

Ганжур Марина Александровна – Донской государственный технический университет; e-mail: mganzhur@yandex.ru; г. Ростов-на-Дону, Россия; тел.: 89081819111; кафедра вычислительных систем и информационной безопасности; старший преподаватель.

Лебедев Борис Константинович – Южный федеральный университет; e-mail: lebedev.b.k@gmail.com; г. Таганрог, Россия; тел.: 89282897933; кафедра систем автоматизированного проектирования им. В.М. Курейчика; д.т.н.; профессор.

Лебедев Олег Борисович – МИРЭА – Российский технологический университет; e-mail: lebedev.ob@mail.ru; г. Москва, Россия; тел.: 89085135512; кафедра информатики; д.т.н.; профессор.

Ganzhur Marina Aleksandrovna – Don State Technical University; e-mail: mganzhur@yandex.ru; Rostov-on-Don, Russia; phone: +79081819111; the Department of Computing Systems and Information; security senior lecturer.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; Taganrog, Russia; phone: +79282897933; the Department of Computer Aided Design named after V.M. Kureichik; dr. of eng. sc.; professor.

Lebedev Oleg Borisovich – MIREA – Russian University of Technology; e-mail: lebedev.ob@mail.ru; Moscow, Russia; phone: +79085135512; the Department of Computer Science; dr. of eng. sc.; professor.

УДК 621.004.42

DOI 10.18522/2311-3103-2025-4-37-46

И.Е. Монсеенко, С.П. Тарасов, И.И. Турулин

АНАЛИЗ УСТОЙЧИВОСТИ И ОСОБЕННОСТЕЙ ПРАКТИЧЕСКОЙ РЕАЛИЗАЦИИ ФИЛЬТРОВ ХОГЕНАУЭРА КАК РЕКУРСИВНЫХ ЦИФРОВЫХ ФИЛЬТРОВ С КОНЕЧНОЙ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКОЙ

Рассматриваются вопросы устойчивости каскадных интегратор-гребенчатых (СИС – cascade integrator-comb) фильтров, используемых в цифровой обработке сигналов, в том числе для децимации и интерполяции. Проведен краткий обзор современных публикаций, касающихся архитектурной оптимизации СИС-фильтров. Основное внимание уделено повышению устойчивости фильтров к переполнению разрядной сетки, анализу их устойчивости и методу синтеза рекурсивных КИХ-фильтров (фильтров с конечной импульсной характеристикой). Для лучшего понимания природы устойчивости СИС-фильтров в работе приведены математические выкладки, иллюстрирующие особенности накопления постоянной составляющей при различных конфигурациях блоков. Предложено изменение структуры СИС-фильтра, заключающееся в перестановке блоков интегратора и гребенчатого фильтра. Доказано, что такое изменение предотвращает накопление постоянной составляющей сигнала в интеграторах и, следовательно, исключает переполнение разрядной сетки вследствие накопления постоянной составляющей в интеграторе. Этот подход базируется на свойстве линейных фильтров, согласно которому изменение порядка включения не влияет на передаточную функцию, амплитудно-частотную характеристику, но в случае цифровых реализаций позволяет существенно снизить вероятность переполнения. Возможности аппаратной и программной реализации таких структур рассматриваются с точки зрения минимизации потерь точности и увеличения надежности работы систем цифровой обработки сигналов. Предложено использование целых чисел или чисел с фиксированной точкой для устранения накопления ошибок квантования. Кроме того, была разработана программа на языке Python, реализующая СИС-фильтр с учетом устойчивости к постоянной составляющей во входном сигнале и точным выполнением операций. Полученные результаты сопоставлены с современными подходами, представленными в научных исследованиях последних лет. Предложенные решения могут быть полезны при разработке цифровых фильтров для систем с ограниченными вычислительными ресурсами и повышенными требованиями к стабильности.

СИС-фильтр; Хогенауэр; цифровая обработка сигналов; децимация; импульсная характеристика; амплитудно-частотная характеристика; фазочастотная характеристика; Python; устойчивость фильтров.

I.E. Moiseenko, S.P. Tarasov, I.I. Turulin

ANALYSIS OF STABILITY AND FEATURES OF PRACTICAL IMPLEMENTATION OF HOGENAUER FILTERS AS RECURSIVE DIGITAL FILTERS WITH FINITE IMPULSE RESPONSE

