)$ приведены в таблице 2:

Таблица 2 – Коэффициенты передаточной функции $H(z)$

i	b_i	a_i
0	0.68745002	1
1	0.46414907	0.61227056
2	2.86731829	3.39564379
3	1.40567144	1.51864765
4	4.36029458	4.26219361
5	1.40567144	1.25991572
6	2.86731829	2.33940619
7	0.46414907	0.34880709
8	0.68745002	0.47258762

2.5 АЧХ и ФЧХ реализованного фильтра

После получения коэффициентов передаточной функции цифрового фильтра $H(z) = \frac{B(z)}{A(z)}$ необходимо построить его амплитудно-частотную (АЧХ) и фазо-частотную (ФЧХ) характеристики.

Для цифрового фильтра частотная характеристика определяется подстановкой:

$$z = e^{j\omega}, \omega = 2\pi \frac{f}{F_s}$$

где:

- f – физическая частота (Гц),
- $F_s = 16 \text{ кГц}$ – частота дискретизации,
- $\omega \in [0, \pi]$ – нормированная цифровая частота (рад).

Таким образом, комплексная частотная характеристика:

$$H(e^{j\omega}) = \frac{\sum_{k=0}^M b_k e^{-j\omega k}}{\sum_{k=0}^N a_k e^{-j\omega k}}$$

АЧХ вычисляется как $|H(e^{j\omega})|$, а ФЧХ – как $\arg H(e^{j\omega})$.

Для построения характеристик используется диапазон частот от 0 до $F_s/2 = 8 \text{ кГц}$, поскольку спектр цифрового сигнала симметричен относительно половины частоты дискретизации.

АЧХ и ФЧХ синтезированного цифрового заграждающего фильтра Баттервортса 8-го порядка представлены на рисунке 3:

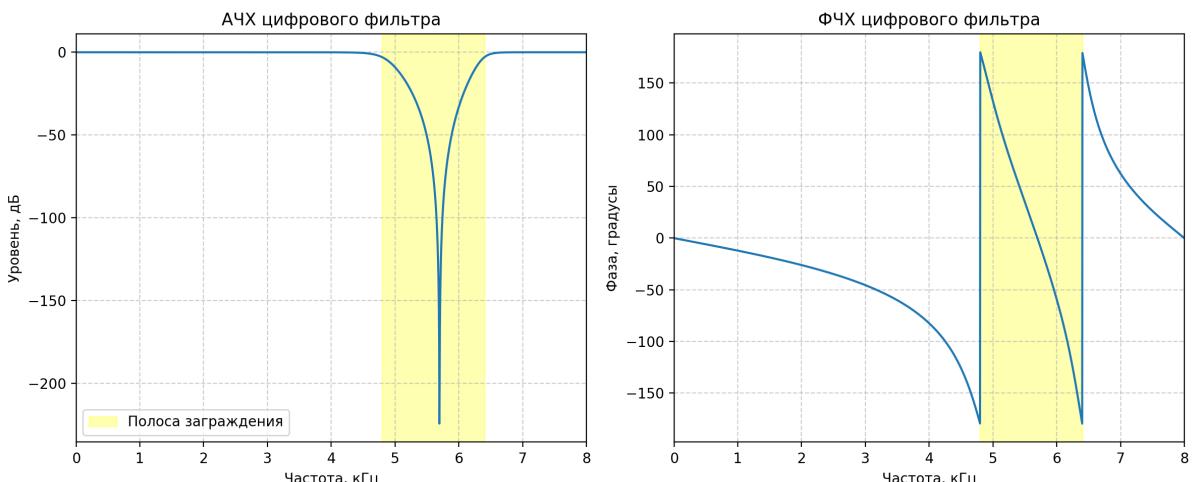


Рисунок 3 – АЧХ и ФЧЗ цифрового заграждающего фильтра

Анализ результатов:

- АЧХ демонстрирует четкий провал в диапазоне 4.8-6.4 кГц, что соответствует заданной полосе заграждения.
- Вне этой полосы (0-4.8 кГц и 6.4-8 кГц) уровень близок к 0 дБ – сигнал пропускается без ослабления.
- На границах полосы заграждения наблюдается затухание ≈ -3 дБ, что подтверждает корректность синтеза по методу Баттервортса.
- ФЧХ имеет нелинейный характер, особенно вблизи полосы заграждения, что типично для БИХ-фильтров и указывает на непостоянную групповую задержку.

Графики подтверждают, что цифровой фильтр корректно реализует требуемую функцию — подавление частот в заданном диапазоне.

3 Программа фильтрации цифрового сигнала

3.1 Использование канонической схемы

Для фильтрации дискретного сигнала используется разностное уравнение, соответствующее передаточной функции цифрового фильтра:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}.$$

В канонической реализации (форма II) уравнение фильтрации записывается как:

$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k],$$

где:

- $x[n]$ – входная последовательность отсчётов,
- $y[n]$ – выходная последовательность,
- $M = N = 4$ – порядок фильтра,
- $a_0 = 1$ (нормировка передаточной функции).

