

Лабораторная работа

Цифровая фильтрация. Реализация фильтра с конечной импульсной характеристикой (КИХ)

1 Цифровая фильтрация

Цифровая фильтрация является одним из наиболее мощных инструментальных средств ЦОС. Кроме очевидных преимуществ устранения ошибок в фильтре, связанных с флуктуациями параметров пассивных компонентов во времени и по температуре, дрейфом ОУ (в активных фильтрах) и т.д., цифровые фильтры способны удовлетворять таким техническим требованиям по своим параметрам, которых, в лучшем случае, было бы чрезвычайно трудно или даже невозможно достичь в аналоговом исполнении. Кроме того, характеристики цифрового фильтра могут быть легко изменены программно. Поэтому они широко используются в телекоммуникациях, в приложениях адаптивной фильтрации, таких как подавление эха в модемах, подавление шума и распознавание речи.

Процесс проектирования цифровых фильтров состоит из тех же этапов, что и процесс проектирования аналоговых фильтров. Сначала формулируются требования к желаемым характеристикам фильтра, по которым затем рассчитываются параметры фильтра. Амплитудная и фазовая характеристики формируются аналогично аналоговым фильтрам. Ключевое различие между аналоговым и цифровым фильтрами заключается в том, что, вместо вычисления величин сопротивлений, емкостей и индуктивностей для аналогового фильтра, рассчитываются значения коэффициентов для цифрового фильтра. Иными словами, в цифровом фильтре числа заменяют физические сопротивления и емкости аналогового фильтра. Эти числа являются коэффициентами фильтра, они постоянно находятся в памяти и используются для обработки (фильтрации) дискретных данных, поступающих от АЦП.

Цифровой фильтр, работающий в реальном масштабе времени, оперирует с дискретными по времени данными в противоположность непрерывному сигналу, обрабатываемому аналоговым фильтром. При этом очередной отсчет, соответствующий отклику фильтра, формируется по окончании каждого периода дискретизации. Вследствие дискретной природы обрабатываемого сигнала, на отсчеты данных зачастую ссылаются по их номерам, например, отсчет 1, отсчет 2, отсчет 3 и т.д. На рисунке 1.1 представлен низкочастотный сигнал, содержащий высокочастотный шум, который должен быть отфильтрован. Вначале сигнал должен быть оцифрован с помощью АЦП для получения выборки $x(n)$. Далее эта выборка поступает на цифровой фильтр, который в данном случае является НЧ фильтром. Отсчеты выходных данных

$y(n)$ используются для восстановления аналогового сигнала с использованием ЦАП с низким уровнем ложного сигнала.

Тем не менее, цифровые фильтры не могут являться решением всех возможных задач фильтрации, возникающих при обработке сигналов. Для работы в реальном масштабе времени, DSP-процессор должен быть рассчитан на выполнение всех шагов в программе фильтрации в пределах промежутка времени, соответствующего одному такту дискретизации, то есть $1/f_s$. Высокопроизводительный универсальный DSP процессор с фиксированной точкой типа ADSP-2189M, обладающий быстродействием 75MIPS, способен выполнить операцию умножения с накоплением при реализации одного каскада фильтра за 13,3 нс. DSP процессор ADSP-2189M затрачивает $N+5$ инструкций при реализации фильтра с количеством каскадов N . Для 100-каскадного фильтра полное время вычисления составляет приблизительно 1,4 мкс. Это соответствует максимально возможной частоте дискретизации 714 кГц, ограничивая, таким образом, ширину полосы частот обрабатываемого сигнала несколькими сотнями килогерц.

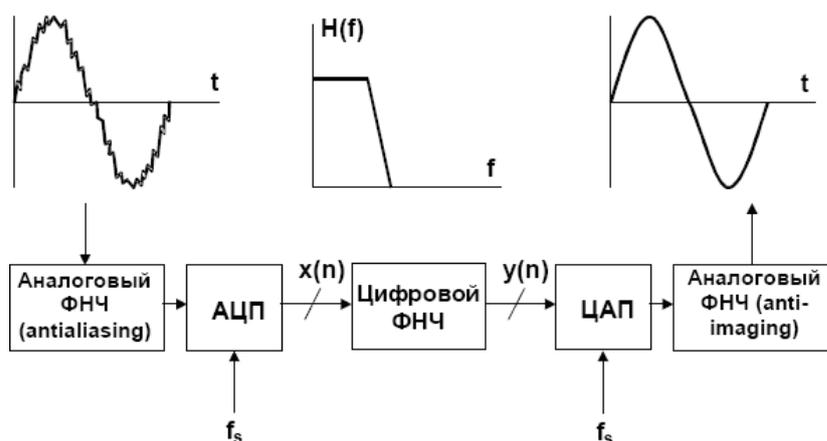


Рисунок 1.1 – Структура цифрового фильтра.

Можно заменить универсальный DSP процессор специализированным аппаратным цифровым фильтром, способным работать на частотах дискретизации, соответствующих видеосигналу. В других случаях ограничения по быстродействию могут быть преодолены сохранением выборки данных, поступающих с большой скоростью от АЦП, в буферной памяти. Затем буферная память читается со скоростью, совместимой с быстродействием цифрового фильтра, основанного на DSP. Используя данный метод, может осуществляться обработка сигнала в псевдореальном масштабе времени в таких системах как радар, где обычно обрабатываются пакеты данных, накапливаемые после каждого излучаемого импульса.

Другой подход заключается в использовании специализированных микросхем цифровых фильтров, подобных фильтрам PulseDSP™ компании Systolix. 16 разрядный сигма-дельта-АЦП AD7725 имеет на своем кристалле фильтр

PulseDSP, который может выполнять за секунду 125 миллионов операций умножения с накоплением.

В дискретных системах, даже с высокой степенью избыточной дискретизации, требуется наличие аналоговых ФНЧ перед АЦП и после ЦАП для устранения эффекта наложения спектра. Более того, с ростом частоты, сигналы выходят за рамки возможностей доступных АЦП, и цифровая фильтрация становится невозможной. Но на крайне высоких частотах и активная аналоговая фильтрация тоже невозможна из-за ограничений, связанных с полосой пропускания и искажениями ОУ, и в этих случаях требования фильтрации удовлетворяются пассивными элементами. Дальнейшее обсуждение будет сфокусировано, в первую очередь, на фильтрах, которые могут работать в реальном масштабе времени и могут быть программно реализованы с использованием DSP.

В качестве примера сравним аналоговый и цифровой фильтры, показанные на рисунке 1.2. Частота среза обоих фильтров равна 1 кГц. Аналоговый фильтр реализован в виде фильтра Чебышева первого рода 6 порядка (характеризуется неравномерностью коэффициента передачи в полосе пропускания и равномерностью коэффициента передачи вне полосы пропускания). На практике этот фильтр может быть собран на трех фильтрах второго порядка, каждый из которых состоит из операционного усилителя и нескольких резисторов и конденсаторов. Проектирование фильтра 6 порядка является непростой задачей, а удовлетворение техническим требованиям по неравномерности характеристики в 0,5 дБ требует точного подбора компонентов.

С другой стороны, представленный цифровой фильтр с конечной импульсной характеристикой (КИХ) имеет неравномерность характеристики всего 0,002 дБ в полосе пропускания, линейную фазовую характеристику и значительно более крутой спад частотной характеристики. Таких показателей невозможно достичь аналоговыми методами. На практике существует много других факторов, учитываемых при сравнительной оценке аналоговых и цифровых фильтров. В большинстве современных систем обработки сигналов используются комбинации аналоговых и цифровых методов для реализации желаемых функций и используются преимущества всех методов, как аналоговых, так и цифровых.

