

2. ДИСКРЕТНЫЕ ФИЛЬТРЫ

2.1. Цели работы

- Пропускание дискретного сигнала через дискретный фильтр и получение графика выходного сигнала.
- Анализ сигналов, проходящих через элементы памяти фильтра, при реализации фильтра в разных формах.
- Получение графиков характеристик фильтра с помощью среды визуализации фильтров FVTool.
- Разложение функции передачи фильтра на простые дроби и получение аналитической формулы для импульсной характеристики фильтра.

2.2. Индивидуальное задание

В данной лабораторной работе используется сигнал из индивидуального задания для лабораторной работы № 1.

Этот сигнал в процессе выполнения работы пропускается через дискретный рекурсивный фильтр 4-го порядка, функция передачи которого имеет вид

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}. \quad (2.1)$$

Коэффициенты фильтра $b_0 \dots b_4$, $a_1 \dots a_4$ являются индивидуальными для каждой бригады и выдаются преподавателем в виде табл. 2.1.

Таблица 2.1

b_0	b_1	b_2	b_3	b_4	a_1	a_2	a_3	a_4

2.3. Указания к выполнению работы

1. Подготовка к началу работы

Запустите MATLAB и сделайте текущей папкой папку вашей бригады на сервере дисплейного класса.

Создайте в редакторе MATLAB новый файл MATLAB-программы и скопируйте в него часть кода из программы, созданной при выполнении лабораторной работы № 1. Необходимо выбрать фрагмент кода, с помощью которого формируется исходный дискретный сигнал и строится его график.

Сохраните новую программу с осмысленным именем и добейтесь ее работоспособности (программа должна формировать вектор отсчетов дискретного сигнала и строить его график).

Замечание. В данной работе при построении графиков сигналов для оцифровки горизонтальной оси целесообразно использовать номера отсчетов, вызывая функции **plot** и **stem** с одним аргументом: **plot(x)** и т. п.

2. Пропускание сигнала через дискретный фильтр

Чтобы увидеть затухающие переходные процессы, возникающие после завершения сигнала, добавьте к нему нулевые отсчеты в конце. Число нулевых отсчетов должно быть равно числу отсчетов исходного дискретного сигнала.

Справка. Функция **zeros(m, n)** возвращает заполненную нулями матрицу с **m** строками и **n** столбцами. Функция **length(x)** возвращает длину вектора **x**. Соединить две матрицы по горизонтали можно, записав их в квадратных скобках через пробел или запятую. Таким образом, если сигнал хранится в переменной **x** и представляет собой вектор-строку, дополнить его нулями можно так:

```
x0 = [x, zeros(1, length(x))];
```

Получите выходной сигнал фильтра, воспользовавшись для его реализации имеющейся в MATLAB готовой функцией **filter**.

Справка. Функция **filter** реализует обработку сигнала дискретным фильтром. Функция используется следующим образом:

```
y = filter(b, a, x);
```

Здесь **b** — вектор коэффициентов числителя функции передачи [**b0 b1 b2 b3 b4**], **a** — вектор коэффициентов знаменателя функции передачи [**1 a1 a2 a3 a4**] (см. формулу (2.1)), **x** — входной сигнал, **y** — выходной сигнал.

Внимание! Обратите внимание на то, что для фильтра 4-го порядка векторы коэффициентов содержат *пять* элементов, так как в них присутствуют коэффициенты в том числе для *нулевой* степени полиномов. Кроме того, не забудьте, что отсутствующий в индивидуальном задании коэффициент a_0 (он всегда равен единице, см. формулу (2.1)) при вызове функции **filter** должен быть включен в состав вектора параметров **a**.

Постройте график выходного сигнала с помощью функции **stem**. График должен демонстрировать искаженный фильтром сигнал, а также затухающий переходный процесс, происходящий после окончания сигнала на входе фильтра.

3. Анализ прямой формы реализации дискретного фильтра

Структурная схема прямой формы реализации дискретного фильтра показана на рис. 2.1.