The article considers the issues of stability of cascade integrator-comb (CIC) filters used in digital signal processing, including decimation and interpolation. A brief review of modern publications on the architectural optimization of CIC filters is given. The main attention is paid to increasing the stability of filters to the overflow of the bit grid, analyzing their stability and the method of synthesis of recursive FIR filters (filters with a finite impulse response). For a better understanding of the nature of the stability of CIC filters, the paper presents mathematical calculations illustrating the features of the accumulation of the constant component for various block configurations. A change in the structure of the CIC filter is proposed, consisting in the permutation of the integrator and comb filter blocks. It is proved that such a change prevents the accumulation of the constant component of the signal in the integrators and, therefore, eliminates the overflow of the bit grid due to the accumulation of the constant component in the integrator. This approach is based on the property of linear filters, according to which changing the order of inclusion does not affect the transfer function, amplitude-frequency characteristic, but in the case of digital implementations it allows to significantly reduce the probability of overflow. The possibilities of hardware and software implementation of such structures are considered from the point of view of minimizing the loss of accuracy and increasing the reliability of digital signal processing systems. It is proposed to use integers or numbers with a fixed point to eliminate the accumulation of quantization errors. In addition, a program in Python was developed that implements a CIC filter taking into account the stability of the constant component in the input signal and the accurate execution of operations. The obtained results are compared with modern approaches presented in scientific research in recent years. The proposed solutions can be useful in developing digital filters for systems with limited computing resources and increased stability requirements.

CIC filter; Hogenauer; digital signal processing; decimation; impulse response; amplitude-frequency response; phase-frequency response; Python; filter stability.

Введение. СИС-фильтры (Cascaded Integrator–Comb фильтры) решают ряд важных задач в цифровой обработке сигналов, особенно связанных с изменением частоты дискретизации [1]. Рассмотрим основные проблемы, которые они помогают решать: децимация, минимизация вычислительной сложности, первичная фильтрация, простота аппаратной реализации.

СИС-фильтр – это комбинированная структура, состоящая из интегратора (БИХ-фильтр) и разностного КИХ-фильтра. Фильтр-интегратор – это элемент, который накапливает (интегрирует) входные значения сигнала [2].

Интегратор работает по рекуррентному принципу, где текущее значение выхода вычисляется как сумма предыдущего выхода и текущего входного значения [3]. Математически это можно записать как:

$$y[n] = y[n - 1] + x[n], \quad (1)$$

где $x[n]$ – входной сигнал, $y[n]$ – выходной сигнал интегратора.

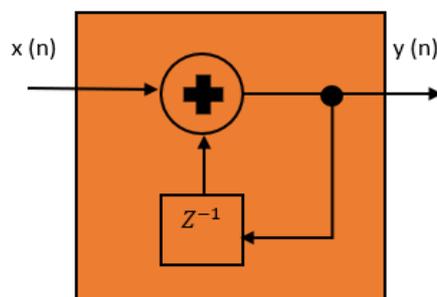


Рис. 1. Структурная схема интегратора

Гребенчатый (разностный) фильтр в составе СИС-фильтра обеспечивает формирование итоговой фильтрующей характеристики [4]. Гребенчатый фильтр реализует операцию вычитания из текущего значения входного сигнала его значение, полученное с некоторой задержкой [5]. Математически это можно записать как:

$$y[n] = x[n] + x[n - R], \quad (2)$$

где $x[n]$ – входной сигнал, R – параметр задержки, который часто совпадает с коэффициентом децимации [4].

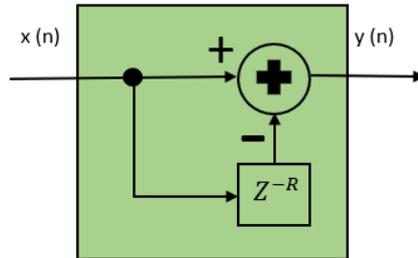


Рис. 2. Структурная схема гребенчатого фильтра

В СИС-фильтре несколько интеграторов последовательно аккумулируют входные данные, а затем соответствующее количество гребенчатых фильтров применяется для вычитания задержанных версий сигнала [6]. Такое чередование приводит к формированию общей низкочастотной характеристики с характеристическим sinc-подобным затуханием, что полезно для подавления высокочастотного шума и предотвращения алиасинга при децимации [7]. На рис. 3 изображена структурная схема простого СИС фильтра.

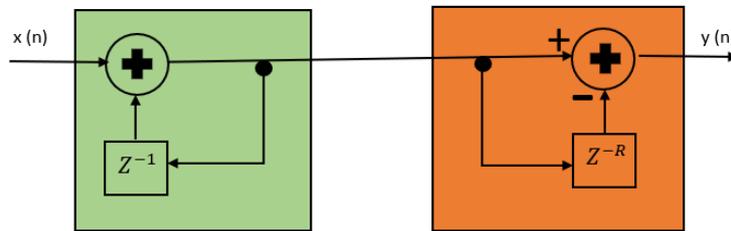


Рис. 3. СИС-фильтр первого порядка

При реализации СИС-фильтров важно учитывать разрядность вычислителя, поскольку от неё зависят точность, динамический диапазон и корректность работы фильтра [8].