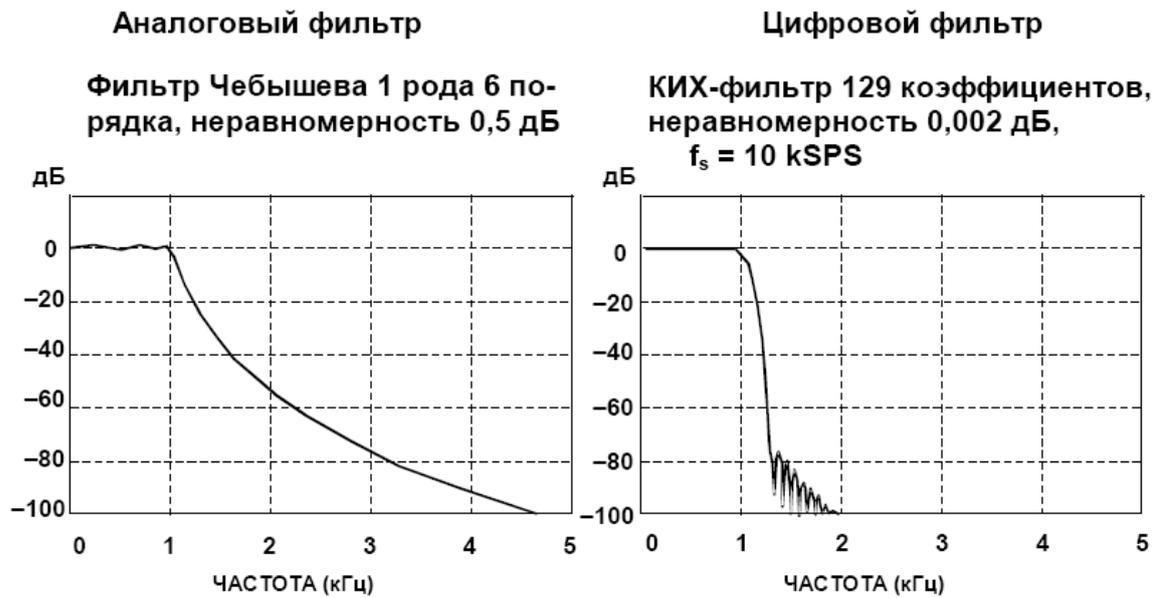


Рисунок 1.2– Сравнение АЧХ цифрового и аналогового фильтров.

Таблица 1.1 – Сравнение характеристик цифровых и аналоговых фильтров.

Цифровые фильтры	Аналоговые фильтры
Высокая точность	Низкая точность из-за допуска на элементы
Линейная фаза (КИХ фильтр)	Нелинейная фаза
Нет дрейфа вследствие изменения параметров компонентов	Дрейф вследствие изменения параметров компонентов
Гибкость, возможна адаптивная фильтрация	Реализация адаптивных фильтров затруднена
Легки в моделировании и проектировании	Сложны в моделировании и проектировании
Ограничения при работе в реальном масштабе времени – вычисление должно быть завершено в течение интервала дискретизации	Аналоговые фильтры требуются на высоких частотах и для устранения эффекта наложения спектра

Существует много приложений, в которых цифровые фильтры должны работать в реальном масштабе времени. В них накладываются определенные требования на процессор DSP в зависимости от частоты дискретизации и сложности фильтра. Ключевым моментом является то, что процессор DSP должен проводить все вычисления в течение интервала дискретизации, чтобы быть готовым к обработке следующего отсчета данных. Пусть ширина полосы частот обрабатываемого сигнала равна f_a . Тогда частота дискретизации АЦП f_s должна быть, по крайней мере, в два раза больше, то есть $2f_a$. Интервал

дискретизации равен $1/f_s$. Все вычисления, связанные с реализацией фильтра (включая все дополнительные операции), должны быть закончены в течение этого интервала. Время вычислений зависит от числа звеньев фильтра и быстродействия и эффективности процессора DSP. Каждое звено при реализации фильтра требует одной операции умножения и одной операции сложения (умножения с накоплением). Процессор DSP оптимизируется для быстрого выполнения операций умножения с накоплением. Кроме того, многие процессоры DSP имеют дополнительные особенности, такие как реализация циклической адресации и организация программных циклов с автоматической проверкой условия продолжения цикла, минимизирующие количество дополнительных инструкций, которые в противном случае были бы необходимы.

Требования к цифровой фильтрации для работы в реальном масштабе времени:

1. Полоса сигнала = f_a
2. Частота дискретизации $f_s > 2f_a$
3. Период дискретизации $=1/f_s$
4. Время вычисления фильтра + доп. операции $<$ период дискр.
5. Зависит от числа коэффициентов фильтра
6. Скорости операций умножения с накоплением (MAC)
7. Эффективности ЦОС
8. Поддержка циклических буферов
9. Отсутствие дополнительных операций

Фильтры с конечной импульсной характеристикой (КИХ)

Существует два основных типа цифровых фильтров: фильтры с конечной импульсной характеристикой (КИХ) и фильтры с бесконечной импульсной характеристикой (БИХ). Как следует из терминологии, эта классификация относится к импульсным характеристикам фильтров. Изменяя веса коэффициентов и число звеньев КИХ фильтра, можно реализовать практически любую частотную характеристику. КИХ фильтры могут иметь такие свойства, которые невозможно достичь методами аналоговой фильтрации (в частности, совершенную линейную фазовую характеристику). Но высокоэффективные КИХ фильтры строятся с большим числом операций умножения с накоплением и поэтому требуют использования быстрых и эффективных процессоров DSP. С другой стороны, БИХ фильтры имеют тенденцию имитировать принцип действия традиционных аналоговых фильтров с обратной связью. Поэтому их импульсная характеристика имеет бесконечную длительность. Благодаря использованию обратной связи, БИХ фильтры могут быть реализованы с меньшим количеством коэффициентов, чем КИХ фильтры. Другим способом реализации КИХ или БИХ фильтрации являются решетчатые фильтры, которые часто используются в задачах

обработки речи. Цифровые фильтры применяются в приложениях адаптивной фильтрации, благодаря своему быстрдействию и простоте изменения характеристик воздействием на его коэффициенты.

Типы цифровых фильтров:

1. Фильтр скользящего среднего
2. Фильтр с конечной импульсной характеристикой (КИХ)

1. Линейная фаза
2. Легкость проектирования
3. Значительные вычислительные затраты

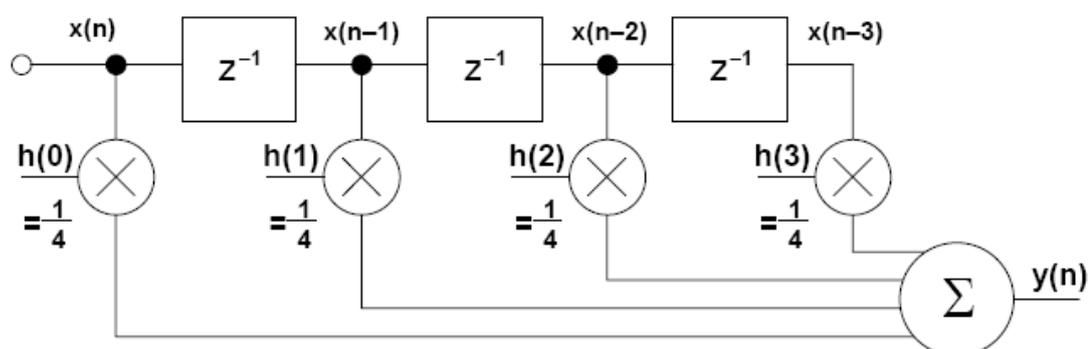
3. Фильтр с бесконечной импульсной характеристикой

1. Основаны на классических аналоговых фильтрах
2. Высокая вычислительная эффективность

4. Решетчатые фильтры (могут быть КИХ или БИХ)
5. Адаптивные фильтры

Элементарной формой КИХ фильтра является фильтр скользящего среднего (moving average), показанный на рисунке 1.3. Фильтры скользящего среднего популярны для сглаживания данных, например, для анализа стоимости акций и т.д. Входные отсчеты $x(n)$ пропускаются через ряд регистров памяти (помеченных z^{-1} в соответствии с представлением элемента задержки при z -преобразовании). В приведенном примере имеется четыре каскада, соответствующих 4-точечному фильтру скользящего среднего. Каждый отсчет умножается на 0,25, и результаты умножения суммируются для получения значения скользящего среднего, которое подается на выход $y(n)$. На рисунке также представлено общее уравнение фильтра скользящего среднего на N точек. N относится к числу точек при вычислении фильтра, а не к разрешающей способности АЦП или ЦАП.

4-ТОЧЕЧНЫЙ ФИЛЬТР СКОЛЬЗЯЩЕГО СРЕДНЕГО



$$\begin{aligned}
 y(n) &= h(0) x(n) + h(1) x(n-1) + h(2) x(n-2) + h(3) x(n-3) \\
 &= \frac{1}{4} x(n) + \frac{1}{4} x(n-1) + \frac{1}{4} x(n-2) + \frac{1}{4} x(n-3) \\
 &= \frac{1}{4} [x(n) + x(n-1) + x(n-2) + x(n-3)]
 \end{aligned}$$

Для N-точечного
фильтра скользящего среднего $y(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(n-k)$

Рисунок 1.3 – 4-х точечный фильтр скользящего среднего.