При использовании прямой формы реализации дискретного фильтра в элементах памяти (элементах задержки) хранятся непосредственно отсчеты входного и выходного сигналов, поэтому все необходимые для анализа этой формы сигналы уже были получены при выполнении предыдущего пункта работы.

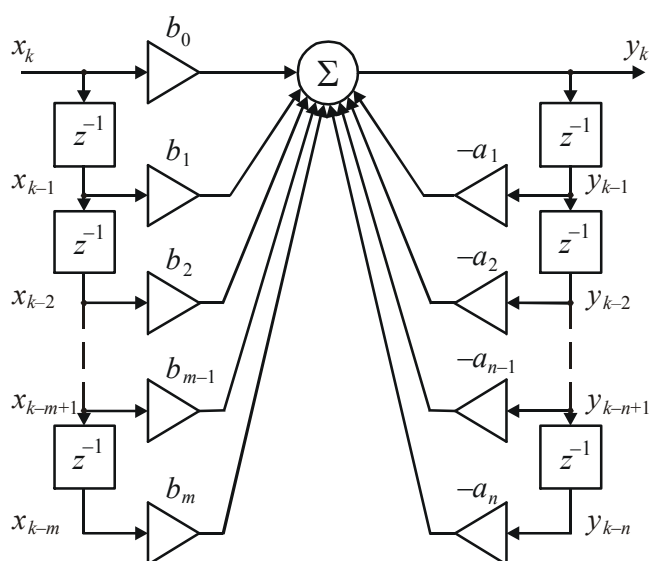


Рис. 2.1

Определите максимальное абсолютное значение сигналов, хранящихся в элементах памяти при работе фильтра (то есть выберите наибольший из максимумов модулей входного и выходного сигналов), и занесите его в табл. 2.2.

Таблица 2.2

Форма реализации фильтра	Макс. абс. внутреннее состояние
Прямая	
Каноническая	
Транспонированная	

Справка. Для определения максимального значения элемента вектора служит функция **max**:

xmax = max(x) ;

С помощью этой же функции можно выбрать максимум из двух чисел:

max_a_b = max(a, b) ;

Для вычисления абсолютных значений чисел служит функция **abs**.

4. Анализ канонической формы реализации дискретного фильтра

Структурная схема канонической формы реализации дискретного фильтра показана на рис. 2.2.

При использовании канонической формы реализации дискретного фильтра в элементах памяти (элементах задержки) хранятся отсчеты сигнала, прошедшего через рекурсивную часть фильтра, то есть через фильтр с функцией передачи следующего вида:

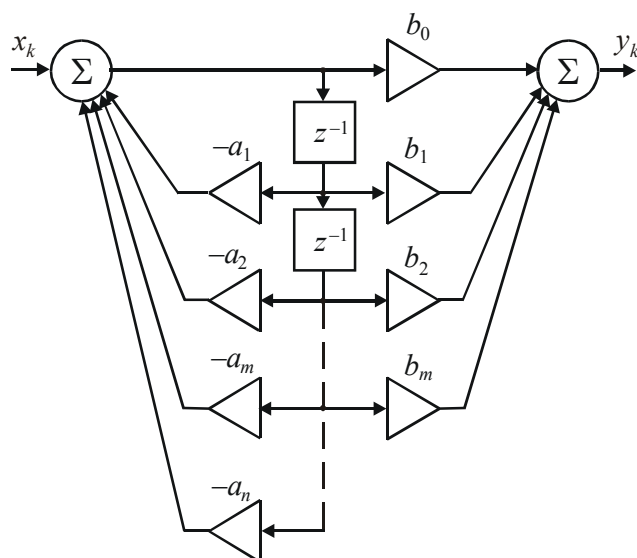


Рис. 2.2

$$H_1(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}. \quad (2.2)$$

Получите сигнал, хранящийся в элементах памяти фильтра, реализованного в соответствии с канонической формой, пропустив входной сигнал через фильтр с функцией передачи (2.2), используя для этого функцию **filter**.

Постройте график полученного сигнала с помощью функции **stem**.

Определите максимальное абсолютное значение полученного сигнала и занесите его в табл. 2.2.