В ряде современных работ также исследуются вопросы устойчивости и оптимизации СИС-фильтров. В частности, в [9] представлена оптимизация структуры СИС-фильтра для аппаратной реализации с учетом энергетических затрат. В [10] сравниваются подходы к усилению частотных характеристик FIR-фильтров с применением техник их уточнения (sharpening). Статья [11] уделяет внимание FPGA-реализации СИС-фильтров, а в [12] предложены способы адаптации фильтров к изменениям входного сигнала в реальном времени. Обзор современных подходов к квантованию и структурам КИХ-фильтров приведен в [13].

Улучшение устойчивости СИС-фильтра к переполнению. Для этого покажем, что гребенчатый фильтр S можно представить в виде каскадного соединения фильтра с прямоугольной КИХ:

$$hr[n] = 1 \text{ при } 0 \leq n \leq R-1, \text{ в остальных случаях } hr[n] = 0, \quad (3)$$

и дискретного дифференциатора [4], выполняющего операцию:

$$\forall \{h[n]\} = \forall h[n] = h[n] - h[n - 1]. \quad (4)$$

Подав на вход дискретного дифференциатора единичный импульс ($\delta[n] = 1$ при $n = 0$ и $\delta[n] = 0$ при других n), и выполнив численное дифференцирование по формуле (4), получим КИХ дифференциатора

$$\nabla\{h[n]\} = \nabla h[n] = h[n] - h[n-1], h_{\nabla}[n] = 1, \quad (5)$$

если $n = 0$; $h_{\nabla}[n] = -1$,

если $n = 1$; $h_{\nabla}[n] = 0$ – в других случаях.

Выполнив численное дифференцирование формулы (3) по формуле (1), получим КИХ каскадного соединения КИХ (3) и дискретного дифференциатора

$$\begin{aligned} h_C[n] &= 1 \text{ при } n = 0, \\ h_C[n] &= -1 \text{ при } n = R, \\ h_C[n] &= 0 \text{ – в других случаях.} \end{aligned} \quad (6)$$

Как видно, формула (6) есть КИХ гребенчатого фильтра С, т.е. фильтр С содержит дискретный дифференциатор.

Если на вход фильтра (рис. 4) подать сигнал с постоянной составляющей, то первый (входной) интегратор будет ее накапливать, содержимое его сумматора будет расти, и в конце концов произойдет переполнение разрядной сетки [14]. На выходе первого гребенчатого фильтра С также будет постоянная составляющая, которая будет аналогично интегрироваться следующим интегратором с теми же последствиями в виде переполнение разрядной сетки.

Если поменять местами блоки I и С, т.е. блок С включить первым, то постоянная составляющая входного сигнала будет подавлена дискретным дифференциатором, входящим в блок С, и, при наличии во входном сигнале постоянной составляющей, переполнения не будет. Аналогично постоянной составляющей не будет на входе других интеграторов I в схеме на рис. 4.

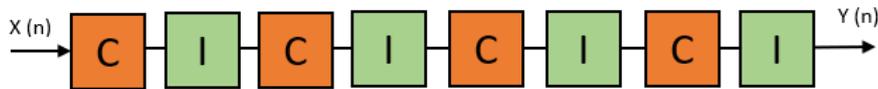


Рис. 4. Обеспечение устойчивости СИС-фильтра при округлениях результата вычислений

Заметим, что блоки в схеме на рис. 4 линейны, а, как известно, порядок каскадного включения линейных фильтров не существен [4]. Однако в цифровых фильтрах присутствует источник нелинейности в виде ограниченной длины разрядной сетки, которая проявляется при ее превышении [15]. Если превышения не было, фильтр линеен. Поэтому с учетом практической реализации порядок включения линейных фильтров может быть существенен, как в данном случае.

В источнике [16] есть предположение, что подобная расстановка фильтров в системе может привести к ошибкам при округлении результатов вычислений и большом количестве отсчетов.

Для решения этой проблемы была предложена обратная расстановка гребенчатых фильтров и интеграторов.

Анализ устойчивости. Для анализа устойчивости выполним некоторые выкладки. Выполнив z-преобразование формулы (4), получим системную функцию дифференциатора:

$$H_{\nabla}(z) = 1 - z^{-1}. \quad (7)$$

Операция дискретного интегрирования производит суммирование всех предыдущих отсчетов входного сигнала $x[n]$, т. е. выходной сигнал интегратора:

$$y[n] = \sum_{k=-\infty}^n x[k]. \quad (8)$$

Линейное разностное уравнение интегратора I:

$$y[n] = x[n] + y[n-1]. \quad (9)$$

Выполнив z -преобразование формулы (8), получим его системную функцию:

$$H_{\Sigma}(z) = 1/(1 - z^{-1}). \quad (10)$$

Сравнив формулы (7) и (10) можно убедиться, что операции дискретного дифференцирования и интегрирования взаимнообратны, т. е. $H_{\nabla}(z) \cdot H_{\Sigma}(z) = 1$.