Для наглядности была сделана каноническая схема фильтра, представленная на рисунке 4:

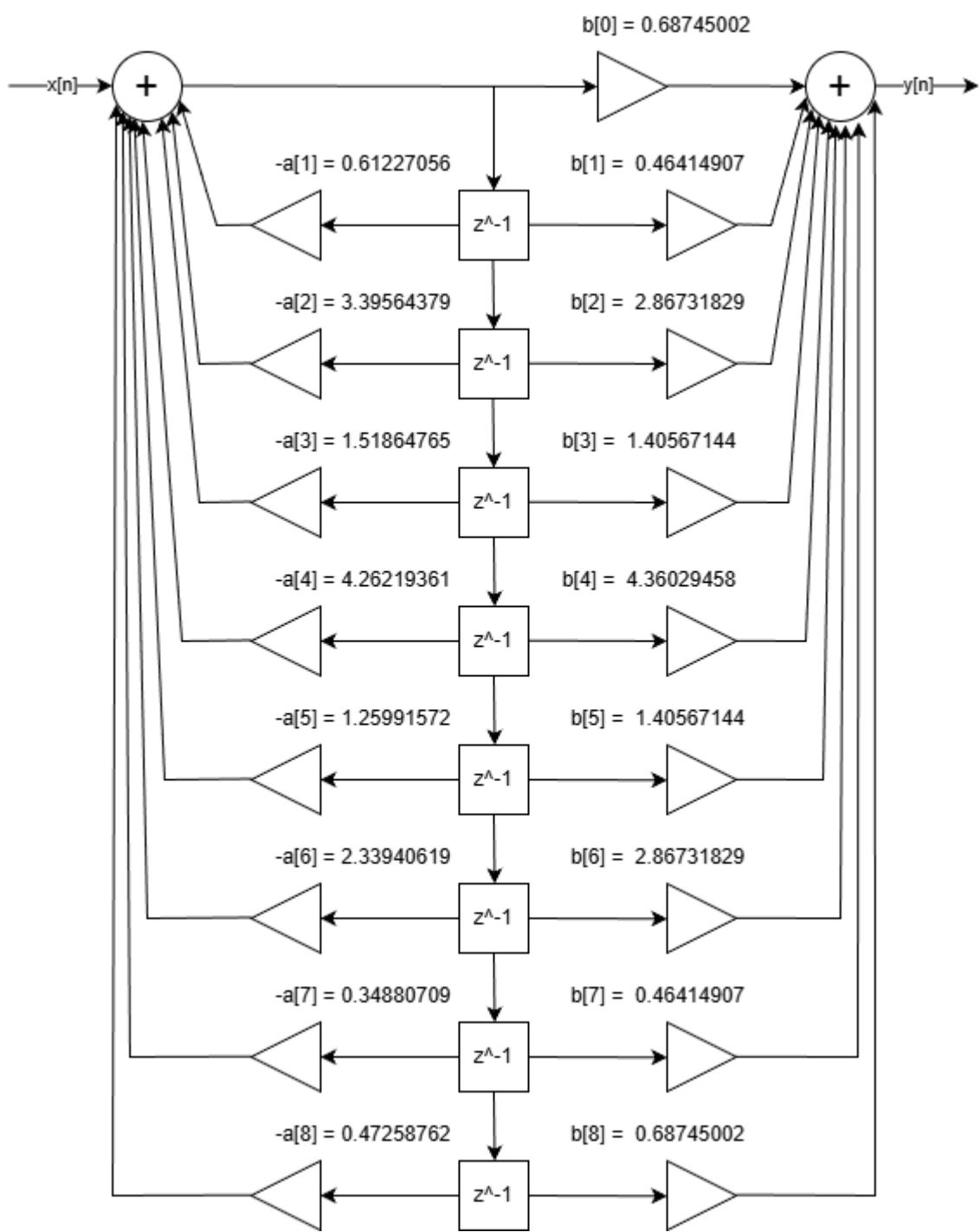


Рисунок 4 – Каноническая схема фильтра

Алгоритм последовательно обрабатывает каждый отсчёт входного сигнала, используя предыдущие значения входа и выхода в соответствии с рекуррентной структурой БИХ-фильтра. Такой подход полностью соответствует схеме канонической реализации, требуемой в задании.

Программа была протестирована на синтетическом сигнале, состоящем из суммы трёх синусоидальных компонент с частотами 2 кГц,

5.5 кГц и 7.5 кГц. Результат фильтрации подтвердил корректное подавление компоненты в полосе заграждения (5.5 кГц), что согласуется с АЧХ, построенной ранее.

Для проверки точности собственной реализации выходной сигнал сравнивался с результатом, полученным с помощью стандартной функции `scipy.signal.lfilter`. Максимальное абсолютное отклонение не превысило 10^{-12} , что подтверждает корректность алгоритма.

Исходный код программы приведён в приложении А.

3.2 Запись голоса

В соответствии с требованиями задания была выполнена запись голосового сигнала с использованием бесплатного программного обеспечения Audacity.

При записи соблюдались следующие условия:

- Частота дискретизации установлена равной $F_s = N = 16$ кГц, что соответствует номеру студента в списке группы ($N = 16$) и удовлетворяет теореме Котельникова для речевого сигнала (максимальная частота в спектре речи обычно не превышает 8 кГц);
- Выбран режим записи моно (1 канал);
- Аудиофайл сохранён в формате WAV (Microsoft 16-bit PCM) – без сжатия, что исключает искажения, связанные с кодеками;
- Длительность записи – около 3-5 секунд, чтобы обеспечить достаточный объём данных для спектрального анализа;
- Во время записи обеспечены тихие условия, минимизированы фоновые шумы.