С учетом равенства коэффициентов, наиболее простой путь исполнения фильтра скользящего среднего представлен на рисунке 1.4. Обратите внимание, что первым шагом является запоминание первых четырех отсчетов $x(0)$, $x(1)$, $x(2)$, $x(3)$ в регистрах. Эти величины суммируются и затем умножаются на 0,25 для получения первого выхода $y(3)$. Обратите внимание, что начальные значения выходов $y(0)$, $y(1)$ и $y(2)$ некорректны, потому что, пока отсчет $x(3)$ не получен, не все регистры заполнены.

$$\begin{aligned}
 y(3) &= 0.25 \left[\begin{array}{c} x(3) + x(2) + x(1) + x(0) \end{array} \right] \\
 y(4) &= 0.25 \left[\begin{array}{c} x(4) + x(3) + x(2) + x(1) \end{array} \right] \\
 y(5) &= 0.25 \left[\begin{array}{c} x(5) + x(4) + x(3) + x(2) \end{array} \right] \\
 y(6) &= 0.25 \left[\begin{array}{c} x(6) + x(5) + x(4) + x(3) \end{array} \right] \\
 y(7) &= 0.25 \left[\begin{array}{c} x(7) + x(6) + x(5) + x(4) \end{array} \right]
 \end{aligned}$$

-
- Вычисление каждого выходного значения требует
- 1 умножения, 1 сложения и 1 вычитания
-

Рисунок 1.4 – Вычисление выходного сигнала фильтра скользящего среднего.

Когда получен отсчет $x(4)$, он суммируется с результатом, а отсчет $x(0)$ вычитается из результата. Затем новый результат должен быть умножен на 0,25. Поэтому вычисления, требуемые для получения нового значения на выходе, состоят из одного суммирования, одного вычитания и одного умножения, независимо от длины фильтра скользящего среднего.

Реакция 4-точечного фильтра скользящего среднего на ступенчатое воздействие представлена на рисунке 1.5. Скользящего среднего не имеет выброса по фронту входного сигнала. Это делает его полезным в приложениях обработки сигналов, где требуется фильтрация случайного белого шума при сохранении характера входного импульса. Из всех возможных линейных фильтров, фильтр скользящего среднего дает самый низкий уровень шума при заданной крутизне фронта импульса. Это показано на рисунке 1.6, где уровень шума понижается по мере увеличения числа точек. Существенно, что время реакции фильтра на ступенчатое воздействие от 0 % до 100 % равно произведению общего количества точек фильтра на период дискретизации.

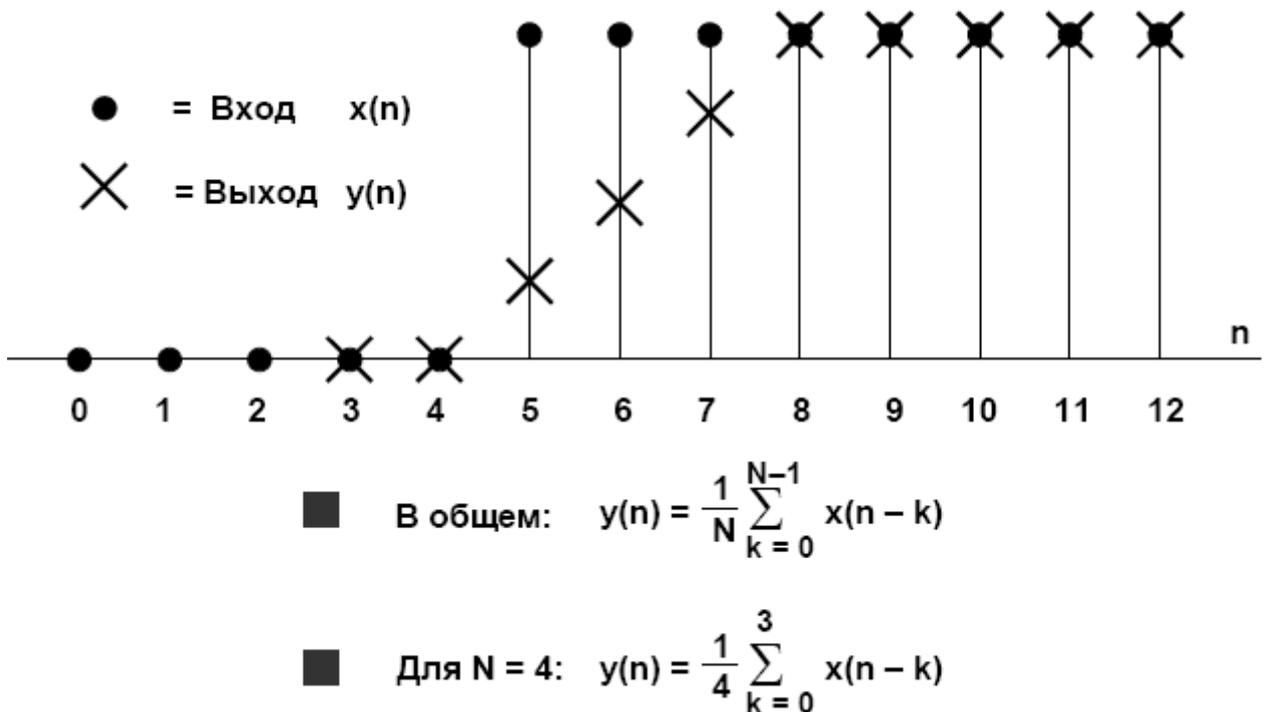


Рисунок 1.5 – Реакция фильтра скользящего среднего на ступенчатое воздействие.

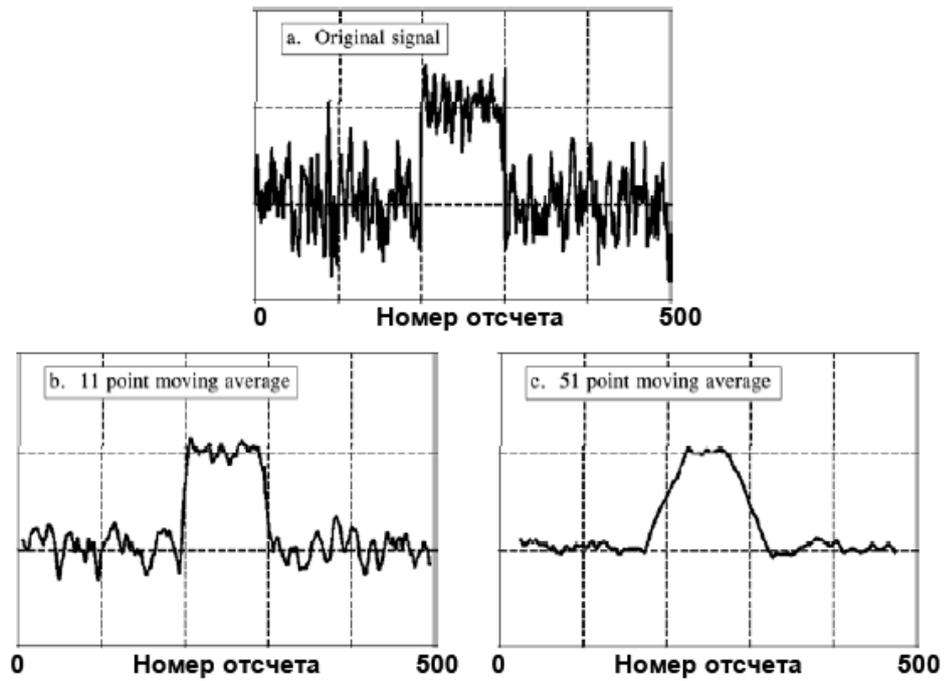


Рисунок 1.5 – Реакция фильтра скользящего среднего на воздействие в виде смеси шума и ступенчатого сигнала.

Частотная характеристика простого фильтра скользящего среднего выражается функцией $\sin(x)/x$. Она представлена в линейном масштабе на рисунке 1.6. Увеличение числа точек при реализации фильтра сужает основной лепесток, но существенно не уменьшает амплитуду боковых лепестков частотной характеристики, которая равна приблизительно -14 дБ для фильтра с 11 и с 31 отводами (длиной буфера). Естественно, эти фильтры не подходят в том случае, где требуется большое ослабление в полосе задержания.