Таким образом, фильтр с прямоугольной КИХ, представляющий собой каскадное соединение гребенчатого фильтра C и интегратора I , является рекурсивным по структуре и КИХ-фильтром по длине импульсной характеристики, с системной функцией

$$H_I(z) = (1 - z^{-R})/(1 - z^{-1}). \quad (11)$$

Если поделить числитель на знаменатель, получится функция

$$\sum_{k=0}^{R-1} z^{-k}, \quad (12)$$

соответствующая нерекурсивному фильтру с такой же прямоугольной КИХ.

Сам по себе интегратор является рекурсивным БИХ-фильтром, причем неустойчивым, поскольку его импульсная характеристика представляет собой единичную ступенчатую функцию, которые не затухает во времени, то есть фильтр не возвращается в исходное состояние после прекращения входного воздействия [17]. Если операции в фильтре выполняются точно, то преобразования формулы (11) в формулу (12) происходит точно. В противном случае в структуре как бы остается интегратор с малым коэффициентом передачи, порядка погрешности квантования (следствие неточного выполнения операций в фильтре), неустойчивость остаётся, хоть и сопровождается малой погрешностью, которая накапливается и циркулирует в интеграторе [18, 19].

Обычно даже в случае неточного выполнения операций такая погрешность проблем не вызывает, поскольку реальные системы чаще всего работают на ограниченных временных интервалах [20]. Однако для того, чтобы фильтр был строго устойчивым, необходимо точное выполнение операций в фильтре [21]. Это может быть достигнуто использованием целых чисел или чисел с фиксированной точкой.

В рекурсивных КИХ-фильтрах, описанных в [22, 23], обычно используется несколько каскадно включенных интеграторов, здесь необходимо точное выполнение операции в фильтре, поскольку постоянная погрешность на выходе первого интегратора интегрируется во втором, и на его выходе появляется линейно нарастающая функция, на выходе третьего интегратора – параболически нарастающая функция, и выходной сигнал такого фильтра быстро выходит за пределы разрядной сетки сумматора.

О методе синтеза рекурсивных КИХ-фильтров [22, 23]. Как видно из (3), КИХ фильтра Хогенауэра (3) задана в виде полинома нулевой степени аргумента n . В результате дискретного дифференцирования исходной прямоугольной КИХ по формуле (4) получим примитивный КИХ-фильтр – гребенчатый фильтр C с парой ненулевых коэффициентов (6), что резко снижает вычислительную сложность. Дискретный интегратор I , включенный каскадно с C (после него – для предотвращения накопления постоянной составляющей и переполнения) восстанавливает исходную КИХ.

Такой подход можно распространить на исходную КИХ любой степени p аргумента n [22, 23]. В этом случае КИХ дифференцируется по формуле (4) $p + 1$ раз, в результате чего получается примитивная КИХ с числом ненулевых коэффициентов не более $s \cdot 2(p + 1)$. Рекурсивный КИХ-фильтр реализуется подключением к выходу примитивного фильтра $p + 1$ интеграторов, восстанавливающих исходную КИХ. Вычислительная сложность полученной структуры при большой длине исходной КИХ существенно ниже, чем у исходного нерекурсивного КИХ-фильтра. Интеграторы подключаются обязательно к выходу примитивного фильтра – для предотвращения накопления постоянной составляющей и переполнения каждого интегратора. Операции при фильтрации в рекурсивном КИХ-фильтре должны выполняться точно (это достигается использованием целых чисел или чисел с фиксированной точкой), иначе фильтр становится неустойчивым.

Реализация СИС-фильтра на Python с учетом устойчивости к постоянной составляющей во входном сигнале и точным выполнением операций. В программе реализованы фильтры Хогенауэра разной каскадности и производится их сложение для получения фильтра с заданными свойствами.

После запуска программа предлагает ввести входные данные для синтеза фильтра. Это параметр N , означающий количество стоящих друг за другом фильтров и их каскадность. Допустим, при $N = 3$ реализуется сложение однокаскадного, двухкаскадного и трехкаскадного фильтра. Параметр R – коэффициент децимации. В нашем случае, как было описано выше, он также является параметром задержки. Параметры весовых коэффициентов для каждого фильтра регулируют степень участия каждого фильтра в итоговой характеристике.

После инициализации необходимых переменных программа производит расчет фильтра интегратора и гребенчатого фильтра.

Фильтр-интегратор реализуется посредством операции свёртки с вектором, состоящим из единиц [24]. В начале функции «*cic_filter_alternating*» задаётся начальная импульсная характеристика фильтра: $h = np.ones(R)$.

Здесь *np.ones(R)* создаёт массив длины R , заполненный единицами. Этот массив является импульсной характеристикой простейшего интегратора, который суммирует R последовательных входных значений (т.е. действует как скользящее суммирование или интегрирование за фиксированный промежуток).