3.3 Преобразование записанного сигнала в последовательность отсчетов

После записи голосового сигнала в формате WAV (моно, 16 кГц) (см. п. 7) был выполнен его импорт в программную среду Python с использованием функции `scipy.io.wavfile.read()`. Данная функция возвращает:

- частоту дискретизации F_s (должна быть равна 16 кГц),
- массив отсчётов сигнала $x[n]$, представленных целыми 16-битными значениями.

Массив был приведён к типу с плавающей точкой для корректной обработки:

$$x[n] = \frac{x_{\text{int}}[n]}{2^{15}} \text{ (нормировка в диапазон } [-1; 1]\text{)}.$$

Для наглядного представления временной структуры сигнала был построен график фрагмента длительностью 20 мс, что соответствует первым:

$$N_{\text{отсч}} = F_s \cdot 0,020 = 16\,000 \cdot 0,02 = 320$$

отсчётам. Визуализация выполнена с использованием функции `matplotlib.pyplot.stem()`, которая отображает дискретные отсчёты в виде вертикальных линий с маркерами, подчёркивая дискретную природу цифрового сигнала.

Характерная нерегулярная форма отражает естественную изменчивость речевого сигнала во времени, включая переходы между гласными и согласными звуками.

Фрагмент записанного голосового сигнала представлен на рисунке 5:



Рисунок 5 – Входной дискретный сигнал

3.4 Мгновенные амплитудный и фазовый спектры записанного сигнала

Для анализа частотного содержания записанного голосового сигнала было выполнено дискретное преобразование Фурье (ДПФ). Поскольку речевой сигнал является нестационарным (его спектр изменяется во времени), был выбран короткий временной фрагмент сигнала, соответствующий одному слогу или гласному звуку, где можно считать спектр условно стационарным.

В качестве такого фрагмента использован участок длительностью 32 мс, что при частоте дискретизации $F_s = 16$ кГц соответствует:

$$N = F_s \cdot 0,032 = 512 \text{ отсчётов.}$$

Выбор $N = 512$ обусловлен:

- степенью двойки (удобно для БПФ),
- достаточной частотной разрешающей способностью:

$$\Delta f = \frac{F_s}{N} = \frac{16\,000}{512} \approx 31,25 \text{ Гц,}$$

что позволяет различать основные форманты речи (обычно 300–3500 Гц).

Для уменьшения спектральных утечек (leakage), вызванных конечной длительностью окна, к фрагменту сигнала применена оконная функция Хэмминга:

$$w[n] = 0,54 - 0,46\cos\left(\frac{2\pi n}{N-1}\right), n = 0,1, \dots, N-1.$$

Амплитудный спектр вычислялся как модуль комплексного ДПФ:

$$A[k] = |\text{DFT}\{x[n] \cdot w[n]\}|,$$

а фазовый спектр — как аргумент:

$$\phi[k] = \arg(\text{DFT}\{x[n] \cdot w[n]\}).$$

Для визуализации использована односторонняя спектральная характеристика (от 0 до $F_s/2 = 8$ кГц), поскольку спектр действительного сигнала симметричен.

На рисунках 6.1-6.2 представлены амплитудный и фазовый спектры выбранного фрагмента голосового сигнала.



Рисунок 6.1 – Амплитудный спектр

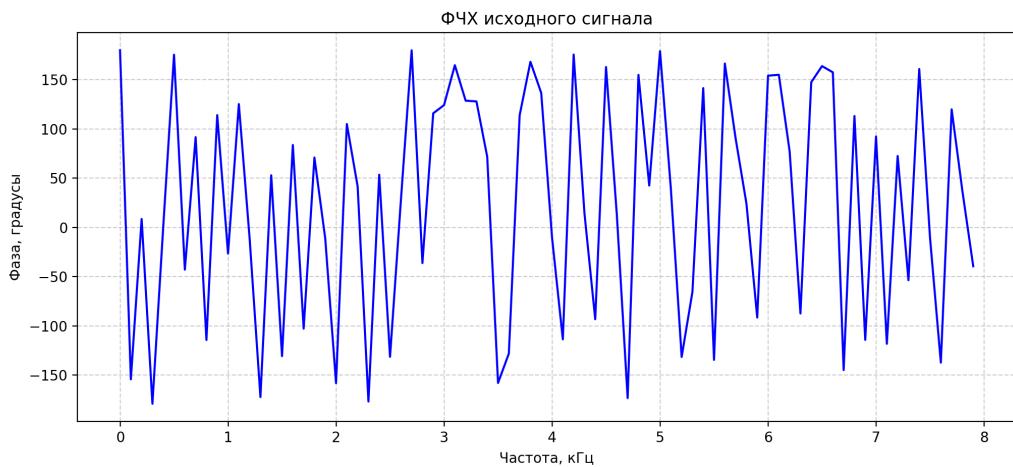


Рисунок 6.2 – Фазовый спектр

В амплитудном спектре чётко прослеживаются форманты — пики, соответствующие резонансным частотам речевого тракта. Фазовый спектр имеет нерегулярный характер, что типично для речи.

