

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Чувашский государственный университет имени И.Н. Ульянова»

А.А. Андреева

# ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Практикум

Чебоксары  
2015

УДК 621.391(075.8)  
А65

*Рецензенты:*

кафедра управления в технических системах и программирования  
Чебоксарского политехнического института (филиала) ФГБОУ ВПО  
«Московский государственный машиностроительный университет  
(МАМИ)» (зав. кафедрой канд. техн. наук, профессор *О.Н. Зайцев*);  
директор по науке – заведующий отделом перспективных разработок  
ООО «НПП «Экра» канд. техн. наук *Н.А. Дони*

**Андреева А.А.**

**А65** Цифровая обработка сигналов: практикум / А.А. Андреева  
Чебоксары: Изд-во Чуваш ун-та, 2015. 130 с.

ISBN

Рассмотрены базовые операции цифровой обработки сигналов, проектирование и реализация цифровых фильтров, дискретные спектральные преобразования и быстрые алгоритмы их вычисления, спектральный анализ.

Для студентов IV курса направления «Информатика и вычислительная техника».

Работа выполнена при финансовой поддержке федеральной программы «Кадры для регионов».

Автор выражает благодарность студентам Артему Авивову, Ирине Дискантовой, Сергею Иванову и Николаю Леонтьеву за помощь в подготовке рукописи к изданию.

Ответственный редактор канд. техн. наук, доцент Л.А. Павлов

Утверждено Учебно-методическим советом университета

ISBN

УДК 621.391(075.8)  
© Издательство Чувашского  
университета, 2015  
Андреева А.А., 2015

## ПРЕДИСЛОВИЕ

Любая система цифровой обработки сигналов (ЦОС) содержит цифровое вычислительное устройство – универсальную ЭВМ, микропроцессор или специально разработанное для решения конкретной задачи. Поэтому сигнал должен быть преобразован к виду, пригодному для обработки на ЭВМ, т.е. должен быть представлен в цифровой форме. Дискретные по времени сигналы обычно получают при периодической выборке из непрерывного сигнала. Квантование по уровню дискретных сигналов дает цифровые сигналы. При малых ошибках квантования по уровню пренебрегают эффектами квантования, т.е. вместо цифровых сигналов рассматривают дискретные. Поэтому математической основой ЦОС является теория линейных дискретных систем. Ошибки квантования являются предметом специального исследования, например, при реализации цифровых фильтров выбирается структурная схема, дающая минимальные ошибки квантования.

В лабораторной работе 1 «Базовые операции ЦОС» изучаются способы описания систем ЦОС, основанные на теории линейных дискретных систем: разностное уравнение, передаточная функция, частотная и импульсная характеристики.

Лабораторная работа 2 посвящена проектированию, исследованию и программной реализации цифровых фильтров. Рассматривается синтез БИХ- и КИХ-фильтров, их построение в различных формах, реализация на языках программирования высокого и низкого уровня.

Работа 3 «Быстрое преобразование Фурье» знакомит студентов с алгоритмами БПФ: с прореживанием по времени и частоте (с основанием 2), а также с произвольным основанием (алгоритм Кули – Тьюки).

Завершает практикум лабораторная работа 4 «Цифровой спектральный анализ», в которой рассматриваются классические методы спектрального анализа, основанные на быстром преобразовании Фурье и оконных функциях.

Лабораторные работы выполняются на персональном компьютере с установленными операционной системой Windows, математической системой MATLAB (или ее свободно распространяемым аналогом Octave) и любой средой программирования на языке высокого уровня.

# ЛАБОРАТОРНАЯ РАБОТА 1

## Базовые операции цифровой обработки сигналов

### Теоретические сведения

#### Разностные уравнения

В задачах цифровой обработки сигналов объектом изучения являются сигналы и системы. Дискретизированные по времени сигналы возникают естественным путем или – и это достаточно распространенная ситуация – получаются в результате дискретизации аналоговых сигналов. Дискретные сигналы определяются лишь для дискретных значений независимой переменной – времени. Обычно время квантуется равномерно, т.е.  $t = nT$ , где  $T$  – интервал между отсчетами (период дискретизации). Математически дискретные сигналы представляются в виде последовательности чисел  $x(n)$  или  $x(nT)$ , где  $N_1 \leq n \leq N_2$ .

Дискретная система, по существу, является алгоритмом преобразования одной последовательности, называемой входной, в другую, называемую выходной (рис. 1.1).



Рис. 1.1. Представление дискретной системы

Пусть связь между входной  $x(n)$  и выходной  $y(n)$  последовательностями системы задается оператором  $\Phi$ :

$$y(n) = \Phi[x(n)].$$

Дискретная система называется линейной, если выполняются условия:

$$\Phi[\alpha \cdot x(n)] = \alpha \cdot \Phi[x(n)],$$

$$\Phi[x_1(n) + x_2(n)] = \Phi[x_1(n)] + \Phi[x_2(n)].$$

Наиболее общим способом представления линейной дискретной системы является линейное разностное уравнение  $N$ -го порядка с постоянными коэффициентами:

$$\sum_{i=0}^N a_i y(n-i) = \sum_{i=0}^N b_i x(n-i),$$

где  $a_N \neq 0$ .

Рассмотрим примеры получения таких уравнений.

*Пример 1*

Ссуда под недвижимость в сумме 2,5 млн рублей должна быть возвращена через 30 лет равными ежемесячными взносами размером  $p$  рублей. Выплачиваемый процент установлен на уровне 15% в год от невозвращенной суммы.

Пусть  $r = 0,15/12 = 0,0125$  – ежемесячная норма процента,  $P(n)$  – неоплаченная часть ссуды, оставшаяся после выплаты  $n$ -го ежемесячного взноса. Тогда

$$n \geq 0, P(0) = 2,50 \cdot 10^6, P(360) = 0,$$

$$P(n) = (1+r)P(n-1) - p,$$

$$P(n) - (1+r)P(n-1) = -p.$$

Получили уравнение первого порядка, в котором

$$a_0 = 1, b_0 = -1,$$

$$a_1 = -(1+r),$$

$$b_i = 0 (i \geq 1), a_i = 0 (i \geq 2),$$

$$0 \leq n \leq 360,$$

$$y(n) = P(n), x(n) = p.$$

*Пример 2*

Однородная цепь лестничного типа (рис. 1.2).

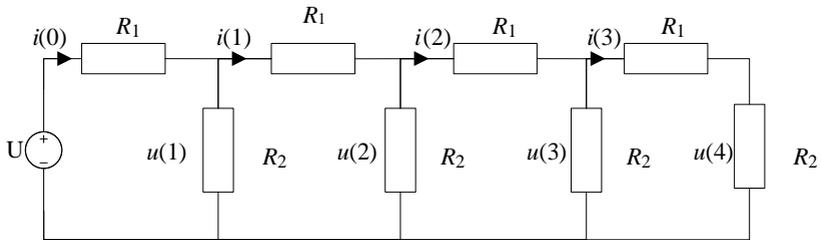


Рис. 1.2. Однородная лестничная цепь

В соответствии с законами Кирхгофа имеем

$$u(n-1) = u(n-2) - i(n-2)/R_1,$$

$$u(n) = u(n-1) - i(n-1)R_1,$$

$$i(n-1) = i(n-2) - u(n-1)/R_2.$$

Решая систему уравнений, получим однородное разностное уравнение второго порядка, описывающее распределение напряжений в цепи:

$$R_2 u(n) - (2R_2 + R_1)u(n-1) + R_2, \quad u(n-2) = 0.$$

При  $R_2 = R_1 = 1$  Ом уравнение принимает вид:

$$u(n) - 3u(n-1) + u(n-2) = 0.$$

### Прямое решение линейных разностных уравнений.

Пусть задано разностное уравнение, описывающее линейную дискретную систему:

$$y(n) = \sum_{i=0}^N b_i x(n-i) - \sum_{i=1}^N a_i y(n-i),$$

(считаем  $a_0 = 1$ ).

Тогда зная входную последовательность  $x(n), 0 \leq n \leq \infty$  и  $N$  начальных условий  $y(-N), \dots, y(-1)$ , можно найти  $y(0)$  и т.д.

В качестве примера рассмотрим решение задачи о ссуде, описываемой разностным уравнением

$$P(n) = (1+r)P(n-1) - p$$

при начальном условии

$$P(0) = 2,50 \cdot 10^6.$$

Тогда

$$P(1) = (1+r)P(0) - p,$$

$$P(2) = (1+r)((1+r)P(0) - p) - p = (1+r)^2 P(0) - (1+(1+r))p$$

$$P(3) = (1+r)^3 P(0) - (1+(1+r)+(1+r)^2)p,$$

$$P(n) = (1+r)^n P(0) - p(1+(1+r)+\dots+(1+r)^{n-1}) =$$

$$= (1+r)^n P(0) - p \frac{1-(1+r)^n}{1-1-r} = (1+r)^n P(0) - p \frac{(1+r)^n - 1}{r}.$$

Отсюда

$$p = \frac{((1+r)^n P(0) - P(n))r}{(1+r)^n - 1}.$$

Поскольку через 30 лет ( $n = 360$  месяцев) ссуда должна быть полностью выплачена, то

$$P(360) = 0, P(0) = 250 \cdot 10^3, r = 0,0125,$$

$$p = \frac{(1 + 0,0125)^{360} \cdot P(0) \cdot 0,0125}{(1 + 0,0125)^{360} - 1} \approx 0,012644P(0) \approx 31,6 \cdot 10^3.$$

Сумма возврата  $360 p \approx 11,38 \cdot 10^6$ .