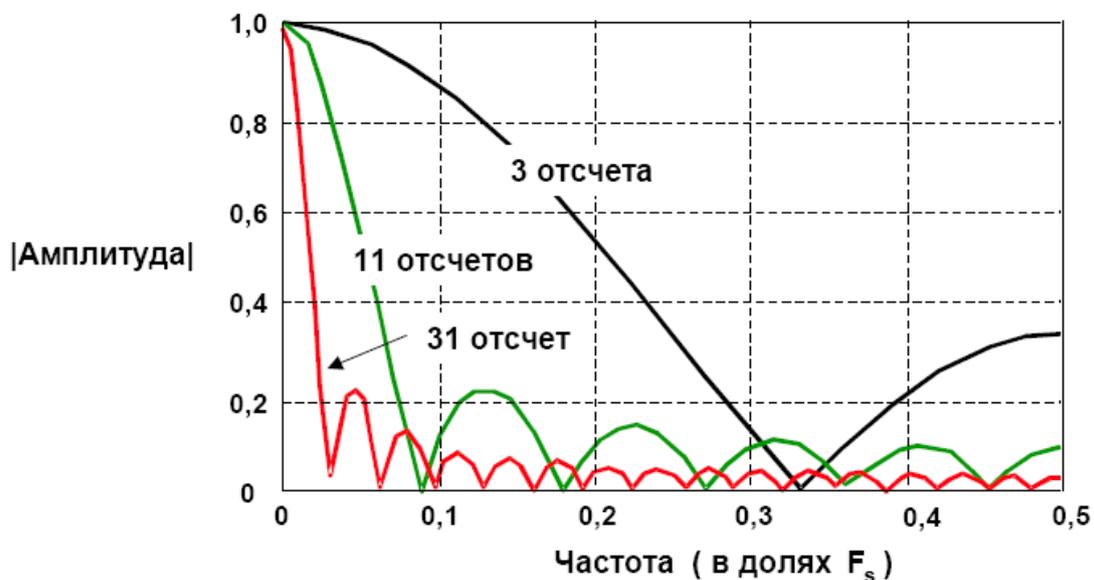
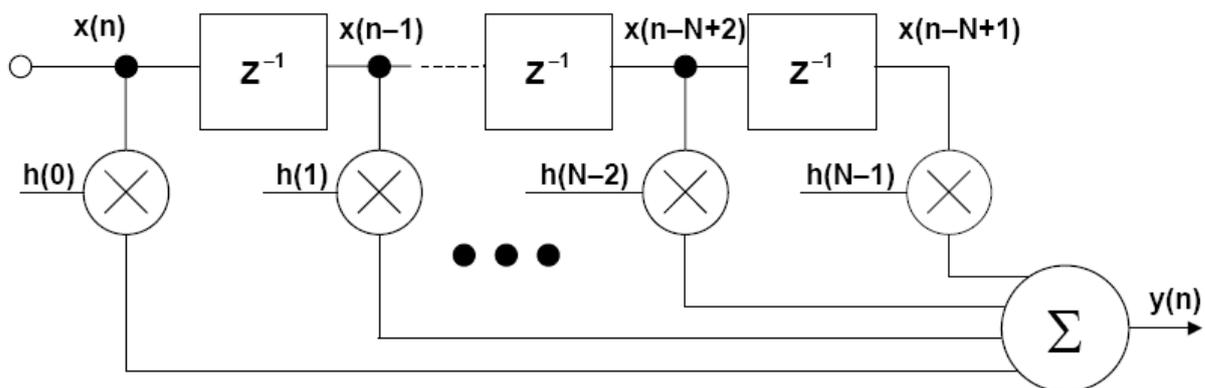


Рисунок 1.6 – АЧХ фильтра скользящего среднего.

Можно существенно улучшить эффективность простого КИХ фильтра скользящего среднего, выбирая разные веса или значения коэффициентов вместо равных значений. Крутизна спада может быть увеличена добавлением большего количества звеньев в фильтр, а характеристики полосы затухания улучшаются выбором надлежащих коэффициентов фильтра. Обратите внимание, что, в отличие от фильтра скользящего среднего, для реализации каждой ступени обобщенного КИХ фильтра требуется цикл умножения с накоплением. Сущность проектирования КИХ фильтра сводится к выбору соответствующих коэффициентов и необходимого числа звеньев при формировании желаемой частотной характеристики фильтра $H(f)$. Для включения необходимой частотной характеристики $H(f)$ в набор КИХ коэффициентов имеются различные алгоритмы и программные пакеты. Большинство этого программного обеспечения разработано для персональных компьютеров и доступно на рынке. Ключевой теоремой проектирования КИХ фильтра является утверждение, что коэффициенты $h(n)$ КИХ фильтра являются просто квантованными значениями импульсной характеристики этого фильтра. Соответственно, импульсная характеристика является дискретным преобразованием Фурье от $H(f)$.



■ $y(n) = h(n) * x(n) = \sum_{k=0}^{N-1} h(k) x(n - k)$

■ * = символ свертки

■ Требуется N операций умножения с накоплением для каждого выходного отсчета

Рисунок 1.7 – Фильтр с конечной импульсной характеристикой порядка N.

Обобщенная форма КИХ фильтра с числом звеньев N представлена на рисунке 1.7. Как было сказано, КИХ фильтр должен работать в соответствии с уравнением, задающим свертку:

$$Y(n) = h(k) * x(n) = \sum_{k=0}^{N-1} h(k)x(n-k),$$

где $h(k)$ – массив коэффициентов фильтра и $x(n-k)$ – входной массив данных фильтра. Число N в уравнении представляет собой число звеньев и определяет эффективность фильтра, как было сказано выше. КИХ фильтр с числом звеньев N требует N циклов (операций) умножения с накоплением.

Согласно рисунку 1.8, диаграммы КИХ-фильтров часто изображаются в упрощенном виде. Операции суммирования представляются стрелками, указывающими в точки, а операции умножения обозначают, помещая коэффициенты $h(k)$ рядом со стрелками на линиях. Элемент задержки z^{-1} показывают, помещая его обозначение выше или рядом с соответствующей линией.

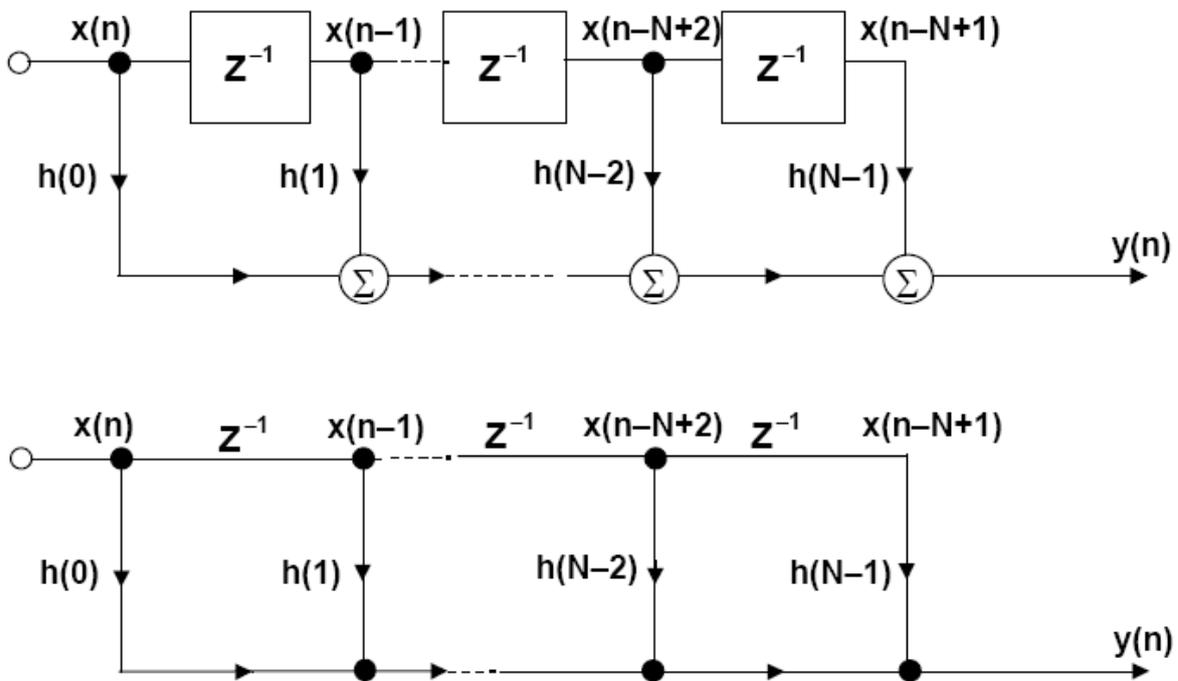


Рисунок 1.8 – Упрощенная схема фильтра.