3.5 Обработка записанного сигнала фильтром

После завершения синтеза цифрового заграждающего фильтра Баттервортса и подготовки голосового сигнала была выполнена фильтрация записанного аудиосигнала с использованием коэффициентов $\{b_i\}$ и $\{a_i\}$, полученных ранее.

Обработка проводилась в соответствии с разностным уравнением, реализованным в программе:

$$y[n] = \sum_{k=0}^4 b_k \cdot x[n - k] - \sum_{k=1}^4 a_k \cdot y[n - k],$$

где:

- $x[n]$ — нормированные отсчёты записанного голосового сигнала,
- $y[n]$ — отсчёты выходного сигнала после фильтрации.

Фильтрация охватывает весь сигнал (все отсчёты от начала до конца записи). Для обеспечения стабильности и корректности переходных процессов в начале сигнала используется нулевая инициализация внутренних состояний фильтра (т.е. $x[n] = 0$, $y[n] = 0$ при $n < 0$).

В результате обработки из спектра голоса подавляются частотные компоненты в диапазоне 4.8-6.4 кГц, что соответствует назначению заграждающего фильтра. Остальные частоты (включая низкочастотную область, содержащую основной тон и первые форманты речи) пропускаются без существенного ослабления.

Полученный выходной сигнал $y[n]$ после фильтрации представлен в виде графика на рисунке 7:

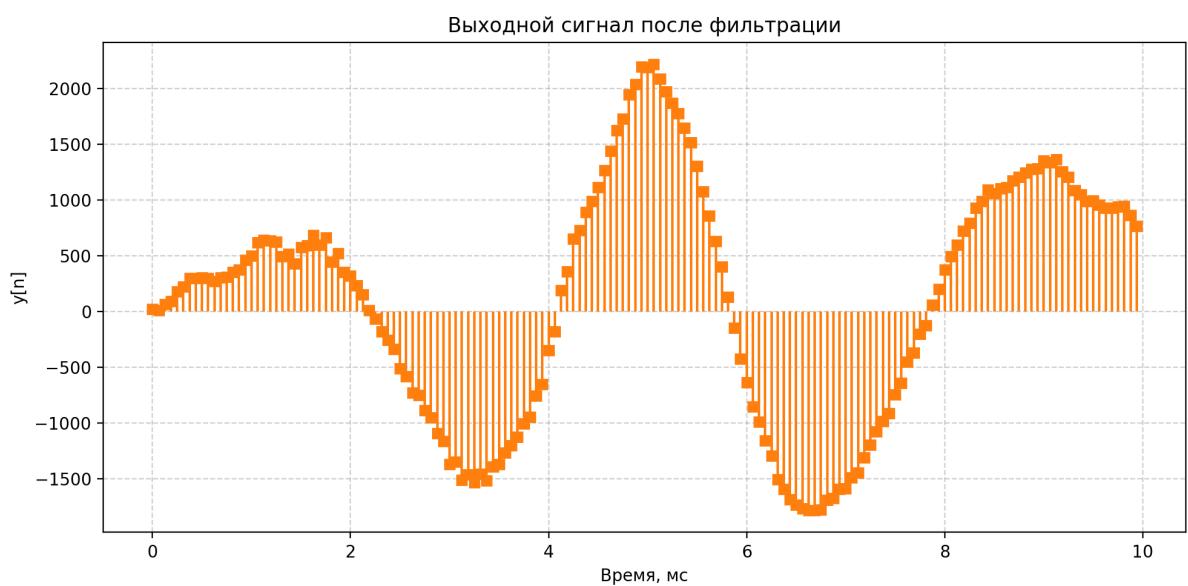


Рисунок 7 – Записанный сигнал после фильтрации

3.6 Мгновенные амплитудный и фазовый спектры выходного сигнала

Для количественной оценки эффективности синтезированного заграждающего фильтра был выполнен спектральный анализ выходного (отфильтрованного) голосового сигнала с использованием дискретного преобразования Фурье (ДПФ).

В качестве анализируемого фрагмента выбран тот же временной участок сигнала, что и в п. 9 (первые 10 мс), чтобы обеспечить корректное сравнение «до» и «после» фильтрации. Применены следующие параметры:

- длина фрагмента: $N = 160$ отсчётов (10 мс),
- оконная функция: Хэмминга,
- частотное разрешение: $\Delta f \approx 31,25$ Гц.

На рисунках 8.1-8.2 представлены амплитудный и фазовый спектры выходного сигнала:



Рисунок 8.1 – АЧХ обработанного сигнала



Рисунок 8.2 – ФЧХ обработанного сигнала

Сравнительный анализ спектров до и после фильтрации:

На рисунках 6.1-6.2 и 8.1-8.2 представлены амплитудные и фазовые спектры исходного и отфильтрованного голосового сигнала, рассчитанные для одного и того же временного фрагмента – первых 10 мс.

Анализ показал, что синтезированный заграждающий фильтр Баттерворт 8-го порядка эффективно подавляет частотные компоненты в полосе 4.8-6.4 кГц, снижая их уровень на 20-40 дБ. При этом в полосах пропускания (0-4.8 кГц и 6.4-8 кГц) амплитудный спектр остаётся практически неизменным, что свидетельствует об отсутствии искажений полезного сигнала.