В цикле *for i in range(N)*: происходит чередование интеграторов и гребенчатых (разностных) фильтров. Каждый раз, когда значение индекса « i » четное (*if i % 2 == 0*), программа применяет очередной интегратор: $h = np.convolve(h, np.ones(R))$

Здесь происходит свёртка текущей импульсной характеристики h с вектором *np.ones(R)*. Это эквивалентно тому, что к уже накопленным данным применяется операция суммирования по блоку длины R . Такая операция реализует дискретное интегрирование, поскольку каждый новый выходной отсчёт является суммой предыдущего выходного отклика и группы входных значений [25].

Гребенчатый фильтр также реализуется в функции *cic_filter_alternating*. Создается массив длины $R + 1$, заполненный нулями.

$$h_comb = np.zeros(R + 1).$$

Затем устанавливаются два значения. Первый элемент $h_comb[0]$ задается равным 1, а последний элемент $h_comb[-1]$ задается равным -1 . Это дает ядро $h_comb = [1, 0, 0, \dots, 0, -1]$.

Такое ядро реализует операцию разности (2), то есть вычитает значение входного сигнала, отстающее на R отсчетов.

После создания ядра гребенчатого фильтра, оно применяется к текущей импульсной характеристике h посредством операции свертки: $h = np.convolve(h, h_comb)$.

Эта свертка осуществляет дифференцирование, устраняя накопленные эффекты, созданные предыдущими этапами интеграции, и формируя итоговую импульсную характеристику фильтра.

В переменной сохраняется список отдельных импульсных характеристик СИС фильтров, которые генерируются для каждого этапа (каскада) согласно введённым пользователем параметрам. Выражение:

$$filters = [np.nan_to_num(cic_filter_alternating(i + 1, R[i])) \text{ for } i \text{ in range}(N)]$$

в виде спискового включения (list comprehension) проходит по всем индексам от 0 до $N - 1$ (где N – число фильтров, введенное пользователем). Для каждого « i »:

Вызывается функция *cic_filter_alternating(i + 1, R[i])*, где:

- ◆ $i + 1$ задаёт количество этапов (каскадов) для СИС фильтра на данном шаге. Таким образом, первый элемент соответствует СИС фильтру с одним этапом, второй – с двумя и так далее;
- ◆ $R[i]$ – коэффициент децимации, введённый пользователем для i -го фильтра.
- ◆ Функция возвращает массив (1D NumPy array), который является импульсной характеристикой этого СИС фильтра;
- ◆ Вызов *np.nan_to_num(...)* используется для замены возможных NaN (если они возникли в результате вычислений) на числовые значения, чтобы гарантировать корректность полученного массива.

После выполнения этого выражения, переменная *filters* становится списком, где:

- ◆ $filters[0]$ – импульсная характеристика СИС фильтра с 1 этапом (интегратор и гребенчатый фильтр) и коэффициентом децимации $R[0]$;

- ◆ $filters[1]$ – импульсная характеристика СИС фильтра с 2 этапами и коэффициентом децимации $R[1]$;
- ◆ И так далее до $filters[N - 1]$.

Впоследствии эти отдельные импульсные характеристики комбинируются с заданными весовыми коэффициентами для формирования итогового фильтра. Каждый фильтр из списка $filters$ приводится к общей длине, затем умножается на соответствующий вес, и суммируется с другими, чтобы получить итоговую импульсную характеристику комбинированного фильтра. На рис. 5 приведена блок-схема моей программы.