Реализация КИХ-фильтра на процессоре DSP с использованием циклических буферов

В рядах, задаваемых уравнениями КИХ фильтров, предполагается последовательное обращение к N коэффициентам от $h(0)$ до $h(N-1)$. Соответствующие точки данных циркулируют в памяти. При этом добавляются новые отсчеты данных, заменяя самые старые, и каждый раз производится вычисление выходного значения фильтра. Для реализации циклического буфера может использоваться фиксированный объем оперативной памяти, как показано на рисунке 1.9 для КИХ фильтра с 4

звеньями. Самый старый отсчет данных заменяется новым после каждой операции вычисления свертки. Выборка из четырех последних отсчетов данных всегда сохраняется в оперативной памяти.

Чтобы упростить адресацию, чтение из памяти старых значений начинается с адреса, который следует непосредственно за адресом только что записанного нового элемента выборки. Например, если значение $x(4)$ только что записано в ячейку памяти 0, то значения данных читаются из ячеек 1, 2, 3 и 0. Этот пример может быть расширен применительно к любому числу звеньев фильтра. Используя адресацию ячеек памяти таким способом, адресный генератор должен лишь вычислять последовательные адреса, независимо от того, является ли данная операция чтением памяти или записью. Такой буфер в памяти данных называется циклическим, потому что, когда достигается его последняя ячейка, указатель автоматически позиционируется на начало буфера.

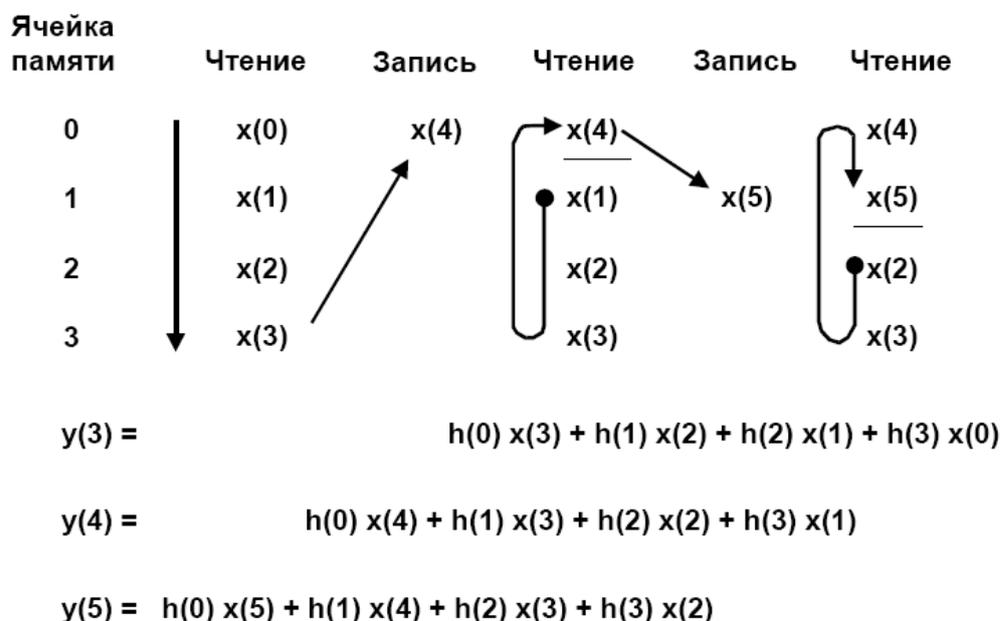


Рисунок 1.9 – Вычисление выходного сигнала КИХ фильтра 4-ого порядка с использованием циклического буфера.

Выборка коэффициентов из памяти осуществляется одновременно с выборкой данных. В соответствии с описанной схемой адресации, самый старый отсчет данных выбирается первым. Поэтому сначала должна осуществляться выборка из памяти последнего коэффициента. При использовании адресного генератора, поддерживающего инкрементную адресацию, коэффициенты могут быть сохранены в памяти в обратном порядке: $h(N-1)$ помещается в первую ячейку, а $h(0)$ – в последнюю. И наоборот, коэффициенты могут быть сохранены в порядке возрастания их номеров, если использовать адресный генератор, поддерживающий декрементную адресацию. В примере, показанном на рисунке 1.9, коэффициенты сохранены в обратном порядке.

Простая итоговая блок-схема для этих операций представлена на рисунке 1.10. Все операции, выполняемые за один цикл фильтра, производятся за один командный цикл процессора, благодаря чему существенно увеличивается эффективность вычислений. Данное преимущество известно как реализация циклов без дополнительных операций.

Ограничение на число звеньев фильтра, реализующего подпрограммы КИХ фильтрации в реальном масштабе времени, определяется, прежде всего, длительностью процессорного цикла, частотой дискретизации и требуемым объемом других вычислений.

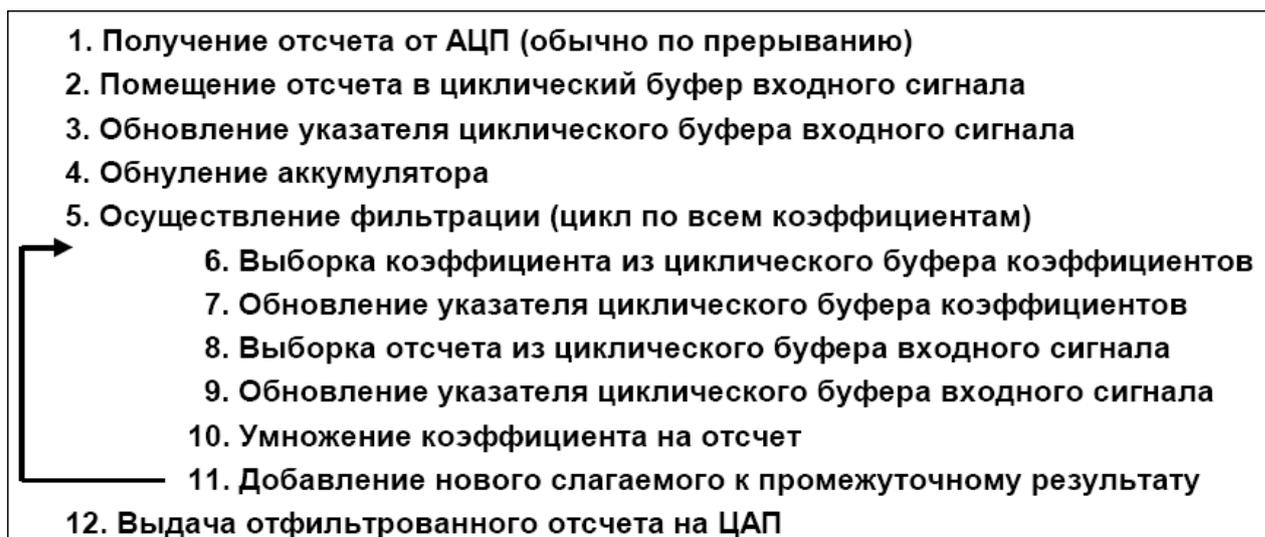


Рисунок 1.10 – Псевдокод программы фильтра, использующей DSP процессор с циклической буферизацией.

Лабораторная работа

Цифровая фильтрация. Реализация фильтра с бесконечной импульсной характеристикой (БИХ)

1. Фильтры с бесконечной импульсной характеристикой – бих

КИХ фильтры не имеют реальных аналоговых эквивалентов. Самой близкой аналогией является фильтр скользящего среднего с взвешиванием. Кроме того, частотные характеристики КИХ фильтров имеют только нули и не имеют полюсов. С другой стороны, БИХ фильтры имеют традиционные аналоговые эквиваленты (фильтры Баттерворта, Чебышева, эллиптический и Бесселя) и могут быть проанализированы и синтезированы с использованием традиционных методов проектирования фильтров.