К сожалению, не всегда начальные условия бывают заданы в удобной форме. Не всегда легко получить выражение для  $y(n)$  в замкнутой форме.

**Реакция при нулевом входном воздействии (решение однородных разностных уравнений).** Рассмотрим на примере однородной лестничной цепи (см. рис. 1.2), которая описывается разностным уравнением

$$u(n) - 3u(n-1) + u(n-2) = 0$$

при следующих начальных условиях:

$$u(0) = U,$$

$$u(n) \rightarrow 0 \text{ при } n \rightarrow \infty.$$

Требуется найти входное сопротивление цепи и распределение напряжений по ветвям.

Прямой путь решения не пригоден.

Известно, что однородные линейные разностные уравнения с постоянными коэффициентами имеют решение в форме

$$u(n) = Az^n,$$

где  $z$  – некоторое подходящее (в общем случае комплексное) число, т.е. дискретные экспоненты.

Подставим  $u(n) = Az^n$  в линейное уравнение

$$Az^n - 3Az^{n-1} + Az^{n-2} = 0,$$

$$Az^{n-2}(z^2 - 3z + 1) = 0.$$

Если  $u(n) = Az^n \neq 0$ , то  $z$  должны быть корнями уравнения

$$z^2 - 3z + 1 = 0, \text{ т.е.}$$

$$z = \frac{3 \pm \sqrt{9-4}}{2} = \frac{3 \pm \sqrt{5}}{2}.$$

Тогда

$$u(n) = A \left( \frac{3 - \sqrt{5}}{2} \right)^n + B \left( \frac{3 + \sqrt{5}}{2} \right)^n,$$

где  $A, B$  – некоторые числа.

$$\left( \frac{3 - \sqrt{5}}{2} \right)^n \rightarrow 0 \text{ при } n \rightarrow \infty,$$

$$\left( \frac{3 + \sqrt{5}}{2} \right)^n \rightarrow \infty \text{ при } n \rightarrow \infty,$$

поэтому  $B = 0$  ( $u(n) \rightarrow 0$  при  $n \rightarrow \infty$ ).

Из условия  $u(0) = U$  находим  $A = U$ .

$$\text{Окончательно } u(n) = U \left( \frac{3 - \sqrt{5}}{2} \right)^n.$$

Входное сопротивление цепи

$$R_{\text{вх}} = \frac{U}{i(0)}.$$

Поскольку

$$i(0) = \frac{u(0) - u(1)}{R} = U \left( 1 - \frac{3 - \sqrt{5}}{2} \right),$$

$$R_{\text{вх}} = \frac{2}{2 - 3 + \sqrt{5}} = \frac{2}{\sqrt{5} - 1} \approx 1,62 \text{ Ом}.$$

*Вывод:* реакция при нулевом воздействии системы, описываемой линейным разностным уравнением с постоянными коэффициентами

$$\sum_{i=0}^N a_i y(n-i) = \sum_{i=0}^N b_i x(n-i)$$

является решением однородного уравнения

$$\sum_{i=0}^N a_i y(n-i) = 0.$$

Это решение имеет вид

$$y(n) = \sum_{k=1}^N A_k z_k^n,$$

где  $\{z_k\}$  – совокупность корней характеристического уравнения

$$\sum_{k=0}^N a_{N-k} z^k = 0$$

и  $\{A_k\}$  – совокупность  $N$  постоянных, выбранных в соответствии с начальными условиями (если характеристическое уравнение имеет кратные корни, то должны быть добавлены в решение члены вида  $n^p z_k^n$ ).

Однородные решения описывают основные свойства системы.

Общее решение является суммой однородного и частного. Частное решение получают, подбирая вид последовательности  $y(n)$  при заданной входной последовательности  $x(n)$ , обычно той же формы, что и  $x(n)$ , согласовывая коэффициенты.

Более удобно решать разностные уравнения методом  $z$ -преобразования.

### **Z-преобразование**

**Определение  $z$ -преобразования.** Одностороннее  $z$ -преобразование последовательности  $x(n)$  определяется формулой

$$X(z) = \sum_{n=0}^{\infty} x(n) z^{-n}.$$

Если  $|x(n)|$  возрастает не быстрее, чем экспонента, то этот ряд будет сходиться для всех  $z$ , находящихся вне некоторого круга на комплексной  $z$ -плоскости (рис. 1.3), радиус которого  $r_0$  называется радиусом сходимости.

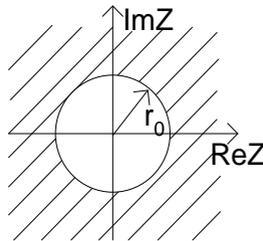


Рис 1.3. Область сходимости  $z$ -преобразования

Полезность  $z$ -преобразования базируется на взаимнооднозначном соответствии между  $X(z)$  и последовательностью  $x(n)$ : каждой последовательности  $x(n)$  определенной для  $n \geq 0$ , соот-

ответствует *одно и только одно* преобразование  $X(z)$ , определенное для  $|z| > r_0$ . Эта теорема единственности является интерпретацией центральной теоремы единственности и сходимости степенных рядов аналитических функций комплексной переменной.

*Пример 1. Экспонента*

$$x(n) = Aa^n, n \geq 0.$$

Применим формулу суммы членов убывающей геометрической прогрессии

$$S = \frac{a_1}{1 - q}.$$

Получаем

$$X(z) = A \sum_{n=0}^{\infty} a^n z^{-n} = \frac{A}{1 - a^{-1}} = \frac{Az}{z - a}.$$

Условие сходимости:  $\left| \frac{a}{z} \right| < 1, |z| > |a|$ . Полнос (корень знаменателя)  $z = a$ , лежит на окружности. Нуль (корень числителя)  $z = 0$  (рис. 1.4).

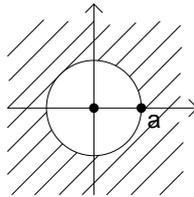


Рис. 1.4. Область сходимости, нуль и полюс  $z$ -преобразования последовательности  $x(n) = Aa^n$

*Пример 2. Единичная ступенчатая функция*

$$x(n) = u(n) = \begin{cases} 1, n \geq 0, \\ 0, n < 0. \end{cases}$$

Тогда

$$X(z) = \frac{1}{1 - z^{-1}}.$$

Преобразование имеет нуль при  $z = 0$ , полюс при  $z = 1$ .

*Пример 3. Функция единичного отсчета*

$$x(n) = \delta(n) = \begin{cases} 1, n = 0 \\ 0, n \neq 0 \end{cases}$$

$$X(z) = 1.$$

В системе MATLAB прямое  $z$ -преобразование выполняет функция `ztrans`:

`ztrans(F, trans_index, eval_point),`

где  $F$  – выражение, вектор, функция или матрица; `trans_index` – дискретная переменная; `eval_point` – частота.

Примеры использования функции `ztrans`.

*Пример 1*

В результате выполнения команд

```
>>syms n, ztrans(n^2)
```

получим ответ:

```
ans =z*(z+1)/(z-1)^3,
```

а в результате выполнения

```
>>syms a n, ztrans(sin(a*n)),
```

получим ответ:

```
ans =z*sin(a)/(z^2-2*z*cos(a)+1) .
```

Тем самым получили два  $z$ -преобразования:

$$n^2 \leftrightarrow z(z+1)/(z-1)^3,$$

$$\sin an \leftrightarrow z \sin a / (z^2 - 2z \cos a + 1).$$

*Пример 2*

Функция `X=ztrans(x)` вычисляет прямое  $z$ -преобразование символьной функции  $x$ .

```
>> syms X x r w0 n
```

```
>> x=r^n*(cos(w0*n))
```

```
x =
```

```
r^n*cos(n*w0)
```

```
>> X=ztrans(x)
```

```
X=- (z*(cos(w0)-z/r))/(r*(z^2/r^2-  
(2*z*cos(w0))/r+1))
```

**Обратное  $z$ -преобразование** рассмотрим на примере. Пусть задано  $z$ -преобразование

$$X(z) = \frac{30z^2}{6z^2 - z - 1}, \quad |z| > \frac{1}{2}.$$

По теореме единственности ему должна соответствовать единственная последовательность  $x(n)$ ,  $n \geq 0$ . Как выполнить обратное  $z$ -преобразование?

*1-й способ*

Разложить  $X(z)$  в степенной ряд по степеням  $z^{-i}$ , например, делением числителя на знаменатель, которые записываются по убывающим степеням  $z$ .