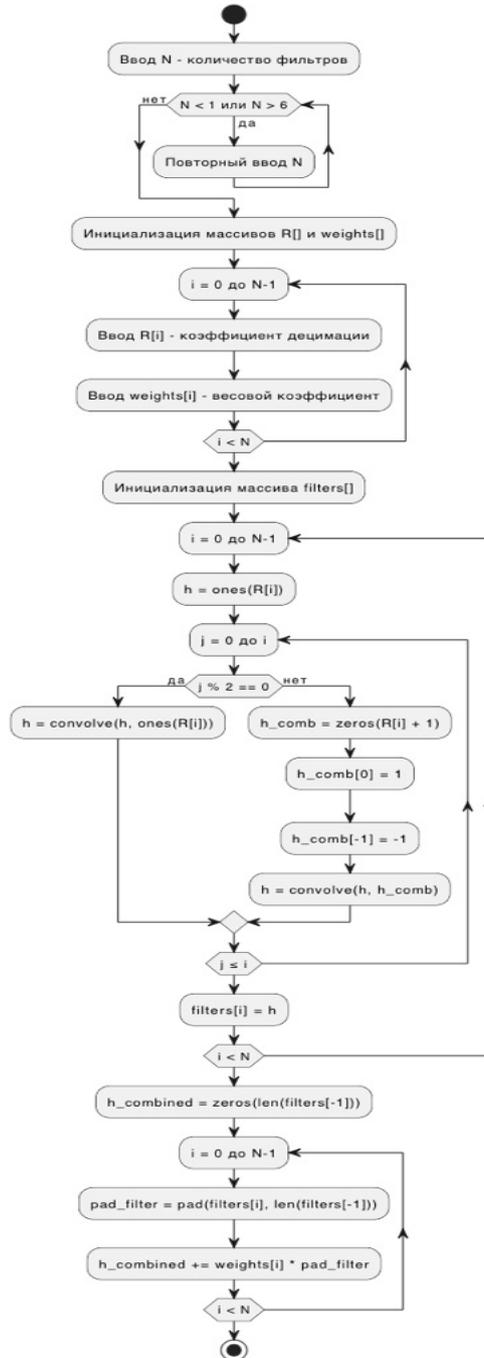


Рис. 5. Блок-схема Python программы

Пример работы программы для $N = 3, R = 12$, весовые коэффициенты 0.3, 0.4, 0.3 соответственно на рис. 6–8:

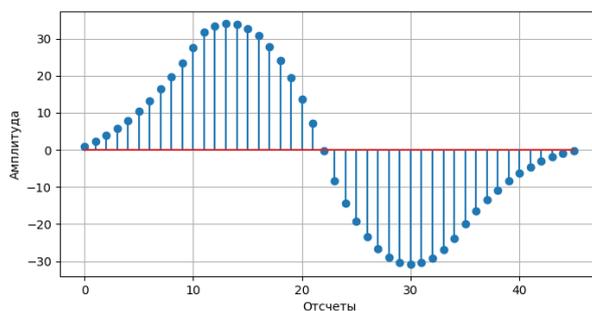


Рис. 6. Импульсная характеристика комбинированного СИС-фильтра

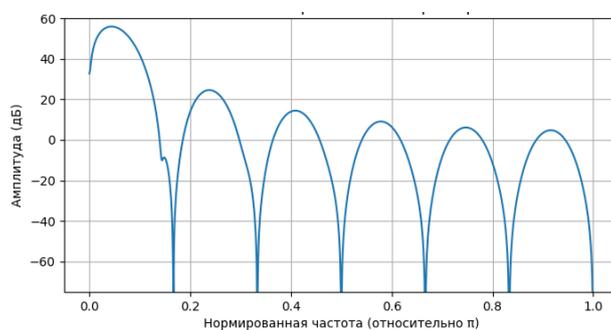


Рис. 7. АЧХ реализованного комбинированного СИС-фильтра

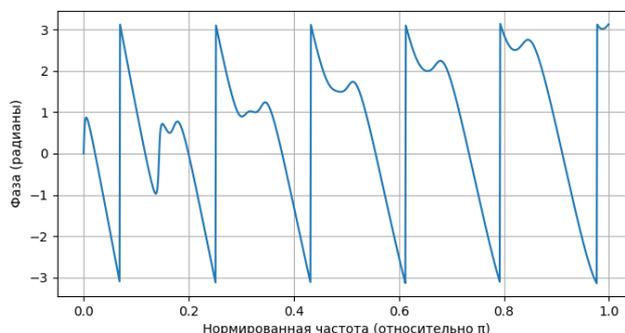


Рис. 8. ФЧХ реализованного комбинированного СИС-фильтра

Выводы. Проведенный анализ устойчивости показывает, для предотвращения накопления ошибок квантования операции в рекурсивных КИХ-фильтрах, в том числе фильтрах Хогенауэра, должны выполняться точно.

Применение метода синтеза рекурсивных КИХ-фильтров позволяет снизить вычислительную сложность фильтра, сохраняя его КИХ.

Перестановка блоков интегратора и гребенчатого фильтра позволяет исключить накопление постоянной составляющей и исключить ошибки переполнения.

Гибкость параметров и простота реализации подчеркивают практическую значимость СИС-фильтров в системах цифровой обработки сигналов, а анализ амплитудно-частотных и фазочастотных характеристик подтверждает корректность их построения.