БИХ фильтры получили такое название, потому что их импульсные характеристики растянуты на бесконечном временном интервале. Это объясняется тем, что данные фильтры являются рекурсивными, то есть

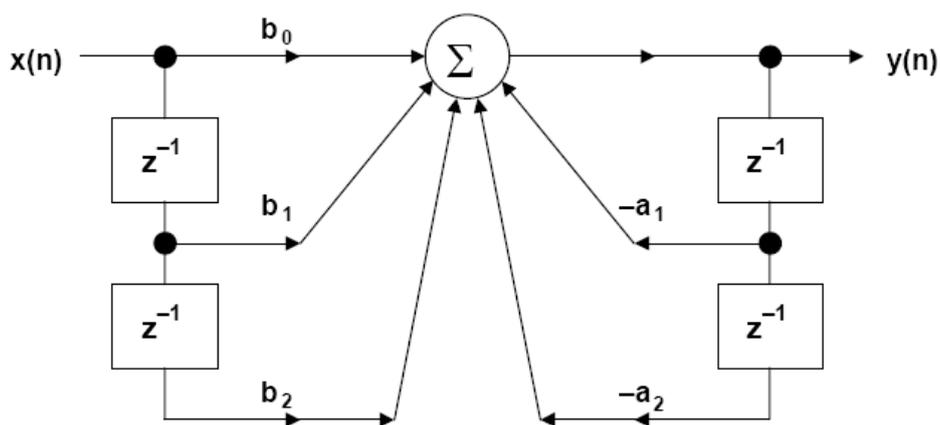
используют обратную связь. Хотя БИХ фильтры могут быть реализованы с меньшим, чем КИХ фильтры, количеством вычислений, БИХ фильтры не могут иметь таких характеристик, которыми обладают КИХ фильтры. Более того, БИХ фильтр не имеет линейной фазовой характеристики. Но вычислительные преимущества БИХ фильтра теряются, когда выходной сигнал фильтра подвергается децимации, поскольку в этом случае всякий раз приходится вычислять заново значение выходной величины.

БИХ фильтры обычно реализуются с помощью звеньев второго порядка, которые называются биквадратными фильтрами, потому что описываются биквадратными уравнениями в z -области. Фильтры высокого порядка проектируют, используя каскадирование биквадратных звеньев. Например, фильтр шестого порядка требует трех биквадратных звеньев.

Особенности фильтра с бесконечной импульсной характеристикой:

1. Имеют обратную связь (рекурсия)
2. Импульсная характеристика имеет бесконечную длительность
3. Потенциально нестабильны
4. Нелинейная фазочастотная характеристика
5. Более эффективны, чем КИХ-фильтры
6. Нет вычислительных преимуществ при децимации по выходу
7. Обычно проектируется по характеристике аналогового фильтра
8. Обычно реализуется каскадным соединением звеньев второго порядка (биквадратные фильтры)

Структура биквадратного БИХ фильтра представлена на рисунке 1.1. Нули формируются коэффициентами прямой связи b_0 , b_1 и b_2 ; а полюса (порядок) определяются коэффициентами обратной связи a_1 и a_2 .



■ $y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2)$

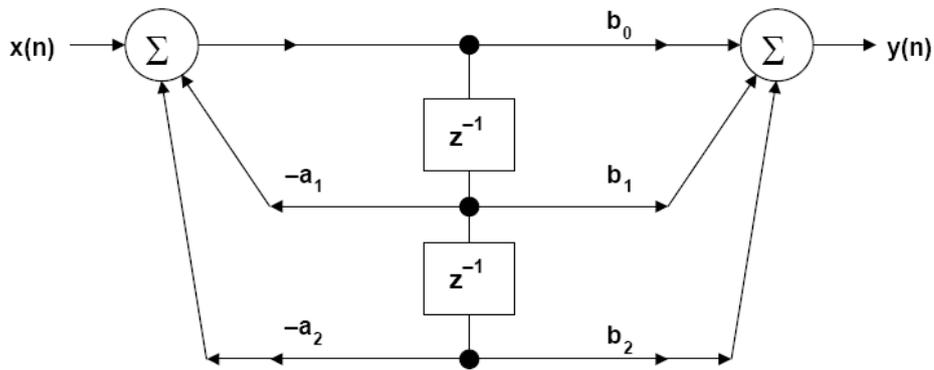
■ $y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k x(n-k)$ ■ $H(z) = \frac{\sum_{k=0}^M b_k z^{-k} \quad (\text{нули})}{1 + \sum_{k=1}^N a_k z^{-k} \quad (\text{полюса})}$

Рисунок 1.1 – Аппаратная реализация БИХ фильтра второго порядка (биквадратного).

Общее уравнение цифрового фильтра, представленное на рисунке 1.1, описывает обобщенную передаточную функцию $H(z)$, которая содержит полиномы и в числителе, и в знаменателе. Корни знаменателя определяют расположение полюсов фильтра, а корни числителя характеризуют расположение нулей. Хотя существует возможность создания непосредственно по этому уравнению БИХ фильтра более высокого порядка (так называемая прямая реализация), накапливающиеся ошибки квантования (из-за арифметики с фиксированной точкой и конечной длины слова) могут вызывать неустойчивость работы фильтра и большие ошибки. По этой причине правильнее расположить каскадно несколько биквадратных звеньев с соответствующими коэффициентами, чем использовать прямую форму реализации. Данные при вычислении биквадратных фильтров могут масштабироваться отдельно, а затем биквадратные звенья каскадируются для минимизации ошибок квантования коэффициентов и накапливающихся ошибок рекурсивного накопления. Каскадные биквадратные фильтры работают более медленно, чем их эквиваленты прямой формы реализации, но они более устойчивы и в них минимизируются эффекты, связанные с арифметическими ошибками конечной разрядности данных.

Первая прямая форма биквадратного звена, представленная на рисунке 1.1, требует использования четырех регистров. Эта конфигурация может быть заменена эквивалентной схемой, представленной на рисунке 1.2, которая называется второй прямой формой реализации и требует использования только двух регистров. Можно показать, что уравнения, описывающие биквадратный БИХ фильтр второй прямой формы реализации, такие же, как и

уравнения первой прямой формы реализации. Как и в случае КИХ фильтра, система обозначений при изображении БИХ фильтра часто упрощается, как показано на рисунке 1.3.



- Приводится к такому же уравнению, как для первой прямой формы:
- $y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2)$
- Требуется только 2 элемента задержки (регистра)

Рисунок 1.2 – Биквадратный БИХ фильтр форма 2.

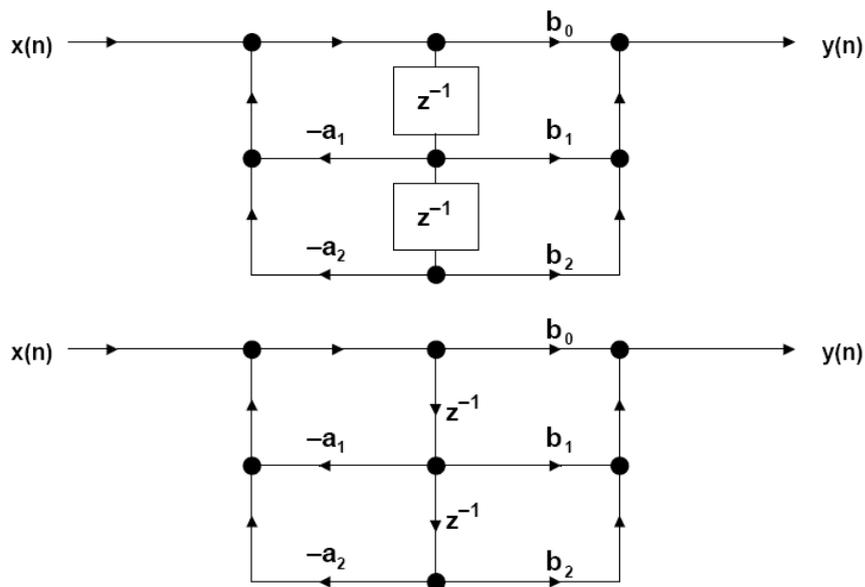


Рисунок 1.3 – Упрощенная схема биквадратного БИХ фильтра.