Перепишем  $X(z)$  в виде

$$X(z) = \frac{5}{1 - \frac{1}{6}z^{-1} - \frac{1}{6}z^{-2}}.$$

Разделим 5 на  $1 - \frac{1}{6}z^{-1} - \frac{1}{6}z^{-2}$  столбиком, в результате получим

$$x(0) = 5, x(1) = \frac{5}{6},$$

$$x(2) = \frac{35}{36}, x(3) = \frac{65}{216} \text{ и т.д.}$$

*2-й способ* – разложение на сумму простейших дробей вида

$$\frac{1}{1 - az^{-1}} \Rightarrow a^n.$$

Пусть

$$X(z) = \frac{60z^2}{6z^2 - z - 1}.$$

Тогда

$$X(z) = \frac{60z^2}{6\left(z - \frac{1}{2}\right)\left(z + \frac{1}{3}\right)} =$$

$$= \frac{5}{\left(1 - \frac{1}{2}z^{-1}\right)\left(1 + \frac{1}{3}z^{-1}\right)} =$$

$$= \frac{k_1}{1 - \frac{1}{2}z^{-1}} + \frac{k_2}{1 + \frac{1}{3}z^{-1}},$$

$$k_1 = X(z) \left(1 - \frac{1}{2}z^{-1}\right) \Bigg|_{z = \frac{1}{2}} = \frac{5}{1 + \frac{1}{3} \cdot 2} = 3,$$

$$k_2 = X(z) \left(1 + \frac{1}{3}z^{-1}\right) \Bigg|_{z = -\frac{1}{3}} = \frac{5}{1 + \frac{1}{2} \cdot 3} = 2.$$

Поскольку  $z$ -преобразование – линейная операция, то на основании теоремы единственности

$$x(n) = 3\left(\frac{1}{2}\right)^n + 2\left(-\frac{1}{3}\right)^n, \quad n \geq 0.$$

В системе MATLAB обратное  $z$ -преобразование выполняет функция `iztrans`:

`iztrans(F, trans_index, eval_point)`,

где  $F$  – выражение, вектор, функция или матрица; `trans_index` – дискретная переменная; `eval_point` – частота.

### Свойства $z$ -преобразования (теоремы):

1. Суперпозиция (линейность):

$$ax(n) + by(n) \Leftrightarrow aX(z) + bY(z).$$

2. Умножение на экспоненту:

$$a^n x(n) \Leftrightarrow X(a^{-1}z).$$

3. Умножение на  $n$ :

$$nx(n) \Leftrightarrow -z \frac{dX(z)}{dz}.$$

4. Задержка на  $N \geq 0$ :

$$x(n-N)u(n-N) \Leftrightarrow z^{-N} X(z),$$

$$\text{здесь } u(n) = \begin{cases} 1, n \geq 0, \\ 0, n < 0. \end{cases}$$

В общем случае

$$x(n-N) \Leftrightarrow x(-N) + x(-N+1)z^{-1} + \dots + x(-1)z^{-(N-1)} + z^{-N} X(z).$$

Свойства  $z$ -преобразования доказывают, исходя из его определения.

*Пример* применения свойств  $z$ -преобразования. Найти  $z$ -преобразование последовательности  $x(n) = na^n, n \geq 0$ .

Известно, что  $a^n \Leftrightarrow \frac{1}{1 - az^{-1}}$ .

Тогда по свойству умножения на  $n$

$$\begin{aligned} na^n &\Leftrightarrow z \frac{d}{dz} (1 - az^{-1})^{-1} = \\ &= z \left\{ -\frac{1}{(1 - az^{-1})^2} \cdot (-a)z^{-2} \right\} = \frac{az^{-1}}{(1 - az^{-1})^2}. \end{aligned}$$

**Применение  $z$ -преобразования к линейным дискретным системам** основано на теореме задержки на  $N$  отсчетов при нулевых начальных условиях:

$$\begin{aligned} x(n - N) &\Leftrightarrow x(-N) + x(-N + 1)z^{-1} + \dots + x(-1)z^{-(N-1)} + z^{-N} X(z) = \\ &= z^{-N} X(z). \end{aligned}$$

Применим  $z$ -преобразование к разностному уравнению

$$y(n) + \sum_{i=1}^N a_i y(n-i) = \sum_{i=0}^N b_i x(n-i)$$

при нулевых начальных условиях. Получим

$$Y(z) \left( 1 + \sum_{k=1}^N a_k z^{-k} \right) = X(z) \sum_{k=0}^N b_k z^{-k},$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}.$$

Отношение  $H(z)$   $z$ -преобразования выходной последовательности к  $z$ -преобразованию входной последовательности при нулевых начальных условиях называют передаточной функцией линейной дискретной системы.

**Реакция на единичный отсчет**

Пусть

$$x(n) = \delta(n) = \begin{cases} 1, & n = 0, \\ 0, & n \neq 0. \end{cases}$$

$Z$ -преобразование

$$\sum_{n=0}^{\infty} \delta(n)z^{-n} = 1.$$

Тогда  $z$ -преобразование реакции на единичный отсчет при нулевых начальных условиях

$$H(z) \cdot 1 = H(z).$$

Реакция на единичный отсчет  $h(n)$  (импульсная характеристика системы) и передаточная функция  $H(z)$  представляют собой пару  $z$ -преобразования

$$h(n) \Leftrightarrow H(z).$$

При заданной  $h(n)$  можно найти реакцию на произвольное входное воздействие  $x(n)$  при нулевых начальных условиях, преобразуя

$$h(n) \Leftrightarrow H(z),$$

$$x(n) \Leftrightarrow X(z)$$

и выполняя обратное  $z$ -преобразование от

$$H(z)X(z).$$

В системе MATLAB вектор выходных значений  $y$  можно найти с использованием встроенной функции `filter`, если известны значения  $b$  и  $a$  – коэффициентов разностного уравнения и вектор входных значений  $x$ :

$$y = \text{filter}(b, a, x).$$

Импульсная характеристика вычисляется и строится функцией `impz`:

$$\text{impz}(b, a);$$

$$\text{impz}(b, a, n);$$

где  $b, a$  – коэффициенты разностного уравнения,  $n$  – количество точек характеристики. График строится при помощи ступенек.

Импульсная характеристика может быть вычислена также при помощи функции `filter`:

$$i = \text{eye}(1, 100); \quad \% \text{Формируем вектор, у которого}$$

$$\% \text{первый отсчет - единица,}$$

$$\% a \text{ остальные 99 отсчетов - нули.}$$

$$h = \text{filter}(b, a, i); \quad \% \text{Вычисляем импульсную}$$

$$\% \text{характеристику.}$$

В этом примере знак % используется для введения комментария, а знак “;” (точка с запятой) отключает вывод переменных на экран монитора.

### Операция свертки

**Теорема свертки и z-преобразование.** Рассмотрим теорему. Пусть

$$y(n) = \sum_{m=0}^n x(m)h(n-m) -$$

свертка  $x(n)$  и  $h(n)$ . Тогда z-преобразование  $y(n)$  имеет вид

$$Y(z) = X(z) \cdot H(z),$$

где  $X(z)$  и  $H(z)$  есть преобразование  $x(n)$  и  $h(n)$  соответственно.

Для доказательства оценим  $Y(z)$  непосредственно:

$$Y(z) = \sum_{n=0}^{\infty} y(n)z^{-n} = \sum_{n=0}^{\infty} \left( \sum_{m=0}^n x(m)h(n-m) \right) z^{-n}.$$

Поскольку

$$h(n-m) = 0, m > n,$$

$$h(n) = 0, n < 0,$$

$$x(n) = 0, n < 0,$$

и

$$z^{-n} = z^{-n+m-m} = z^{-m} \cdot z^{-(n-m)},$$

то

$$Y(z) = \sum_{n=0}^{\infty} \left( \sum_{m=0}^n x(m)h(n-m)u(n-m) \right) z^{-m} z^{-(n-m)}.$$

Изменяем порядок суммирования:

$$\begin{aligned} Y(z) &= \sum_{m=0}^n \sum_{n=0}^{\infty} x(m)h(n-m)u(n-m)z^{-m}z^{-(n-m)} = \\ &= \sum_{m=0}^{\infty} x(m)z^{-m} \sum_{n=0}^{\infty} h(n-m)u(n-m)z^{-(n-m)} = \\ &= X(z)H(z), \text{ что и требовалось доказать.} \end{aligned}$$

*Пример*

Пусть система описывается уравнением

$$y(n) - y(n-1) = x(n-1).$$

При нулевых начальных условиях

$$Y(z) - z^{-1}Y(z) = z^{-1}X(z),$$

$$Y(z)(1 - z^{-1}) = z^{-1}X(z).$$

Передающая функция

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-1}}{1 - z^{-1}}.$$

Импульсная характеристика

$$h(n) = u(n-1).$$

Реакция на произвольное воздействие

$$y(n) = \sum_{m=0}^n x(m)h(n-m) = \sum_{m=0}^n x(m)u(n-1-m).$$

Если учесть, что

$$u(n-1-m) = \begin{cases} 1, & n-1-m \geq 0, \quad m \leq n-1, \\ 0, & n-1-m < 0, \quad m > n-1, \end{cases}$$

то

$$y(n) = \begin{cases} \sum_{m=0}^{n-1} x(m), & n \geq 1, \\ 0, & n = 0. \end{cases}$$

Дискретная система представляет собой накопитель – аналог интегратора в непрерывном времени.

**Свертка и линейные инвариантные во времени системы общего типа.** Формула

$$y(n) = \sum_{m=0}^n x(m)h(n-m)$$

представляет собой частный случай более общей формулы свертки

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m).$$

Общая формула описывает линейную и инвариантную во времени операцию над  $x(n)$  удовлетворяя двум условиям:

1) линейности (суперпозиции): если  $x_1(n)$  и  $x_2(n)$  – какие-либо два произвольных входных воздействия – в результате не-

которой операции над ними дают на выходе соответственно  $y_1(n)$  и  $y_2(n)$ ,

$$x_1(n) \rightarrow y_1(n),$$

$$x_2(n) \rightarrow y_2(n),$$

то говорят, что операция линейна, если выполняется условие

$$x(n) = ax_1(n) + bx_2(n) \rightarrow y(n) = ay_1(n) + by_2(n)$$

при всех постоянных  $a$  и  $b$ ;

2) инвариантности во времени. Если  $x(n)$  – какое-либо произвольное воздействие в результате некоторой операции дает на выходе хорошо определенную последовательность (т.е. в свертке бесконечная сумма сходится)  $y(n)$ , так что

$$x(n) \rightarrow y(n),$$

то говорят, что операция инвариантна во времени, если выполняется условие  $x(n-N) \rightarrow y(n-N)$

для всех  $-\infty < N < \infty$ .