5. Анализ транспонированной формы реализации дискретного фильтра

Структурная схема транспонированной формы реализации дискретного фильтра показана на рис. 2.3.

Из рис. 2.3 видно, что элементы памяти *не образуют линию задержки*, как это было в прямой и канонической формах. Поэтому в данном случае необходимо получить графики для *всех* сигналов, проходящих через элементы памяти фильтра.

Проще всего для этого воспользоваться тем фактом, что функция **filter** реализует обработку сигнала именно по схеме рис. 2.3, и, кроме того, она предоставляет возможность доступа к внутреннему состоянию фильтра.

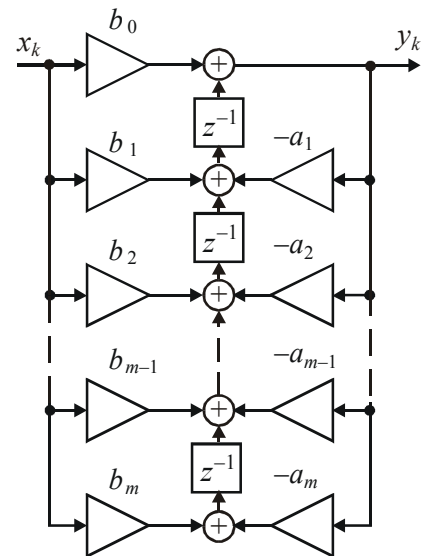


Рис. 2.3

Справка. Для доступа к внутреннему состоянию фильтра функция **filter** используется следующим образом:

```
[y, s_out] = filter(b, a, x, s_in);
```

Дополнительный входной параметр **s_in** задает начальное внутреннее состояние, а в дополнительном выходном результате **s_out** функция возвращает итоговое внутреннее состояние, достигнутое после обработки сигнала **x**.

Для получения всей истории изменения внутреннего состояния в процессе работы фильтра необходимо организовать цикл, в котором через фильтр будут пропускаться отдельные отсчеты сигнала. После обработки каждого отсчета внутреннее состояние фильтра будет сохраняться в виде очередного столбца отведенной для этого матрицы.

В приводимом ниже примере кода предполагается, что обрабатываемый сигнал хранится в векторе с именем **x**.

```
states = []; % заготовка матрицы внутренних состояний
s = [];      % текущее состояние фильтра
for k = 1:length(x)
    % обработка одного отсчета с заданием и сохранением
    % внутреннего состояния фильтра
    [y(k), s] = filter(b, a, x(k), s);
    % добавление нового столбца к матрице состояний
    states = [states s];
end
```

По окончании работы цикла вся история изменения внутреннего состояния фильтра будет записана в матрицу **states**. Разные строки этой матрицы соответствуют разным элементам памяти фильтра, разные столбцы — разным моментам времени.

Постройте в общих координатных осях графики всех внутренних сигналов фильтра с помощью функции **plot**.

Справка. Функция **plot** при передаче ей матричного параметра строит отдельные графики для всех *столбцов* матрицы. Поэтому для получения графиков всех внутренних сигналов фильтра матрицу **states** необходимо транспонировать: **plot(states')**.

Определите максимальное абсолютное значение внутреннего состояния, достигаемое при работе фильтра, и занесите его в табл. 2.2.

Справка. Функция **max** при передаче ей матричного аргумента находит максимум для каждого столбца матрицы отдельно. Чтобы найти значение, максимальное для всей матрицы, нужно вызвать функцию **max** два раза либо преобразовать матрицу в один столбец, указав двоеточие в качестве номера элемента:

A_max = max(max(A)) ;

A_max = max(A(:)) ;

6. Получение графиков характеристик фильтра

Вызовите среду визуализации характеристик фильтров (Filter Visualization Tool, FVTool), передав ей векторы коэффициентов фильтра аналогично тому, как это делается при вызове функции **filter** (приведенная ниже команда набирается в командном окне MATLAB):

fvtool(b, a)

Выбор просматриваемых характеристик и управление параметрами анализа производятся с помощью средств графического интерфейса окна, открывающегося после вызова данной функции.