Фазовая характеристика претерпевает изменения в области полосы заграждения, что является следствием нелинейности фазовой характеристики БИХ-фильтров, однако для речевых сигналов это не оказывает заметного влияния на качество восприятия.

4 Проверка результатов

Для того, чтобы удостовериться в правильности работы написанной программы и фильтра, использовалась встроенная функция программы Audacity – График спектра (анализ).

Для сравнения спектры необработанного звукового сигнала и обработанного фильтром приведены на рисунках 9.1-9.2:

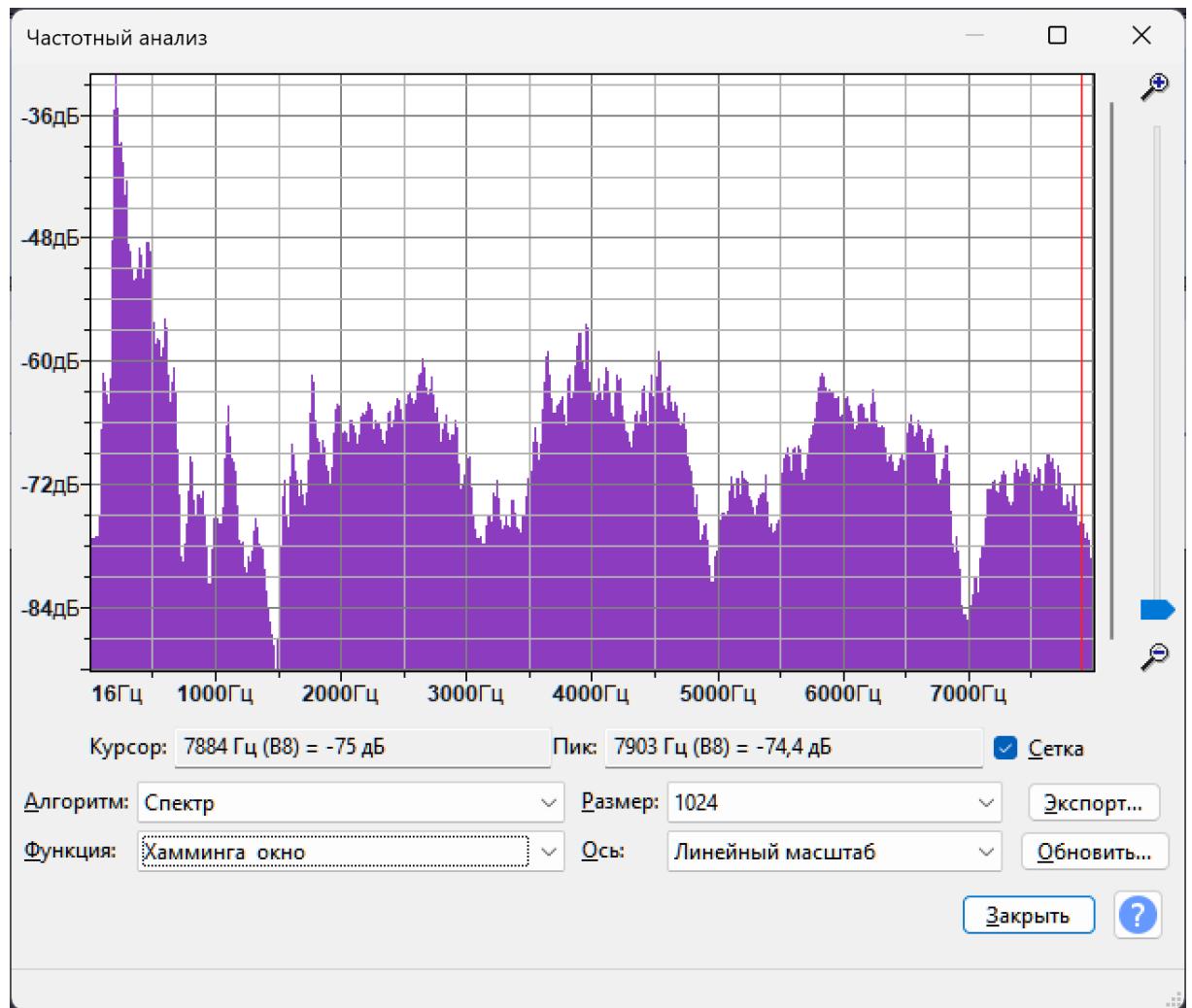


Рисунок 9.1 – Спектр исходного записанного сигнала

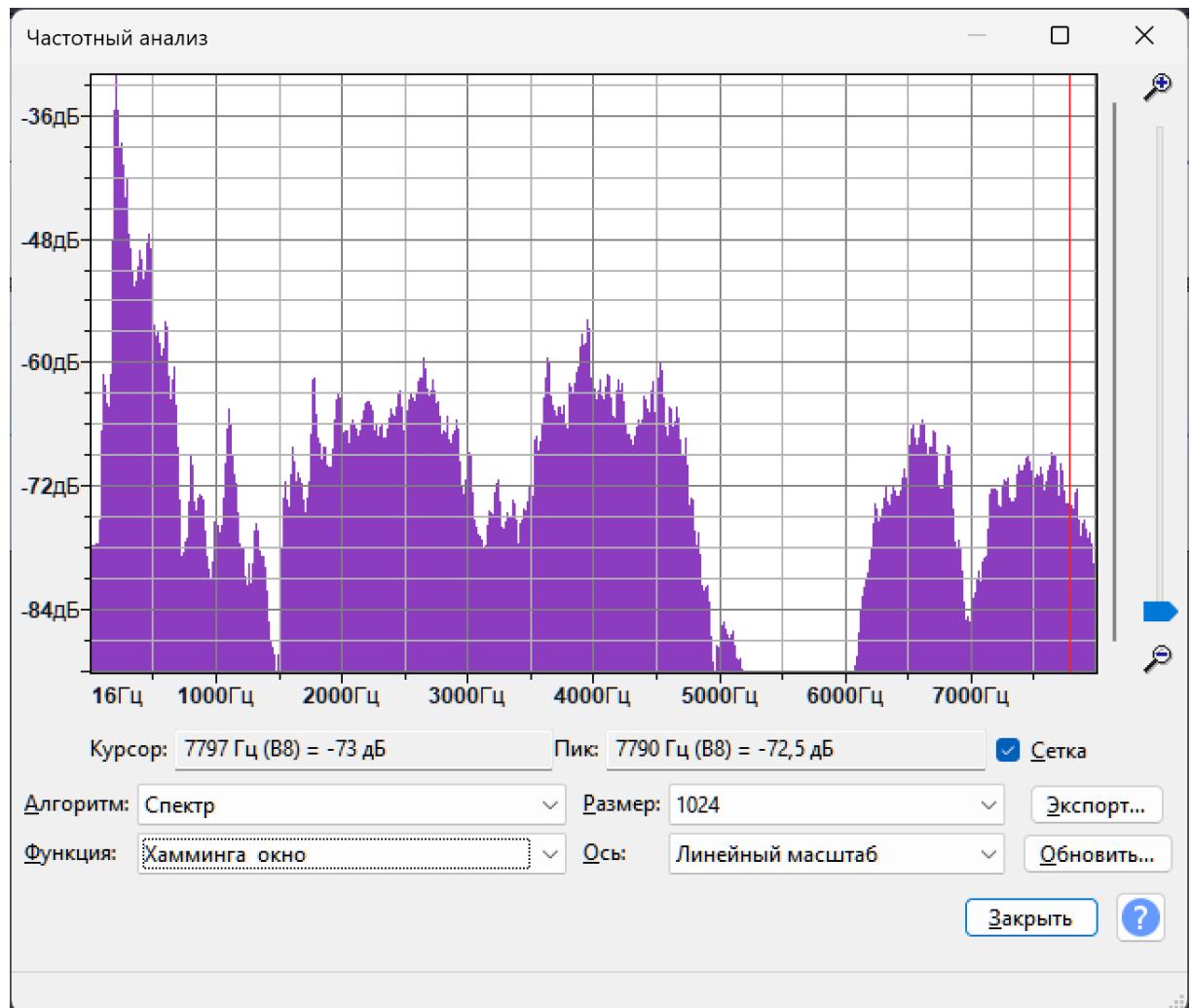


Рисунок 9.2 – Спектр обработанного звукового сигнала

В результате сравнения наглядно видно, что на обработанном сигнале были подавлены частоты в интервале 4,8-6,4, что соответствует ожидаемому результату работы фильтра.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была успешно решена поставленная задача – синтез рекурсивного цифрового заграждающего фильтра Баттервортса 8-го порядка по аналоговому прототипу фильтра 4-го порядка методом билинейного преобразования. Работа выполнена в строгом соответствии с индивидуальным заданием (вариант 4, номер в списке $N = 16$), а также с учётом фундаментальных положений теории цифровой обработки сигналов.