Методы проектирования БИХ фильтров

Популярный метод проектирования БИХ фильтра сводится к тому, что сначала проектируется эквивалентный аналоговый фильтр, а затем функция передачи $H(s)$ преобразуется математически в z -область, $H(z)$. Проектирование фильтров более высоких порядков выполняется каскадированием биквадратных звеньев. Наиболее популярными

аналоговыми фильтрами являются фильтры Баттерворта, Чебышева, эллиптические и Бесселя. Существует множество программ САПР, способных генерировать функцию передачи фильтра, заданную с помощью преобразования Лапласа.

Фильтр Баттерворта, не имеющий нулей частотной характеристики, (также называемый фильтром с максимально плоской характеристикой), не создает пульсаций (неравномерности) в полосе пропускания и в полосе задержки, то есть обладает монотонной характеристикой в обеих полосах. Фильтр Чебышева 1-го рода имеет более быстрый спад частотной характеристики, чем фильтр Баттерворта (при равном порядке), и создает пульсации (неравномерность) в полосе пропускания. Реже используются фильтры Чебышева 2-го рода, имеющие пульсации (неравномерность) в полосе задержки, а не в полосе пропускания.

Эллиптический фильтр (фильтр Кауэра) имеет полюса и нули частотной характеристики и создает пульсации (неравномерность) и в полосе пропускания, и в полосе задержки. Этот фильтр имеет более быстрый спад частотной характеристики, чем фильтр Чебышева при том же числе полюсов (порядке). Эллиптический фильтр часто используется там, где допускается несколько худшая фазовая характеристика.

Наконец, фильтр Бесселя (Томпсона), который не имеет нулей частотной характеристики, обладает оптимальной импульсной характеристикой и линейной фазовой характеристикой, но имеет худший спад частотной характеристики из всех типов обсуждавшихся фильтров при том же числе полюсов (порядке).

Популярные аналоговые фильтры:

1. Баттерворта

1. Нет нулей частотной характеристики, нет пульсаций в полосе пропускания и задержки
2. Максимально плоская характеристика (быстрый спад без пульсаций)

2. Чебышева 1-го рода

1. Нет нулей частотной характеристики, пульсации в полосе пропускания, нет пульсаций в полосе задержки
2. Более короткая область перехода, чем у фильтра Баттерворта для данного порядка
3. Фильтр 2-го рода имеет пульсации в полосе задержки, нет пульсаций в полосе пропускания

3. Эллиптический (Кауэра)

1. Имеет полюса и нули, пульсации и в полосе пропускания, и в полосе задержки
2. Более короткая область перехода, чем у фильтра Чебышева для данного порядка
3. Фазовая характеристика хуже

4. Бесселя (Томпсона)

1. Нет нулей частотной характеристики, нет пульсаций в полосе пропускания и задержки
2. Оптимизирован по линейной фазовой и импульсной характеристикам
3. Самая длинная переходная характеристика из всех фильтров данного порядка

Все вышеперечисленные типы аналоговых фильтров описаны в литературе, их преобразования по Лапласу $H(s)$ доступны либо из таблиц, либо могут быть получены с помощью средств САПР. Существует три метода преобразования изображения по Лапласу в z -изображение: метод инвариантности импульсной характеристики, билинейное преобразование и согласованное z преобразование. Результирующее z изображение может быть преобразовано в коэффициенты биквадратного БИХ фильтра.

Подход САПР при проектировании БИХ фильтра подобен программе Паркса-Макклиллана, используемой для КИХ фильтров. Эта методика использует алгоритм Флетчера-Пауэла (Fletcher-Powell).

При вычислении производительности специального процессора DSP, предназначенного для реализации БИХ фильтров, необходимо исследовать эталонные требования эффективности вычислений для биквадратного звена фильтра.

Методы проектирования БИХ фильтров:

1. Метод инвариантности импульсной характеристики

1. Начинается с определения $H(s)$ для аналогового фильтра
2. Взятие обратного преобразования Лапласа для получения импульсной характеристики
3. Получение z -преобразования $H(z)$ из дискретной импульсной характеристики
4. z -преобразование выдает коэффициенты фильтра
5. Должен быть учтен эффект наложения спектров

2. Метод билинейного преобразования

1. Другой метод для преобразования $H(s)$ в $H(z)$

2. Характеристики определяются дифференциальным уравнением, описывающим аналоговую систему
3. Не важен эффект наложения спектра
3. Метод согласованного z-преобразования
1. Отображает $H(s)$ в $H(z)$ для фильтров и с полюсами, и с нулями
4. Методы САПР
1. Алгоритм Флетчера-Пауэла
2. Осуществляются каскадированием биквадратных звеньев

Скорость обработки данных при реализации БИХ фильтров:

1. Определение количества биквадратных звеньев, требуемых для реализации желаемой частотной характеристики.
2. Умножение этого количества на время выполнения одного биквадратного звена для DSP процессора.
3. Результат (плюс дополнительные операции) является минимально допустимым периодом дискретизации ($1/f_s$) для работы в реальном масштабе времени.

Сравнение КИХ и БИХ фильтров

Выбор между КИХ и БИХ фильтрами может быть своего рода состязанием в проектировании, но несколько основных руководящих принципов дать можно. Как правило, БИХ фильтры более эффективны, чем КИХ фильтры, потому что они требуют меньшего количества памяти и меньшего количества операций умножения с накоплением. БИХ фильтры могут быть разработаны, основываясь на предыдущем опыте проектирования аналоговых фильтров. БИХ фильтры могут приносить проблемы неустойчивости, но это происходит реже, если проектируемые фильтры высокого порядка реализуются как системы, состоящие из каскадов второго порядка.

С другой стороны, КИХ фильтры требуют большего количества звеньев и, соответственно, операций умножения с накоплением для реализации частотной характеристики с заданной частотой среза, но при этом имеют линейную фазовую характеристику. КИХ фильтры работают на конечном временном интервале данных, поэтому, если часть данных испорчена (например, в результате сбоев в работе АЦП), КИХ фильтр будет «звенеть» только на временном интервале, соответствующем $N-1$ отсчетам. БИХ фильтр, из-за наличия обратной связи, будет «звенеть» значительно более длительный период времени.

Таблица 1.1 – Сравнение КИХ и БИХ фильтров

БИХ-фильтры	КИХ-фильтры
Более эффективны	Менее эффективны
Есть аналоговый эквивалент	Нет аналогового эквивалента
Могут быть нестабильными	Всегда стабильные
Нелинейная фазовая характеристика	Линейная фазовая характеристика
Больше «звон» при наличии ложных сигналов	Меньше «звон» при наличии ложных сигналов
Доступны средства САПР	Доступны средства САПР
Децимация не влияет на эффективность	Децимация увеличивает эффективность

Если необходимы фильтры с крутым спадом и испытывается дефицит во времени, отведенном для обработки, хорошим выбором являются эллиптические БИХ фильтры. Если число операций умножения с накоплением не является чрезмерным и требуется линейная фаза, то должен быть выбран КИХ фильтр.

Расчет коэффициентов БИХ фильтра в математическом пакете MATLAB 7.0

Для расчета коэффициентов БИХ фильтра в MATLAB 7.0 предусмотрено множество функций, в том числе и специальная утилита Filter Design & Analysis Tool (FDA Tool), однако, функция `yulewalk(Nc, f, m)` позволяет это сделать наиболее просто.

Входные параметры:

N_c – Количество коэффициентов плюс 1;

f – Вектор точек частоты, изменяется от 0 до 1, где 1 это половина частоты дискретизации (0.5 нормализованной частоты);

m – Величина амплитуды в точках, указанных в векторе f , исходная амплитуда принимается равной единицы. Количество элементов m и f равны.

Выходные параметры – коэффициенты В и А.

Для построения импульсной характеристики используется функция **impz(B, A, N)**, где В и А – коэффициенты БИХ фильтра, а N – количество точек для построения.

Для построения АЧХ и ФЧХ фильтра используется функция **freqz(B, A, N, Fd)**, где В и А – коэффициенты БИХ фильтра, N – количество точек для построения, Fd – частота дискретизации.

На рисунке 1.4 показан пример расчет ФНЧ.