Общая формула свертки представляет собой сумму бесконечного числа членов. Как от общей формулы перейти к частной?

Наложим два условия:

1) система физически реализуема (т.е. казуальна)

$$h(n) \equiv 0, n < 0$$

(отклик не может быть раньше стимулирующего воздействия);

2)  $x(n) \equiv 0, n < 0$ .

Тогда слагаемое

$$x(m)h(n-m) = 0, \text{ при } m < 0 \text{ и при } m > n.$$

Это значит, что при конечном  $n$  сумма содержит конечное число членов и, стало быть, конечна, если  $x(n)$  и  $h(n)$  последовательности конечной амплитуды.

Частный случай свертки соответствует системам управления. Общий – задачам связи или обработки данных.

От казуальности можно отказаться в случаях, когда:

- а)  $n$  является не текущим временем, а, например, пространством;
- б) операции выполняются компьютером не в реальном времени.

Главное различие в поведении казуальной и неказуальной, но устойчивой системы – лишь небольшая задержка, которая для задач не управления не имеет значения.

Результат в общей формуле будет конечным, т.е. существуют и другие условия, налагаемые на  $h(n)$  и  $x(n)$ , которые могут гарантировать существование  $y(n)$  :

1)  $h(n)$  – абсолютно суммируема, т.е.

$$\sum_{n=-\infty}^{\infty} |h(n)| = H_0 < \infty ;$$

2)  $x(n)$  – ограничена по величине,

$$|x(n)| \leq X_{\max} < \infty$$

В этом случае  $y(n)$  ограничена, так как

$$\begin{aligned} |y(n)| &= \sum_{n=-\infty}^{\infty} |x(m)h(n-m)| = \sum_{n=-\infty}^{\infty} |x(m)||h(n-m)| = \\ &= X_{\max} \sum_{n=-\infty}^{\infty} |h(n-m)| = X_{\max} H_0 < \infty. \end{aligned}$$

Ограниченное воздействие порождает ограниченный выход – условие устойчивости, необходимое и достаточное.

**Свойства общей операции свертки.** Введем условное обозначение для формулы свертки

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)$$

в виде

$$y(n) = x(n) * h(n).$$

Тогда свойства общей операции свертки, справедливые и для частного случая, можно записать так:

1) закон коммутативности:

$$x(n) * h(n) = h(n) * x(n), \quad \sum_{m=-\infty}^{\infty} x(m)h(n-m) = \sum_{m=-\infty}^{\infty} h(m)x(n-m);$$

2) закон ассоциативности:

$$x(n) * (h_1(n) * h_2(n)) = ((x(n) * h_1(n)) * h_2(n));$$

3) закон дистрибутивности:

$$(x_1(n) + x_2(n)) * h(n) = x_1(n) * h(n) + x_2(n) * h(n).$$

В системе MATLAB для вычисления свертки двух последовательностей конечной длины используется функция conv:

```
c=conv(a,b).
```

Пусть на входе последовательности  $a = [1\ 2\ 3]$  и  $b = [4\ 5\ 6]$ , вычисляем

```
c=conv(a,b);
```

```
c=
```

```
4 13 28 27 18
```

### Частотная характеристика линейной дискретной системы

Мы рассмотрели описание линейных дискретных систем с помощью: 1) разностных уравнений, 2) передаточных функций  $H(z)$ , 3) импульсных характеристик  $h(n)$ .

Для чего необходима частотная характеристика? Системы ЦОС часто выполняют функции фильтрации. Для описания соответствующих аналоговых фильтров используется частотная характеристика (фильтр верхних частот, нижних частот, полосовой, режекторный). Поскольку система ЦОС должна выполнять то же, что и аналоговый фильтр, то для такой системы следует определить частотную характеристику.

**Определение частотной характеристики.** Рассмотрим класс входных последовательностей вида

$$x(n) = e^{jkn} = \cos \omega n + j \sin \omega n, n \geq 0.$$

Пусть  $x(n)$  поступает на вход линейной дискретной системы с импульсной характеристикой  $h(n)$ , тогда

$$\begin{aligned} y(n) &= \sum_{m=0}^{\infty} h(m)x(n-m) = \sum_{m=0}^{\infty} h(m)e^{j\omega(n-m)} = \\ &= e^{j\omega n} \sum_{m=0}^{\infty} h(m)e^{-j\omega m} = e^{j\omega n} H(e^{j\omega}) = x(n)H(e^{j\omega}). \end{aligned}$$

$H(e^{j\omega})$  – частотная характеристика системы

$$H(e^{j\omega}) = \sum_{m=0}^{\infty} h(m)e^{-j\omega m}, \quad H(e^{j\omega}) \Leftrightarrow H(z).$$

Модуль частотной характеристики  $|H(e^{j\omega})|$  называется амплитудно-частотной характеристикой (АЧХ), аргумент  $\arg H(e^{j\omega})$  – фазочастотной характеристикой (ФЧХ), т.е.

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{j \arg H(e^{j\omega})}.$$

Производная от фазочастотной характеристики, взятая с отрицательным знаком, называется групповым временем задержки (групповой задержкой):

$$\tau(\omega) = - \frac{d \arg H(e^{j\omega})}{d\omega}.$$

**Свойства частотной характеристики:**

1) периодичность, причем ее период равен  $2\pi$ :

$$x(n + 2\pi m) = e^{j\omega(n+2\pi m)} = e^{j\omega n} e^{j\omega 2\pi m} = e^{j\omega n} = x(n),$$

поэтому частотную характеристику рассматривают на интервале  $0 \leq \omega \leq 2\pi$ .

2) если  $h(n)$  – действительна, то  $|H(e^{j\omega})|$  – симметричен относительно оси  $\omega = \pi$ ,  $\arg H(e^{j\omega})$  антисимметричен.

Поэтому для действительных  $h(n)$  частотную характеристику рассматривают на интервале  $0 \leq \omega \leq \pi$ .

Если необходимо выразить частотную характеристику в единицах частоты, связанных с интервалом дискретизации  $T$ , то

$$H(e^{j\omega T}) = \sum_{n=0}^{\infty} h(nT) \cdot e^{-j\omega n T},$$

$$\text{период } H(e^{j\omega T}) = \frac{2\pi}{T}.$$

*Пример*

Частотная характеристика системы 1-го порядка, описываемой разностным уравнением

$$y(n) = x(n) + ky(n-1)$$

с нулевым начальным условием  $y(n-1) = 0$ .

Тогда переходя к  $z$ -форме, получим

$$Y(z) - kz^{-1}Y(z) = X(z).$$

Передающая функция

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - kz^{-1}}.$$

Импульсная характеристика

$$h(n) = k^n, n \geq 0.$$

Частотная характеристика

$$H(e^{j\omega}) = \frac{1}{1 - ke^{-j\omega}}.$$

Чтобы получить АЧХ и ФЧХ, выполним преобразования (преобразуем комплексное число в знаменателе из экспоненциальной формы в тригонометрическую, далее избавимся от комплексных чисел в знаменателе путем умножения числителя и знаменателя число, комплексно-сопряженное знаменателю):

$$\begin{aligned} H(e^{j\omega}) &= \frac{1}{1 - k(\cos \omega - j \sin \omega)} = \frac{1 - k \cos \omega - jk \sin \omega}{(1 - k \cos \omega)^2 + k^2 \sin^2 \omega} = \\ &= \frac{1 - k \cos \omega - jk \sin \omega}{1 - 2k \cos \omega + k^2 \cos^2 \omega + k^2 \sin^2 \omega} = \\ &= \frac{1 - k \cos \omega - jk \sin \omega}{1 + k^2 - 2k \cos \omega}. \end{aligned}$$

Амплитудно-частотная характеристика

$$\begin{aligned} |H(e^{j\omega})| &= \frac{\sqrt{(1 - k \cos \omega)^2 + k^2 \sin^2 \omega}}{(1 - k \cos \omega)^2 + k^2 \sin^2 \omega} = \\ &= \frac{1}{\sqrt{(1 - k \cos \omega)^2 + k^2 \sin^2 \omega}} = \frac{1}{\sqrt{1 + k^2 - 2k \cos \omega}}. \end{aligned}$$

Фазочастотная характеристика

$$\arg H(e^{j\omega}) = \arctg \frac{k \sin \omega}{k \cos \omega - 1}.$$

Отметим, что для дискретных систем

$$Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega}),$$

т.е. свертка во временной области соответствует умножению в частотной области.

Для построения и вычисления комплексной частотной характеристики используется функция `freqz`:

$$H = \text{freqz}(b, a, w);$$

$$[H, w] = \text{freqz}(b, a);$$

где  $b$  и  $a$  – коэффициенты передаточной функции дискретной системы,  $w$  – вектор частот, каждая компонента которого соответствует определенному значению  $H$ .

Для нахождения амплитудно-частотной характеристики (АЧХ) достаточно взять абсолютное значение от комплексной частотной характеристики:

$$m = \text{abs}(H);$$

вектор  $m$  будет содержать значения АЧХ.

Логарифмическая АЧХ (ЛАЧХ) определяется как

$$L = 20 * \log_{10}(m).$$

Для нахождения фазочастотной характеристики (ФЧХ) нужно использовать выражение:

$$p = \text{unwrap}(\text{angle}(H)).$$

Сначала при помощи функции `angle` находится фазовый угол, затем при помощи функции `unwrap` добавляется к нему для коррекции  $\pm 2\pi$ .