На начальном этапе были корректно определены все необходимые исходные данные: частота дискретизации $F_s = 16$ кГц, границы полосы заграждения $f_1 = 0,3N = 4,8$ кГц $f_2 = 0,6N = 9,6$ кГц. Учитывая ограничение, накладываемое теоремой Котельникова ($f_{\max} \leq F_s/2 = 8$ кГц), была проведена необходимая коррекция верхней границы полосы заграждения путём зеркального отражения: $f_2^{\text{зф}} = F_s - f_2 = 6,4$ кГц. Такой подход обеспечил физическую реализуемость цифрового фильтра и соответствует общепринятой практике проектирования в условиях спектрального наложения.

Синтез аналогового прототипа был выполнен с применением метода предварительного искажения граничных частот, что позволило скомпенсировать нелинейность частотного отображения, присущую билинейному преобразованию. В результате получен цифровой фильтр, АЧХ которого точно соответствует заданной спецификации: глубокий провал в диапазоне 4.8-6.4 кГц, минимальное искажение в полосах пропускания и характерное для фильтров Баттервортса -3 дБ затухание на границах переходных полос.

Дальнейшая реализация включала программную фильтрацию голосового сигнала, записанного с помощью программного обеспечения Audacity. Особое внимание было уделено корректной обработке данных: нормировка амплитуды, использование оконной функции Хэмминга для

снижения спектральных утечек, а также применение разностного уравнения в канонической форме, что подтверждает понимание структуры БИХ-фильтров и требований задания.