Копирование графиков в буфер обмена производится с помощью команды меню **Edit ► Copy Figure** окна среды FVTool.

Получите и перенесите в документ Microsoft Word (в качестве заготовки для отчета) следующие графики:

- АЧХ (**Magnitude Response**)
- ФЧХ (**Phase Response**)
- Групповая задержка (**Group Delay Response**)
- Импульсная характеристика (**Impulse Response**)
- Расположение нулей и полюсов функции передачи на комплексной плоскости (**Pole/Zero Plot**)

7. Получение аналитической формулы для импульсной характеристики фильтра

Разложите функцию передачи фильтра на простые дроби с помощью функции **residuez**.

Справка. функция **residuez** используется следующим образом:

[r, p, k] = residuez(b, a)

Входные параметры: **b** и **a** — векторы коэффициентов фильтра (аналогично функции **filter**)

Выходные результаты: **r** — вектор вычетов, **p** — вектор соответствующих им полюсов, **k** — вектор коэффициентов целой части функции передачи. Функции передачи используемых в лабораторной работе фильтров имеют одинаковые степени полиномов числителя и знаменателя, поэтому вектор **k** содержит лишь один элемент.

После вызова функции **residuez** необходимо вычислить модули и фазы полюсов и вычетов (с помощью функций **abs** и **angle** соответственно) и занести эти результаты в табл. 2.3. В эту же таблицу нужно занести и значение целой части функции передачи **k**.

Таблица 2.3

	Модуль	Фаза, рад.
Вычеты		
r₁		
r₂		
r₃		
r₄		
Полюсы		
p₁		
p₂		
p₃		
p₄		
Целая часть		
k		

При подготовке отчета по лабораторной работе необходимо на основании данных из табл. 2.3 получить аналитическую формулу для импульсной характеристики фильтра. Процедура получения формулы описана ниже.

Используя результаты работы функции **residuez**, функцию передачи фильтра можно представить в следующем виде:

$$H(z) = k + \sum_{i=1}^4 \frac{r_i}{1 - p_i z^{-1}}. \quad (2.3)$$

Для получения импульсной характеристики фильтра необходимо произвести обратное z -преобразование функции (2.3). Для этого следует воспользоваться следующими свойствами:

- z -преобразование единичного импульса $\delta(k)$ равно константе: $X_0(z) = 1$;
- z -преобразование экспоненциального сигнала $x(k) = r p^k$, $k \geq 0$, равно

$$X(z) = \frac{r}{1 - pz^{-1}};$$

- z -преобразование линейно.

Поскольку рассматриваемые фильтры имеют вещественные коэффициенты, полюсы и вычеты функции передачи, а значит и экспоненциальные слагаемые импульсной характеристики, образуют комплексно-сопряженные пары. После получения импульсной характеристики в комплексной форме необходимо выполнить суммирование комплексно-сопряженных пар, приведя импульсную характеристику к *вещественному виду*.

8. Подготовка материалов для отчета

В завершение работы необходимо скопировать в документ Microsoft Word (в качестве заготовки для отчета) созданный программный код и все полученные графики (входной и выходной сигналы фильтра; внутренний сигнал фильтра, реализованного в канонической форме; внутренние сигналы фильтра, реализованного в транспонированной форме; графики характеристик фильтра, полученные в среде FVTool).

Справка. Для переноса графиков в документы Word следует использовать команду **Copy Figure** из меню **Edit** окна графика. Перед этим целесообразно задать векторный формат копирования, выбрав в том же меню команду **Copy Options** и установив переключатели **Preserve Information** и **Transparent Background**.

2.4. Содержание отчета

Отчет должен содержать созданный в процессе работы программный код, оформленный в виде законченного документа (с заголовками разделов, формулами и комментариями к коду). Полученные в ходе работы графики и заполненные таблицы размещаются в соответствующих разделах отчета.

Кроме того, в соответствующем разделе должен быть приведен вывод аналитической формулы для импульсной характеристики фильтра.

В конце отчета должны быть приведены выводы по результатам работы.