Разработанная программа демонстрирует успешную реализацию СИС-фильтра, учитывающую устойчивость к постоянной составляющей входного сигнала вследствие точного выполнения операций.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Hogenauer E.B.* An Economical Class of Digital Filters for Decimation and Interpolation // *IEEE Transactions on Acoustics, Speech and Signal Processing*. – 1981. – No. 29. – P. 155-162.
2. *Oppenheim A.V., Schaffer R.W.* Discrete-Time Signal Processing. – Upper Saddle River, NJ: Prentice-Hall, 1989.
3. *Crochiere R.E., Rabiner L.R.* Multirate Digital Signal Processing. – Englewood Cliffs, NJ: Prentice-Hall, 1983.
4. *Lyons R.G.* Understanding Digital Signal Processing. – Upper Saddle River, NJ: Prentice-Hall, 2004.
5. *Proakis J.G., Manolakis D.G.* Digital Signal Processing: Principles, Algorithms, and Applications. – Upper Saddle River, NJ: Prentice-Hall, 1996.
6. *Vaidyanathan P.P.* Multirate Systems and Filter Banks. – Englewood Cliffs, NJ: Prentice-Hall, 1993.
7. *Ziemer R.E., Tranter W.H., Fannin D.R.* Signals and Systems: Continuous and Discrete. – Upper Saddle River, NJ: Prentice-Hall, 1998.
8. *Mitra S.K.* Digital Signal Processing: A Computer-Based Approach. – New York: McGraw-Hill, 2006.
9. *Wang J., Hu Y., Duan Y., Jin W.* Recursive Algorithm FIR Filter Quantization and Low-Cost Structure Optimization Design Based on MATLAB and Simulink // 2024 7th International Conference on Signal Processing and Information Communications (ICSPIC). – IEEE, 2024. – DOI: 10.1109/ICSPIC60374.2024.10387811.
10. *Ali I., Hasan M., Ahmed S.* Demonstration: Comparison of Three FIR Digital Filter Sharpening Techniques // 2020 IEEE International Symposium on Circuits and Systems (ISCAS). – 2020. – DOI: 10.1109/ISCAS45731.2020.9180956.
11. *Liu C., Li J., Zhou Y.* Efficient Design of Sparse FIR Digital Filter // 2021 IEEE International Symposium on Circuits and Systems (ISCAS). – 2021. – DOI: 10.1109/ISCAS51556.2021.9362902.
12. *Zhou M., Li Y., Wang X.* Adaptive Digital Filter Design Based on CIC Structure // 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA). – 2020. – DOI: 10.1109/ICIEA48937.2020.9084880.
13. *Tang L., Wei J., Chen R.* Design of FIR Filter Based on FPGA // 2020 IEEE 3rd International Conference on Electronics and Communication Engineering (ICECE). – 2020. – DOI: 10.1109/ICECE49637.2020.9163582.
14. *Antoniou A.* Digital Signal Processing: Signals, Systems, and Filters. – New York: McGraw-Hill, 2006.
15. *Smith S.W.* The Scientist and Engineer's Guide to Digital Signal Processing. – California: California Technical Publishing, 1997.
16. Hogenauer Digital Filters (CIC Filters). – URL: <http://www.dsplib.ru/content/cic/cic.html> (accessed: 06.02.2025).
17. *Пахомов А.И.* Дискретная фильтрация сигналов. – М.: Высшая школа, 1998.
18. *Oppenheim A.V.* The Evolution of Digital Signal Processing // *Proceedings of the IEEE*. – 2000. – No. 88. – P. 746-800.
19. *Rabiner L.R.* Digital Filters: An Introduction // *IEEE Spectrum*. – 1977. – No. 14. – P. 61-73.
20. *Берендеев П.В., Козлов С.А.* Цифровая обработка сигналов: теория и приложения. – М.: Наука, 2005. – С. 78-82.
21. *Harris F.J.* On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform // *Proceedings of the IEEE*. – 1978. – No. 66. – P. 51-83.
22. *Турулин И.И.* Метод проектирования рекурсивных фильтров общего вида с конечной импульсной характеристикой // *Известия вузов. Электроника*. – 2001. – № 6. – С. 97-98.
23. *Турулин И.И.* Основы теории рекурсивных КИХ-фильтров: монография. – Таганрог: Изд-во ЮФУ, 2016. – 264 с.
24. *Rabiner L.R., Gold B.* Theory and Application of Digital Signal Processing. – Englewood Cliffs, NJ: Prentice-Hall, 1975.
25. *Соловьев В.Н.* Цифровые фильтры: теория и практика. – СПб.: Политехническое изд-во, 2004. – С. 45-50.

REFERENCES

1. *Hogenauer E.B.* An Economical Class of Digital Filters for Decimation and Interpolation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1981, No. 29, pp. 155-162.
2. *Oppenheim A.V., Schaffer R.W.* Discrete-Time Signal Processing. Upper Saddle River, NJ: Prentice-Hall, 1989.
3. *Crochiere R.E., Rabiner L.R.* Multirate Digital Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1983.
4. *Lyons R.G.* Understanding Digital Signal Processing. Upper Saddle River, NJ: Prentice-Hall, 2004.