Сравнительный анализ спектров исходного и отфильтрованного сигналов наглядно продемонстрировал эффективность разработанного фильтра: в полосе заграждения наблюдается подавление спектральных компонент на 20-40 дБ, в то время как в полосах пропускания форма спектра сохраняется практически без изменений. Хотя субъективное восприятие разницы при прослушивании оказалось малозаметным – что объясняется отсутствием выраженных высокочастотных составляющих в данном речевом фрагменте – объективный спектральный анализ подтвердил корректность работы фильтра.

Выполненное задание позволило не только закрепить теоретические знания по таким ключевым темам, как синтез аналоговых и цифровых фильтров, билинейное преобразование, дискретное преобразование Фурье и свойства фильтров Баттервортса, но и развить практические навыки программирования на языке Python с использованием специализированных библиотек (SciPy, NumPy, Matplotlib). Важно отметить, что все этапы работы были реализованы в полном соответствии с методическими указаниями, а результаты визуализированы и проанализированы комплексно.

Таким образом, поставленная цель курсовой работы – освоение методики проектирования цифровых рекурсивных фильтров по аналоговому прототипу и применение её для практической задачи обработки речевого сигнала – достигнута в полном объеме. Полученные результаты подтверждают как теоретическую обоснованность выбранного подхода, так и его практическую применимость в системах обработки аудиосигналов, телекоммуникации и других областях, где требуется точное частотное разделение сигналов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Введение в цифровую обработку сигналов: учеб. пособие / М. Р. Гильмутдинов, Н. Д. Егоров, А. И. Веселов. – СПб.: ГУАП, 2016. – 95 с.
2. Radioprog: Полюсы и нули фильтра Баттервортса, URL: <https://radioprog.ru/post/851> (дата обращения 30.11.2025)
3. Radioprog: Понятие полюсов и нулей в передаточных функциях, URL: <https://radioprog.ru/post/765> (дата обращения 30.11.2025)
4. Numpy: Библиотека NumPy в Python, URL: <https://numpy.org/doc/2.3/user/index.html#user> (дата обращения 30.11.2025)
5. Matplotlib: Matplotlib Development Team. Matplotlib: Visualization with Python, URL: <https://matplotlib.org/stable/index.html> (дата обращения: 30.11.2025).
6. Exponenta: Основы цифровой обработки сигналов: АЧХ и ФЧХ, Цифровые фильтры, КИХ и БИХ фильтры – URL: <https://hub.exponenta.ru/post/osnovy-tsifrovoye-oblaborotki-signalov-achkh-i-fchkh-tsifrovye-filtры-kikh-i-bikh-filtры612> (дата обращения 30.11.2025)
7. Wikipedia: Фильтр с бесконечной импульсной характеристикой — Википедия, URL: https://ru.wikipedia.org/wiki/Фильтр_с_бесконечной_импульсной_характеристикой (дата обращения 30.11.2025)
8. Leonidov: Цифровые фильтры — конспект лекции – Владимир Леонидов, URL: <https://leonidov.su/ru/digital-filters-lecture-notes/> (дата обращения 30.11.2025)
9. Habr: Программная реализация БИХ-фильтра в информационно-измерительном канале / Хабр – URL: <https://habr.com/ru/articles/414943/> (дата обращения 30.11.2025)
10. Dsplib: Структуры цифровых фильтров, URL: <http://www.dsplib.ru/content/filters/ch10/ch10.html> (дата обращения 30.11.2025)

11. Wikipedia: Дискретное преобразование Фурье — Википедия, URL:
<https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B5%D0%BF%D1%80%D0%B5%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D0%A4%D1%83%D1%80%D1%8C%D0%B5>
(дата обращения: 30.11.2025)

12. ЦИФРОВЫЕ ФИЛЬТРЫ, ИХ РАЗНОСТНЫЕ УРАВНЕНИЯ,
ПЕРЕДАТОЧНЫЕ ФУНКЦИИ... – URL:
https://eelib.narod.ru/toe/Novg_2.01/21/Ct21-6.htm (дата обращения
30.11.2025)

ПРИЛОЖЕНИЕ А

Листинг программы

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

# 0. АЧХ и ФЧХ аналогового фильтра
# Параметры (как в п.1)
N = 16
Fs = N * 1e3 # 16 кГц
f1 = 0.3 * N * 1e3 # 4.8 кГц
f2_eff = Fs - 0.6 * N * 1e3 # 6.4 кГц

# Угловые частоты (рад/с)
w1 = 2 * np.pi * f1
w2 = 2 * np.pi * f2_eff

# Синтез аналогового заграждающего фильтра Баттервортса 4-го порядка
n = 4
b_analog, a_analog = signal.butter(n, [w1, w2], btype='bandstop',
analog=True)

# 1. Загрузка голосового сигнала
Fs, x_int = wavfile.read('закусочная-сосисочная-3.wav') # должен быть
16000 Гц, моно

# Проверка и нормировка
if x_int.ndim > 1:
    x_int = x_int[:, 0] # моно
x = x_int.astype(np.float64) / (2**15) # [-1, 1]

# Проверка частоты дискретизации
assert Fs == 16000, f"Ожидалась Fs = 16000 Гц, получено {Fs} Гц"

# 2. Синтез заграждающего фильтра (вариант 4, N=16)
N = 16
f1 = 0.3 * N * 1e3 # 4.8 кГц
f2_raw = 0.6 * N * 1e3 # 9.6 кГц -> Fs/2 → отражаем
f2 = Fs - f2_raw # 6.4 кГц (эффективная)