Групповая задержка вычисляется при помощи функции `grpdelay`:

```
[gd, w] = grpdelay(b, a);  
[gd, f] = grpdelay(b, a, n, Fs);  
gd = grpdelay(b, a, w);  
gd = grpdelay(b, a, f, Fs);
```

где  $w$  и  $f$  – частоты, измеряемые в  $c^{-1}$  и герцах соответственно;  $n$  – количество отсчетов;  $F_s$  – частота дискретизации;  $b$  и  $a$  – коэффициенты передаточной функции дискретной системы.

Приведем текст программы для вычисления и построения частотных характеристик системы с передаточной функцией  $H(z) = 1/(1 - 1/6z^{-1} - 1/6z^{-2})$ :

```
Fd=1000; %частота дискретизации  
b=[1]; %коэффициенты числителя передаточной функции  
a=[1 -1/6 -1/6]; %коэффициенты знаменателя  
%передаточной функции  
[H, w]=freqz(b,a); %вычисление комплексной  
%частотной характеристики  
f=w/pi*Fd; %пересчет масштаба частот  
A=abs(H); %вычисление АЧХ  
Ph=unwrap(angle(H)); %вычисление ФЧХ  
plot(f,A);pause %построение АЧХ
```

plot(f, Ph); %построение ФЧХ

### Задание к лабораторной работе

Выполнить теоретически и в системе MATLAB вариант задания, соответствующий порядковому номеру студента в списке группы. Варианты заданий приведены ниже.

1-4. Построить частотные характеристики линейной дискретной системы (АЧХ, ФЧХ, АФХ, ЛАЧХ, характеристику групповой задержки) с заданной передаточной функцией (табл. 1.1).

5-8. По заданной импульсной характеристике линейной дискретной системы (табл. 1.1) определить передаточную функцию, отобразить расположение нулей и полюсов на комплексной плоскости. Построить частотные характеристики цифрового фильтра.

9-12. По заданной передаточной функции линейной дискретной системы (табл. 1.1) определить и построить ее импульсную и временную характеристики.

13-14. При помощи графического курсора (функция `ginput` системы MATLAB), определяя произвольно местонахождение полюсов и нулей на комплексной плоскости, найти передаточную функцию цифрового фильтра, используя функцию `zp2tf`. Построить частотные характеристики цифрового фильтра.

Таблица 1.1

Параметры к вариантам 1-12

Номер варианта	Передаточная функция $H(z)$ или импульсная характеристика $h(n)$ линейной дискретной системы
1, 9	$H(z) = \frac{0.0471101 (1 + z^{-1})(1 + 2z^{-1} + z^{-2})}{1 - 0,862573 z^{-1} + 0,429829 z^{-2}}$
2, 10	$H(z) = a_0 + \sum_{k=1}^p a_k z^{-k}, a_0 = a_{10} = 0,0118785,$ $a_1 = a_9 = -0.0000003, a_2 = a_8 = -0,0621937,$ $a_3 = a_7 = 0,0000008, a_4 = a_6 = 0,3007862, a_5 = 0,4999990$
3, 11	$H(z) = 0,0154663 \frac{1 - z^{-2}}{1 + 0,969067 z^{-2}}$

Номер варианта	Передаточная функция $H(z)$ или импульсная характеристика $h(n)$ линейной дискретной системы
4, 12	$H(z) = \frac{0,034439 + 0,068882 z^{-1} + 0,034439 z^{-2}}{1 - 1,527344 z^{-1} + 0,681641 z^{-2}}$

Окончание табл. 1.1

Номер варианта	Передаточная функция $H(z)$ или импульсная характеристика $h(n)$ линейной дискретной системы
5	$h(n) = \left(\frac{1}{3}\right)^n \sin\left(n \frac{\pi}{6}\right), n \geq 0$
6	$h(n) = \left(\frac{1}{2}\right)^n \cos\left(\frac{\pi n}{3}\right), n \geq 0$
7	$h(n) = \text{ch}(2n), n \geq 0$
8	$h(n) = (n+1)3^{-n}, n \geq 0$

15. Исследовать поведение частотных характеристик линейной дискретной системы 2-го порядка при различных значениях параметров передаточной функции.

16-20. Построить графики оконных функций (табл. 1.2), которые используются при проектировании КИХ-фильтров (см. лабораторную работу 2), во временной и частотной областях в линейном и полулогарифмическом масштабах. Исследовать поведение этих функций при различных входных параметрах.

Таблица 1.2

Параметры к вариантам 16-20

Номер варианта	Оконная функция	Функция MATLAB
16	Кайзера	kaiser(n, beta)
17	Чебышева	chebwin(n, rs)
18	Хэмминга	hamming(n)
19	Бартлетта	bartlett(n)
20	Хэннинга	hanning(n)

### Порядок выполнения работы

1. Изучить основные сведения по работе.
2. Выполнить задание теоретически.

3. В соответствии с заданием составить программу с использованием системы MATLAB и получить результаты на компьютере.

4. Добиться совпадения результатов 2 и 3.

5. Результаты работы представить преподавателю.

### Содержание отчета

1. Цель работы.

2. Текст задания, общая схема решения.

3. Теоретическое выполнение задания.

4. Текст программы.

4. Результаты выполнения программы.

5. Выводы.

Пример выполнения лабораторной работы приведен в приложении 1.

### Вопросы для самоконтроля

1. Как связаны между собой передаточная функция линейной дискретной системы и ее импульсная характеристика?

2. Как определяется частота дискретизации сигнала по теореме Котельникова?

3. Каковы преимущества цифровой обработки сигналов по сравнению с аналоговой?

4. Как найти однородное решение линейного разностного уравнения?

5. Дать определение одностороннего  $z$ -преобразования последовательности  $x(n)$ . Всегда ли существует это преобразование?

6. Доказать свойство  $z$ -преобразования «умножение на экспоненту»: если  $x(n) \leftrightarrow X(z)$ , то  $a^n x(n) \leftrightarrow X(a^{-1}z)$ .

7. Доказать свойство  $z$ -преобразования «умножение на  $n$ »: если  $x(n) \leftrightarrow X(z)$ , то  $x(n) \leftrightarrow -z dX(z)/dz$ .

8. Дать определение радиуса сходимости  $z$ -преобразования.

9. Назвать способы выполнения обратного  $z$ -преобразования.

10. На чем основано применение  $z$ -преобразования к линейным дискретным системам?

11. Дать определение передаточной функции линейной дискретной системы.

12. Как связаны между собой передаточная функция линейной дискретной системы и ее импульсная характеристика?

13. Как вычислить выходной сигнал линейной дискретной системы, если известен входной сигнал и импульсная характеристика системы?

14. Дать определение и назвать условия устойчивости линейной дискретной системы.

15. Каковы свойства частотной характеристики линейной дискретной системы?

## **ЛАБОРАТОРНАЯ РАБОТА 2**

### **Проектирование, реализация и исследование характеристик цифрового фильтра**

#### **Теоретические сведения**

В наиболее общем смысле цифровой фильтр (ЦФ) представляет собой линейную инвариантную систему, которая реализуется на основе использования арифметического устройства с ограниченной точностью. Таким образом, к цифровому фильтру как к линейной дискретной системе относятся все понятия и положения, упомянутые в первой лабораторной работе.

Проектирование цифрового фильтра включает пять этапов:

- 1) определение требуемых свойств системы;
- 2) решение задачи аппроксимации с целью определения коэффициентов фильтра, при которых фильтр удовлетворяет заданным требованиям;
- 3) выбор конкретной схемы построения фильтра и квантование найденных значений его коэффициентов в соответствии с фиксированной длиной машинного слова;
- 4) квантование переменных величин фильтра, т.е. выбор длины слова входных, выходных и промежуточных переменных;
- 5) проверка моделированием, удовлетворяет ли полученный фильтр заданным требованиям.

Если заданные требования не удовлетворяются, то после этапа 5 нужно возвращаться к этапам 3 и 4.

Требования к частотным характеристикам принимают вид допусков на ошибки аппроксимации. В случае фильтра нижних частот (рис. 2.1) имеется полоса пропускания, в пределах которой характеристика должна аппроксимироваться величиной, равной 1, с ошибкой  $\pm\delta_p$ , т.е.

$$1 - \delta \leq |H(e^{j\omega})| \leq 1 + \delta_p, \omega \leq \omega_p.$$

Имеется полоса непропускания (задерживания), в которой

$$|H(e^{j\omega})| < \delta, \omega_s \leq \omega \leq \pi,$$

где  $\omega_p$  и  $\omega_s$  – частоты среза полосы пропускания и непропускания соответственно. Имеет место также переходная полоса ненулевой ширины  $(\omega_p - \omega_s)$ , в которой характеристика плавно спадает. Обычно ошибки задаются в децибелах:

$$R_p = -20 \lg \delta_p;$$

$$R_s = -20 \lg \delta_s.$$

Цифровые фильтры можно разделить на две группы: с бесконечной импульсной характеристикой (БИХ-фильтры) и конечной импульсной характеристикой (КИХ-фильтры).