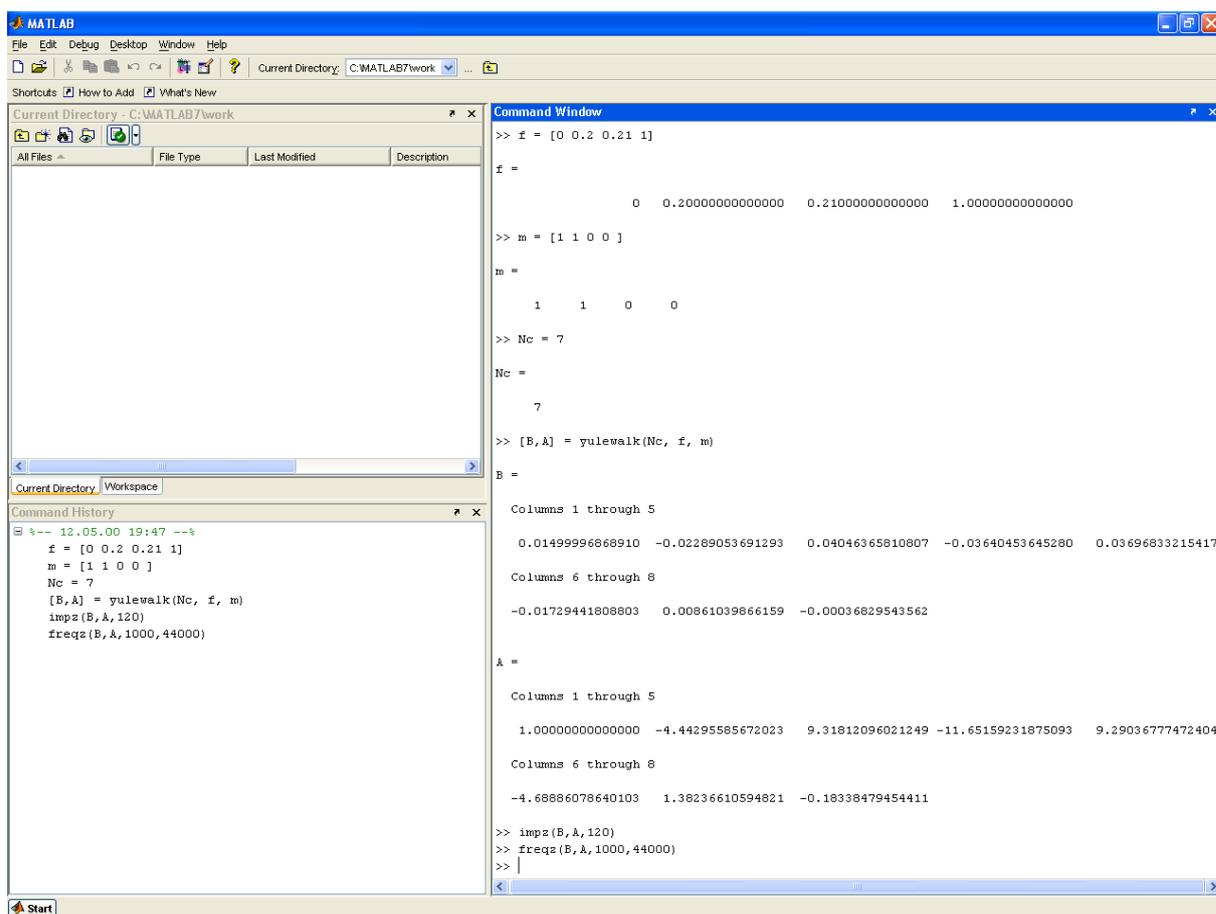


Рисунок 1.4 – Пример расчета ФНЧ фильтра в среде MATLAB.

1. Фильтр низкой частоты с нормальной частотой среза 0.1

Таблица 1.2 – коэффициенты ФНЧ.

f = [0 0.2 0.21 1], m = [1 1 0 0]				
A	Умноженное на 512	Округление до целого	Погрешность	Погрешность, %
1	512.0000000000	512	0.0000000000	0.0000000000

- 4.442955857	- 2274.7933986408	-2275	0.0004035183	0.0403518280
9.31812096	4770.8779316288	4771	0.0002384148	0.0238414788
- 11.65159232	- 5965.6152672005	-5966	0.0007514312	0.0751431249
9.290367775	4756.6683006587	4757	0.0006478503	0.0647850276
- 4.688860786	- 2400.6967226373	-2401	0.0005923386	0.0592338599
1.382366106	707.7714462455	708	0.0004463941	0.0446394052
- 0.183384795	-93.8930148066	-94	0.0002089555	0.0208955456
				0.3288902699
В	Умноженное на 512	Округление до целого	Погрешность	Погрешность, %
0.014999969	7.6799839688	8	0.0006250313	0.0625031311
- 0.022890537	-11.7199548994	-12	0.0005469631	0.0546963087
0.040463658	20.7173929513	21	0.0005519669	0.0551966892
- 0.036404536	-18.6391226638	-19	0.0007048385	0.0704838547
0.036968332	18.9277860629	19	0.0001410428	0.0141042846
- 0.017294418	-8.8547420611	-9	0.0002837069	0.0283706912
0.008610399	4.4085241147	4	0.0007978987	0.0797898662
- 0.000368295	-0.1885672630	0	0.0003682954	0.0368295436
				0.4019743692
Разность суммы погрешностей (В - А)				0.0730840993

2. Фильтр низкой частоты с нормальной частотой среза 0.2

Таблица 1.3 – коэффициенты ФНЧ.

$$f = [0 \ 0.4 \ 0.41 \ 1], \quad m = [1 \ 1 \ 0 \ 0]$$

A	Умноженное на 512	Округление до целого	Погрешность	Погрешность, %
1	512.0000000000	512	0.0000000000	0.0000000000
- 1.536825788	-786.8548032301	-787	0.0002835874	0.0283587441
2.56905269	1315.3549772448	1315	0.0006933149	0.0693314931
- 2.18828026	-1120.3994929432	-1120	0.0007802597	0.0780259655
1.669214529	854.6378389899	855	0.0007073457	0.0707345723
- 0.767060026	-392.7347335594	-393	0.0005180985	0.0518098517
0.254588948	130.3495414251	130	0.0006826981	0.0682698096
- 0.037346468	-19.1213916878	-19	0.0002370931	0.0237093140
				0.3902397503
B	Умноженное на 512	Округление до целого	Погрешность	Погрешность, %
0.055419464	28.3747655520	28	0.0007319640	0.0731963969
0.119305357	61.0843427943	61	0.0001647320	0.0164732020
0.190984545	97.7840867969	98	0.0004217055	0.0421705475
0.227660285	116.5620658969	117	0.0008553400	0.0855340045
0.185596674	95.0254971016	95	0.0000497990	0.0049799027
0.121731981	62.3267743792	62	0.0006382312	0.0638231209
0.049738973	25.4663540577	25	0.0009108478	0.0910847769
0.0131243	6.7196417292	7	0.0005475747	0.0547574748
				0.4320194261
Разность суммы погрешностей (B - A)				0.0417796759

3. Полосовой фильтр с нормальной частотой пропускания 0.165 – 0.33

Таблица 1.4 – коэффициенты полосового фильтра.

$f [0 \ 0.32 \ 0.33 \ 0.66 \ 0.67 \ 1] \ m = [0 \ 0 \ 1 \ 1 \ 0 \ 0]$

A	Умноженное на 512	Округление до целого	Погрешность	Погрешность, %
1	512.0000000000	512	0.0000000000	0.0000000000
0.017922482	9.1763109225	9	0.0003443573	0.0344357270
0.940927029	481.7546389739	482	0.0004792208	0.0479220754
0.010435385	5.3429172093	5	0.0006697602	0.0669760174
0.660088668	337.9653980685	338	0.0000675819	0.0067581897
0.0341823	17.5013374301	18	0.0009739503	0.0973950332
0.112911141	57.8105044096	58	0.0003701086	0.0370108575
0.005846488	2.9934020710	3	0.0000128866	0.0012886580
				0.2917865583
B	Умноженное на 512	Округление до целого	Погрешность	Погрешность, %
0.119059995	60.9587173885	61	0.0000806301	0.0080630101
0.012257812	6.2759996612	6	0.0005390618	0.0539061838
-				
0.181295087	-92.8230843331	-93	0.0003455384	0.0345538412
-				
0.025091566	-12.8468817547	-13	0.0002990591	0.0299059073
0.18151005	92.9331456229	93	0.0001305750	0.0130574955
0.030708454	15.7227282019	16	0.0005415465	0.0541546481
-				
0.119353206	-61.1088417217	-61	0.0002125815	0.0212581488
-				
0.017831151	-9.1295491884	-9	0.0002530258	0.0253025759
				0.2402018106
Разность суммы погрешностей (B - A)				0.0515847477