5. Proakis J.G., Manolakis D.G. Digital Signal Processing: Principles, Algorithms, and Applications. Upper Saddle River, NJ: Prentice–Hall, 1996.
6. Vaidyanathan P.P. Multirate Systems and Filter Banks. – Englewood Cliffs, NJ: Prentice–Hall, 1993.
7. Ziemer R.E., Tranter W.H., Fannin D.R. Signals and Systems: Continuous and Discrete. Upper Saddle River, NJ: Prentice–Hall, 1998.
8. Mitra S.K. Digital Signal Processing: A Computer-Based Approach. New York: McGraw–Hill, 2006.
9. Wang J., Hu Y., Duan Y., Jin W. Recursive Algorithm FIR Filter Quantization and Low-Cost Structure Optimization Design Based on MATLAB and Simulink, *2024 7th International Conference on Signal Processing and Information Communications (ICSPIC)*. IEEE, 2024. DOI: 10.1109/ICSPIC60374.2024.10387811.
10. Ali I., Hasan M., Ahmed S. Demonstration: Comparison of Three FIR Digital Filter Sharpening Techniques, *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020. DOI: 10.1109/ISCAS45731.2020.9180956.
11. Liu C., Li J., Zhou Y. Efficient Design of Sparse FIR Digital Filter, *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021. DOI: 10.1109/ISCAS51556.2021.9362902.
12. Zhou M., Li Y., Wang X. Adaptive Digital Filter Design Based on CIC Structure, *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2020. DOI: 10.1109/ICIEA48937.2020.9084880.
13. Tang L., Wei J., Chen R. Design of FIR Filter Based on FPGA, *2020 IEEE 3rd International Conference on Electronics and Communication Engineering (ICECE)*, 2020. DOI: 10.1109/ICECE49637.2020.9163582.
14. Antoniou A. Digital Signal Processing: Signals, Systems, and Filters. New York: McGraw–Hill, 2006.
15. Smith S.W. The Scientist and Engineer's Guide to Digital Signal Processing. California: California Technical Publishing, 1997.
16. Hogenauer Digital Filters (CIC Filters). Available at: <http://www.dsplib.ru/content/cic/cic.html> (accessed: 06 February 2025).
17. Pakhomov A.I. Diskretnaya fil'tratsiya signalov [Discrete Filtering of Signals]. Moscow: Vysshaya shkola, 1998.
18. Oppenheim A.V. The Evolution of Digital Signal Processing, *Proceedings of the IEEE*, 2000, No. 88, pp. 746-800.
19. Rabiner L.R. Digital Filters: An Introduction, *IEEE Spectrum*, 1977, No. 14, pp. 61-73.
20. Berendeev P.V., Kozlov S.A. Tsifrovaya obrabotka signalov: teoriya i prilozheniya [Digital signal processing: theory and applications]. Moscow: Nauka, 2005, pp. 78-82.
21. Harris F.J. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, *Proceedings of the IEEE*, 1978, No. 66, pp. 51-83.
22. Turulin I.I. Metod proektirovaniya rekursivnykh fil'trov obshchego vida s konechnoy impul'snoy kharakteristikoy [A Method for Designing General-Purpose Recursive Filters with Finite Impulse Response], *Izvestiya vuzov. Elektronika* [Izvestiya Universiteta. Elektronika], 2001, No. 6, pp. 97-98.
23. Turulin I.I. Osnovy teorii rekursivnykh KIKh-fil'trov: monografiya [Fundamentals of the Theory of Recursive FIR Filters: monograph]. Taganrog: Izd-vo YuFU, 2016, 264 p.
24. Rabiner L.R., Gold B. Theory and Application of Digital Signal Processing. Englewood Cliffs, NJ: Prentice–Hall, 1975.
25. Solov'ev V.N. Tsifrovye fil'try: teoriya i praktika [Digital Filters: Theory and Practice]. Saint Petersburg: Politekhnikeskoe izd-vo, 2004, pp. 45-50.

Моисеенко Иван Евгеньевич – Южный федеральный университет; e-mail: ivmoiseenko@sfedu.ru; г. Таганрог, Россия; тел.: +79649108479; аспирант.

Тарасов Сергей Павлович – Южный федеральный университет; e-mail: tsp-47@mail.ru; г. Таганрог, Россия; тел.: +79034007904; д.т.н.; профессор.

Турулин Игорь Ильич – Южный федеральный университет; e-mail: iiturulin@sfedu.ru; г. Таганрог, Россия; тел.: +79618170609; д.т.н.; профессор.

Moiseenko Ivan Evgenievich – Southern Federal University; e-mail: ivmoiseenko@sfedu.ru; Taganrog, Russia; phone: +79649108479; postgraduate student.

Tarasov Sergey Pavlovich – Southern Federal University; e-mail: tsp-47@mail.ru; Taganrog, Russia; phone: +79034007904; dr. of eng. sc.; professor.

Turulin Igor Ilyich – Southern Federal University; e-mail: iiturulin@sfedu.ru; Taganrog, Russia; phone: +79618170609; dr. of eng. sc.; professor.