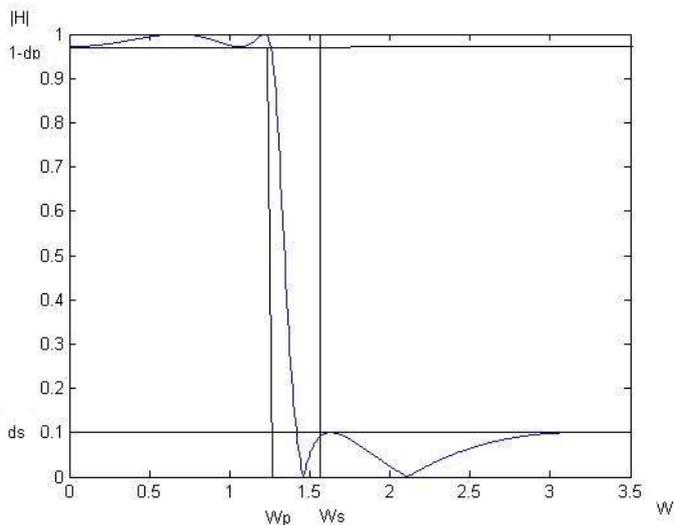


Рис. 2.1. Допустимые пределы для аппроксимации идеального фильтра нижних частот

Для БИХ-фильтра:

$$y(n) = \sum_{i=0}^N b_i x(n-i) - \sum_{i=1}^N a_i y(n-i),$$

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

В КИХ-фильтре

$$y(n) = \sum_{i=0}^N b_i x(n-i),$$

$$H(z) = \sum_{i=0}^N b_i z^{-i}.$$

т.е. коэффициенты  $b_i$  являются отсчетами импульсной характеристики.

Достоинство КИХ-фильтров заключается в возможности проектирования фильтров со строго линейной фазовой характеристикой, что важно, например, в системах обработки речи. В этом случае  $b_i = b_{N-i}$ .

Часто КИХ-фильтры можно представить в рекурсивной форме, например:

$$y(n) = x(n) + x(n-1) + x(n-2) + x(n-3) + x(n-4),$$

$$H(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} = \sum_{i=0}^4 z^{-i},$$

но в то же время по формуле суммы конечного числа членов геометрической прогрессии

$$H(z) = \frac{1 - z^{-4} z^{-1}}{1 - z^{-1}}.$$

Нерекурсивные КИХ-фильтры всегда устойчивы, в них легко минимизировать ошибки округления.

К недостаткам КИХ-фильтров относится гораздо больший порядок фильтра для обеспечения заданных требований к частотной характеристике по сравнению с БИХ-фильтрами.

### **Проектирование БИХ-фильтра на основе аналогового прототипа**

В системе MATLAB имеется возможность рассчитывать и проектировать БИХ-фильтры при помощи:

- 1) аналогового прототипа цифрового фильтра;
- 2) параметрического моделирования.

Метод, основанный на применении аналогового прототипа, является традиционным для расчета цифровых фильтров. Использование прототипа является разумным по следующим причинам:

1) способы расчета аналоговых фильтров являются хорошо отработанными и, поскольку могут быть получены полезные результаты, оказывается выгодным использовать уже развитые для аналоговых фильтров процедуры расчета;

2) многие полезные методы расчета аналоговых фильтров имеют сравнительно простые расчетные процедуры в замкнутой

форме. Поэтому методы расчета цифрового фильтра, основанные на таких аналоговых расчетных формулах, являются довольно простыми;

3) во многих приложениях представляет интерес использование цифрового фильтра для работы аналогового фильтра.

Данный метод проектирования включает 2 основных этапа:

1. Построение аналогового прототипа.

На этом этапе находится передаточная функция аналогового прототипа. В пакете MATLAB используются 4 вида аппроксимации для аналогового прототипа:

- 1) Баттерворта;
- 2) Чебышева 1-го типа;
- 3) Чебышева 2-го типа;
- 4) при помощи эллиптических функций.

Фильтр, использующий эти аппроксимации, имеет соответствующие названия.

Фильтры Баттерворта характеризуются тем, что имеют максимально гладкую АЧХ в полосе пропускания. Квадрат АЧХ:

$$|H(\omega)|^2 = \frac{1}{\left(1 + \left(\frac{\omega}{\omega_c}\right)^2\right)^n},$$

где  $n$  – порядок фильтра,  $\omega_c$  – частота среза.

Отличительной чертой фильтров Чебышева является наименьшая величина максимальной ошибки аппроксимации в заданной полосе частот. В зависимости от того, где минимизируется ошибка – в полосе пропускания или в полосе непропускания, различают фильтры Чебышева 1-го и 2-го типа.

Фильтры Чебышева 1-го типа обеспечивают пульсации частотной характеристики в полосе пропускания и монотонное изменение ослабления в полосе непропускания. Квадрат АЧХ

$$|H(\omega)|^2 = \frac{1}{\left(1 + \varepsilon^2 T_n^2\left(\frac{\omega}{\omega_c}\right)\right)},$$

где  $T_n(\omega)$  – полином Чебышева  $n$ -го порядка,  $\varepsilon$  – параметр, характеризующий пульсации в полосе пропускания,  $\omega_c$  – частота среза.

Фильтры Чебышева 2-го типа (обратные фильтры Чебышева) обеспечивают монотонное изменение ослабления в полосе пропускания и равновеликие пульсации в полосе непропускания. Квадрат АЧХ

$$|H(\omega)|^2 = \frac{1}{\left(1 + \varepsilon^2 \left(\frac{T_n(\omega_c)}{T\left(\frac{\omega_c}{\omega}\right)}\right)^2\right)}$$

где  $\omega_c$  – наименьшая частота, на которой в полосе непропускания достигается заданный уровень ослабления.

Эллиптические фильтры характеризуются тем, что их АЧХ имеет равновеликие пульсации и в полосе пропускания, и в полосе непропускания. Квадрат АЧХ

$$|H(\omega)|^2 = \frac{1}{\left(1 + \varepsilon^2 R_n^2(\omega, L)\right)}$$

где  $R_n(\omega, L)$  – рациональная функция Чебышева, а  $L$  – параметр, характеризующий пульсации функции  $R_n(\omega, L)$ . Для заданных порядка фильтра и уровня пульсаций не существует других фильтров с более быстрым переходом от полосы пропускания к полосе непропускания.

АЧХ фильтров представлены на рис. 2.2 и 2.3.

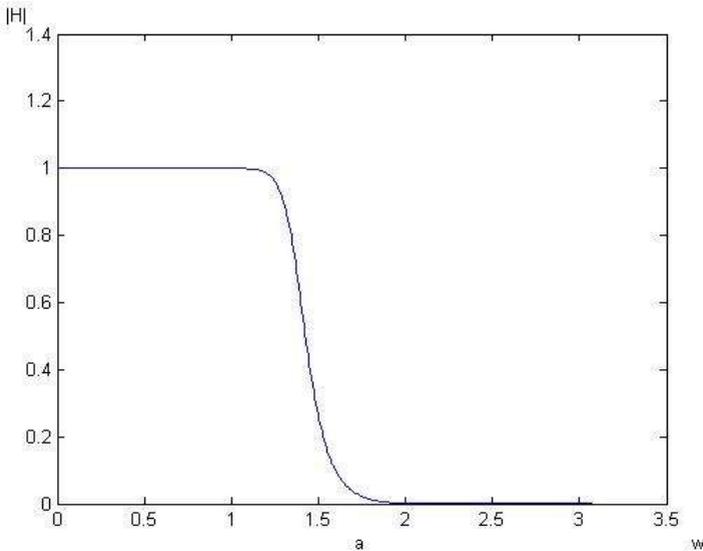
При переходе частотной характеристики реального фильтра из полосы пропускания в полосу непропускания существует переходная область, поэтому при проектировании в качестве требований к АЧХ задают не частоту среза, а частоты, характеризующие начало и конец переходной области. Частота среза будет находиться между ними. Для определения передаточной функции нужно найти такой порядок фильтра, при котором АЧХ удовлетворяла бы заданным требованиям.

## 2. Дискретизация.

На этом этапе происходит получение передаточной функции цифрового фильтра из передаточной функции аналогового фильтра. При билинейном преобразовании передаточная функция цифрового фильтра  $H(z)$  рассчитывается с помощью алгебраической подстановки

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

где  $T$  – период дискретизации. При этом преобразовании ось  $j\omega$  из непрерывной  $s$ -области отображается в единичную окружность на дискретной  $z$ -плоскости, левая половина  $s$ -области отображается во внутреннюю область единичного круга, а правая половина  $s$ -области – во внешнюю область единичного круга.