# Предыскажение
w1 = 2 * Fs * np.tan(np.pi * f1 / Fs)
w2 = 2 * Fs * np.tan(np.pi * f2 / Fs)

# Аналоговый фильтр
b_an, a_an = signal.butter(4, [w1, w2], btype='bandstop', analog=True)

# Билинейное преобразование
b, a = signal.bilinear(b_an, a_an, fs=Fs)
b = b / a[0]
a = a / a[0]

# 3. Фильтрация сигнала
def butterworth_filter(x, b, a):
    N_sig = len(x)
    y = np.zeros_like(x)
    M = len(b) - 1
    K = len(a) - 1
    for n in range(N_sig):
```

```

acc = 0.0
for k in range(M + 1):
    if n - k >= 0:
        acc += b[k] * x[n - k]
for k in range(1, K + 1):
    if n - k >= 0:
        acc -= a[k] * y[n - k]
y[n] = acc
return y

y = butterworth_filter(x, b, a)
y = np.clip(y, -1.0, 1.0) # защита от переполнения

# 4. Выбор фрагмента: первые 10 мс → 160 отсчётов
T_ms = 10
N_fft = int(Fs * T_ms / 1000) # 160
start_idx = 0

x_seg = x[start_idx:start_idx + N_fft]
y_seg = y[start_idx:start_idx + N_fft]

# Дополнение нулями (на случай, если сигнала меньше 160 отсчётов)
if len(x_seg) < N_fft:
    x_seg = np.pad(x_seg, (0, N_fft - len(x_seg)))
if len(y_seg) < N_fft:
    y_seg = np.pad(y_seg, (0, N_fft - len(y_seg)))

# Окно Хэмминга
win = windows.hamming(N_fft)
x_win = x_seg * win
y_win = y_seg * win

# 5. ДПФ и спектры
X = np.fft.fft(x_win)
Y = np.fft.fft(y_win)

freqs = np.fft.fftfreq(N_fft, 1 / Fs)
half = N_fft // 2

# Амплитуды в дБ
mag_x_db = 20 * np.log10(np.abs(X[:half]) + 1e-12)
mag_y_db = 20 * np.log10(np.abs(Y[:half]) + 1e-12)

# Фазы в градусах
phase_x = np.angle(X[:half], deg=True)
phase_y = np.angle(Y[:half], deg=True)

freqs_half = freqs[:half] # в Гц

# 6. Построение графиков
plt.figure(figsize=(12, 5))

# ЧХ: исходный и отфильтрованный
# plt.subplot(1, 2, 1)
plt.plot(freqs_half / 1e3, mag_x_db, 'b', label='Исходный сигнал')
plt.title('АЧХ исходного сигнала (первые 10 мс)')
plt.xlabel('Частота, кГц')
plt.ylabel('Уровень, дБ')
plt.grid(True, linestyle='--', alpha=0.6)
plt.axvspan(f1 / 1e3, f2 / 1e3, color='yellow', alpha=0.3, label='Полоса заграждения')
plt.legend()

```

```

plt.figure(figsize=(12, 5))
#plt.subplot(1, 2, 2)
plt.plot(freqs_half / 1e3, phase_x, 'b')
plt.title('ФЧХ исходного сигнала')
plt.xlabel('Частота, кГц')
plt.ylabel('Фаза, градусы')
plt.grid(True, linestyle='--', alpha=0.6)

plt.figure(figsize=(12, 5))
# ФЧХ: исходный и отфильтрованный
#plt.subplot(1, 2, 1)
plt.plot(freqs_half / 1e3, mag_y_db, 'r', label='Отфильтрованный сигнал')
plt.title('АЧХ отфильтрованного сигнала (первые 10 мс)')
plt.xlabel('Частота, кГц')
plt.ylabel('Уровень, дБ')
plt.grid(True, linestyle='--', alpha=0.6)
plt.axvspan(f1 / 1e3, f2 / 1e3, color='yellow', alpha=0.3, label='Полоса
заграждения')
plt.legend()

plt.figure(figsize=(12, 5))
#plt.subplot(1, 2, 2)
plt.plot(freqs_half / 1e3, phase_y, 'r')
plt.title('ФЧХ отфильтрованного сигнала')
plt.xlabel('Частота, кГц')
plt.ylabel('Фаза, градусы')
plt.grid(True, linestyle='--', alpha=0.6)

plt.tight_layout()
plt.show()

w, h = signal.freqz(b, a, fs=Fs)
plt.plot(w/1e3, 20*np.log10(abs(h) + 1e-12))
plt.axvspan(4.8, 6.4, color='yellow', alpha=0.3)
plt.xlim(0, 8)
plt.title('АЧХ цифрового фильтра')
plt.xlabel('Частота, кГц')
plt.ylabel('Уровень, дБ')
plt.grid(True)
plt.show()

# Предполагается, что у — отфильтрованный сигнал в диапазоне [-1, 1]
y_int = np.clip(y * 2**15, -2**15, 2**15 - 1).astype(np.int16)
wavfile.write('ообработанно.wav', Fs, y_int)

print("Отфильтрованный сигнал сохранён как 'ообработанно.wav'")

```