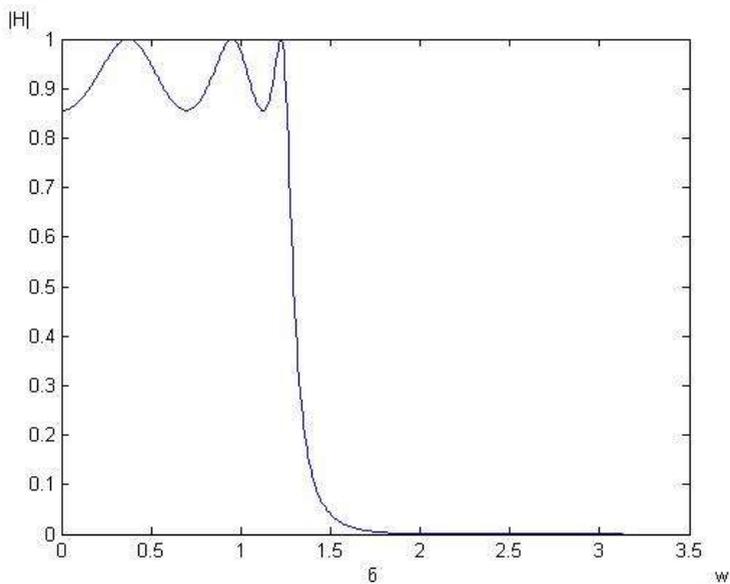


Рис. 2.2. Амплитудно-частотные характеристики фильтров нижних частот: а) Баттерворта; б) Чебышева 1-го типа

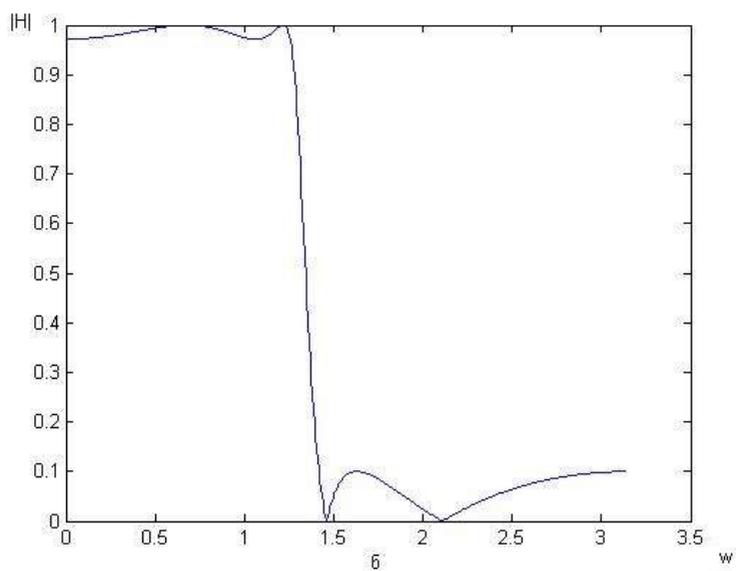
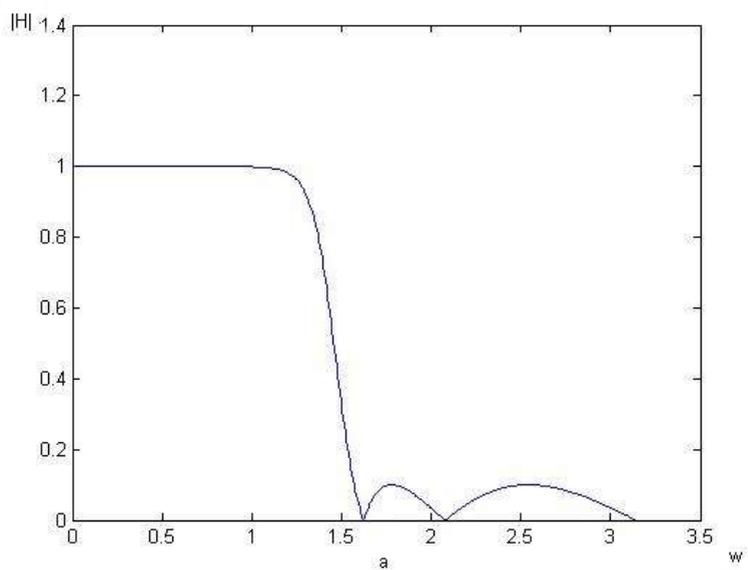


Рис. 2.3. Амплитудно-частотные характеристики фильтров нижних частот: а) Чебышева 2-го типа; б) эллиптического

При этом частота корректируется при помощи выражения

$$\omega = \frac{2}{T} \operatorname{tg}\left(\frac{w}{2}\right),$$

где  $w$  – круговая частота, используемая в дискретной области.

Все вышеуказанные этапы проектирования БИХ-фильтра могут реализовываться и при помощи функций, которые находят параметры фильтра по заданным требованиям (табл. 2.1).

Таблица 2.1

Функции, определяющие параметры передаточной функции цифрового фильтра по заданным требованиям

Фильтр	Функции
Баттерворта	$[n, W_n] = \text{buttord}(W_p, W_s, R_p, R_s)$ $[b,a] = \text{butter}(n, W_n, \text{options})$
Чебышева 1-го типа	$[n, W_n] = \text{cheb1ord}(W_p, W_s, R_p, R_s)$ $[b,a] = \text{cheby1}(n, R_p, W_n, \text{options})$
Чебышева 2-го типа	$[n, W_n] = \text{cheb2ord}(W_p, W_s, R_p, R_s)$ $[b,a] = \text{cheby2}(n, R_s, W_n, \text{options})$
Эллиптический	$[n, W_n] = \text{ellipord}(W_p, W_s, R_p, R_s)$ $[b,a] = \text{ellip}(n, R_p, R_s, W_n, \text{options})$

Вектор  $W_p$  содержит нормализованные частоты, соответствующие полосе пропускания, а  $W_s$  – нормализованные частоты, соответствующие полосе непропускания.  $R_p$  задает максимальное значение пульсаций в полосе пропускания в децибеллах, а  $R_s$  – минимальное затухание в полосе непропускания в децибеллах. Параметр `options` может принимать значения ‘high’ для фильтра верхних частот (по умолчанию синтезируется фильтр нижних частот) и ‘stop’ – для полосозаграждающего фильтра (иначе создается полосовой фильтр). После применения первой из пары функций для каждого типа фильтра  $n$  будет содержать порядок фильтра, а вектор  $W_n$  – нормализованные частоты среза, применение второй функции даст параметры  $b$  и  $a$  фильтра.

Ниже приведен пример проектирования фильтра Чебышева 1-го типа нижних частот с частотой дискретизации 1000 Гц с максимальной величиной пульсаций 3 дБ в полосе пропускания от 0 до 100 Гц и с минимальным затуханием 15 дБ в полосе от 150 до 500 Гц (половины частоты дискретизации).

```
Fd = 1000; % Задаем частоту дискретизации
Fd2 = Fd/2;
Wp = 100/Fd2; Ws = 150/Fd2;% Нормализуем частоты
Rp = 3; Rs = 15; % Задаем величины затуханий
[n, Wn] = cheblord(Wp,Ws,Rp,Rs);% Вычисляем по-
рядок фильтра и частоту среза
[b,a]=cheby1(n, Rp,Wn);% Вычисляем параметры
фильтра
[H,w]=freqz(b,a); % Вычисляем частотную характе-
ристику
ff = w/pi*Fd2; % Задаем сетку частот
plot(ff,abs(H)),xlabel('Частота (Гц)'),.
ylabel('Амплитуда'); % Строим АЧХ
```

### **Проектирование БИХ-фильтра при помощи параметрического моделирования**

Многие последовательности  $x(nT)$ , представляющие собой сумму детерминированных и случайных последовательностей, могут быть достаточно хорошо аппроксимированы выходным сигналом  $y(nT)$  линейного дискретного фильтра. Подобрав соответствующим образом параметры этого фильтра, можно добиться отношения  $x(nT) \approx y(nT)$ .

Для авторегрессионного фильтра выходные значения будут вычисляться на основе уравнения линейного предсказания

$$y(nT) = x(nT) = -a(2)x(nT - 1) - a(3)x(nT - 2) - \dots - a(k + 1)x(nT - k - 1).$$

Коэффициенты  $a(k)$  в уравнении являются коэффициентами АР-фильтра, которые вычисляются функцией `lpc`:

$$a = \text{lpc}(x, n),$$

где  $x$  – входной сигнал,  $n$  – порядок АР-фильтра.

Таким образом, зная импульсную характеристику фильтра, можно определить его параметры в передаточной функции.

Для определения параметров авторегрессионного бегущего усредняющего фильтра по его импульсной характеристике применяют метод Прони, названный в честь ученого Гаспара Рише (барона де Прони), проводя моделирование с использованием линейной комбинации экспоненциальных функций. Этот метод тесно связан с алгоритмами линейного предсказания по методу наименьших квадратов, используемых для оценивания параметров фильтров. Данный метод реализуется с помощью функции `prony`. Выражение

$$[b, a] = \text{prony}(x, nb, na)$$

находит параметры передаточной БИХ-фильтра по его заданной в виде вектора  $x$  импульсной характеристике,  $nb$ ,  $na$  – порядки числителя и знаменателя. Ниже приведен пример сравнения точности проектирования БИХ-фильтра с помощью данной функции и с использованием функции `butter`.

```
[b,a] = butter(4,0.2); % Вычисляем параметры
%фильтра с использованием аппроксимации Баттерворта
h = filter(b,a,eye(26)); %Находим импульсную
% характеристику с 26 отсчетами.
[bb,aa] = prony(h,4,4); % По импульсной
% характеристике находим с помощью
% параметрического моделирования параметры фильтра.
```

Следует отметить, что  $bb$  и  $aa$  будут отличаться соответственно от  $b$  и  $a$  с ошибкой  $10^{-13}$ .

В приведенных выше методах сигнал, который аппроксимировался при помощи подобранной модели, был представлен во временной области. Для аппроксимации сигнала, представленного в частотной области, существуют функции `invfreqz` (обратная функции `freqz`) и `yulewalk`. Выражение

$$[b, a] = \text{invfreqz}(h, w, nb, na)$$

находит параметры БИХ-фильтра по заданной комплексной частотной характеристике фильтра, где  $w$  – вектор частот, в которых вычислена  $h$ ;  $nb$ ,  $na$  – порядки, которые должны иметь числитель и знаменатель передаточной функции фильтра.

Функция `yulewalk` находит параметры БИХ-фильтра (фильтр Юла – Уокера) по входным значениям векторов  $n$ ,  $f$ ,  $m$ , где  $n$  – порядок фильтра, а  $f$  и  $m$  характеризуют желаемую АЧХ:

-  $f$  – вектор, содержащий отсчеты частоты, значения которых нормализованы путем деления на половину частоты дискретизации и лежат в пределах отрезка  $[0,1]$  и заполняют вектор в возрастающем порядке. Первый отсчет должен быть всегда равен нулю, а последний – единице;

-  $m$  – вектор, содержащий значения проектируемой АЧХ. Каждому значению вектора  $m$  должно соответствовать определенное значение частоты в векторе  $f$ ,  $f$  и  $m$  должны быть одной длины.

Особенностью данного фильтра является возможность аппроксимации АЧХ произвольной конфигурации. Ниже приведен пример проектирования многополосного фильтра с порядком  $n = 10$ , который имеет АЧХ произвольной конфигурации.

```
m = [0 0 1 1 0 0 1 1 0 0]; % Задаем АЧХ фильтра
f = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 1];
% Задаем вектор частот
n = 10; % Задаем порядок фильтра
[b,a] = yulewalk(n,f,m); % Вычисляем параметры
фильтра
[h,w] = freqz(b,a,128); % Находим частотную ха-
рактеристику.
plot(f,m,w/pi,abs(h)) % Строим проектную и
спроектированную АЧХ
```

### **Проектирование КИХ-фильтров при помощи оконных функций**

В системе MATLAB имеется возможность рассчитывать и проектировать КИХ-фильтры при помощи:

- 1) оконных функций;
- 2) построения оптимального фильтра.

Опишем идею метода, использующего оконные функции. Импульсная характеристика идеального фильтра нижних частот с частотой среза  $\omega_0$  описывается формулой

$$g(t) = \frac{\sin(\omega_0 t)}{\omega_0 t}.$$

Однако такая импульсная характеристика физически не реализуема, так как ее значения отличаются от нуля при  $t < 0$ . Для реализации фильтра его импульсную характеристику усекают с обеих сторон и смещают вправо. Однако резкое усечение импульсной характеристики приводит к появлению пульсаций в полосе пропускания и полосе непропускания, такое явление называют эффектом Гиббса. Для улучшения вида частотной характеристики импульсную характеристику умножают на весовую функцию (временное окно), которая обычно близка к единице в середине и плавно убывает по краям. В результате импульсная характеристика становится более плавной и, как следствие этого, улучшается вид частотной характеристики.

Оконные функции широко используются для проектирования КИХ-фильтров, а также находят применение в спектральном анализе (табл. 2.2). Параметр  $n$  задает количество отсчетов оконной функции,  $R_s$  – минимальную величину затуханий в полосе непропускания,  $\beta$  (или  $\beta$ ) – параметр, назначение которого приводится ниже.

Окно Бартлетта описывается следующим образом:

$$w(k) = \begin{cases} \frac{2(k-1)}{n-1}, & 1 \leq k \leq \frac{n+1}{2} \\ 2 - \frac{2(k-1)}{n-1}, & \frac{n+1}{2} \leq k \leq n \end{cases}.$$

Окно Блэкмана описывается следующей функцией:

$$w(k) = 0,42 - 0,5 \cos\left(2\pi \frac{k-1}{n-1}\right) + 0,08 \cos\left(4\pi \frac{k-1}{n-1}\right),$$

$$k = 1, 2, \dots, n.$$

Прямоугольное окно содержит единицы в количестве  $n$ .

Окно Хэмминга описывается следующей функцией:

$$w(k+1) = 0,54 - 0,46 \cos\left(2\pi \frac{k}{n-1}\right), \text{ где } k = 0, 1, \dots, n-1.$$

Таблица 2.2

## Оконные функции системы MATLAB

Окно	Функция
Бартлетта	Bartlett(n)
Блэкмана	Lackman(n)
Прямоугольное	Boxcar(n)
Чебышева	Chebwin(n,R <sub>с</sub> )
Хэмминга	Hamming(n)
Хэннинга	Hanning(n)
Кайзера	kaiser(n,beta)
Треугольное	triang(n)

Окно Хэннинга:

$$w(k) = 0,5 \left( 1 - \cos \left( 2\pi \frac{k}{n+1} \right) \right), \text{ где } k = 1, 2, \dots, n.$$

Треугольное окно при четном  $n$  описывается следующим образом:

$$w(k) = \begin{cases} \frac{2k}{n+1}, & 1 \leq k \leq \frac{n+1}{2} \\ \frac{2(n-k+1)}{n+1}, & \frac{n+1}{2} \leq k \leq n \end{cases},$$

при нечетном  $n$ :

$$w(k) = \begin{cases} \frac{2k-1}{n}, & 1 \leq k \leq \frac{n}{2} \\ \frac{2(n-k+1)}{n}, & \frac{n}{2} + 1 \leq k \leq n \end{cases}.$$

Достоинством приведенных выше окон является простота их математического описания и, как следствие, в некоторых случаях – простота реализации. Однако эти окна по многим критериям не являются лучшими. Фильтры, использующие при проектировании прямоугольное окно, имеют узкую переходную

полосу. Но большая величина пульсаций в полосах пропускания и непропускания делает эту оконную функцию малоприменимой. Фильтры, использующие при проектировании окна Хэмминга или Хэннинга, имеют минимальную величину пульсаций в полосах пропускания и непропускания. Но при их использовании образуется широкая переходная полоса, что тоже является малоприменимым фактом. Известны другие виды оконных функций, превосходящие по многим параметрам перечисленные выше окна – это окна Кайзера и Чебышева.

Окно Чебышева было разработано в виде дискретной функции (непрерывный аналог ее не найден):

$$W(f) = hT_{n-1}[z_0 \cos(\pi fT)],$$

где  $T_{n-1}(z)$  – полином Чебышева первого рода  $(n-1)$ -го порядка,  $T$  – период дискретизации,  $h$ ,  $z_0$  – параметры, меняющиеся в зависимости от требований к пульсациям в полосе непропускания. Окно Чебышева обеспечивает минимально возможную ширину переходной полосы при заданном уровне пульсаций в полосе непропускания.

Окно Кайзера описывается выражением

$$w(k) = \frac{I_0(\beta \cdot \sqrt{1 - (2k(n-1))^2})}{I_0(\beta)}, \quad -\left(\frac{n-1}{2}\right) \leq k \leq \frac{n-1}{2},$$

где  $I_0$  – функция Бесселя нулевого порядка,  $\beta$  – параметр, определяемый из системы уравнений:

$$\beta = \begin{cases} 0,1102 \cdot (\alpha - 8,7), & \alpha > 50, \\ 0,5842 \cdot (\alpha - 2,1)^{0,4} + 0,07886(\alpha - 21), & 21 \leq \alpha \leq 50, \\ 0, & \alpha < 21, \end{cases}$$

где  $\alpha$  – параметр, содержащий величину пульсаций в полосе непропускания (дБ).

Порядок фильтра, который использует окно Кайзера, вычисляется по формуле

$$n = \frac{\alpha - 8}{2,285\Delta\omega} + 1,$$

где  $\Delta\omega$  – переходная полоса, измеряемая в  $\text{с}^{-1}$ .

Привлекательное свойство этой оконной функции состоит в том, что величину пульсаций можно изменять непрерывно от малого значения, соответствующего окну Блэкмана, до больших значений в случае прямоугольного окна, изменяя только параметр  $\beta$ . К тому же ширина переходной полосы может регулироваться, как и других окон, подбором числа отсчетов  $n$ .

Проектирование КИХ-фильтров с применением временных окон осуществляется функцией `fir1`:

```
b = fir1(n, Wn);
```

```
b = fir1(n, Wn, 'ftype');
```

```
b = fir1(n, Wn, window);
```

```
b = fir1(n, Wn, 'ftype', window);
```

где  $n$  – порядок фильтра,  $W_n$  – вектор, содержащий нормированную частоту среза, т.е. ее значения лежат в пределах  $[0, 1]$ . Если  $W_n = [w_1 \ w_2]$ , то функция `fir1` возвращает коэффициенты полосового фильтра, полоса пропускания которого лежит в пределах  $[w_1 \ w_2]$ . Выражение в кавычках 'ftype' характеризует тип фильтра. Если 'ftype' = 'high', то будут вычисляться параметры фильтра верхних частот с частотой среза  $W_n$ . Если 'ftype' = 'stop' и  $W_n = [w_1 \ w_2]$ , то будет проектироваться полосозаграждающий фильтр. В качестве параметра `window` указывается тип окна, которое нужно применить. При отсутствии этого параметра функция `fir1` использует окно Хэмминга.

Порядок фильтра  $n$  можно вычислить только для окна Кайзера, используя функцию `kaiserord`:

```
[n, wn, beta, ftype] = kaiserord(f, a, dev, wd),
```

где  $f$  – вектор частот,  $a$  – вектор амплитуд,  $dev$  – вектор отклонений амплитуд, остальные параметры описаны выше. Длина вектора  $f$  должна быть равна двум длинам вектора  $a$  минус 2. Например, если  $f = [f_1 \ f_2 \ f_3 \ f_4]$ , то  $a = [a_1 \ a_2 \ a_3]$ , причем на участке частот  $0 - f_1$  амплитуда частотной характеристики должна быть равна  $a_1$ , на участке  $f_2 - f_3$  –  $a_2$ , на участке  $f_4$  – половина частоты дискретизации –  $a_3$ .

Функция `fir1` позволяет проектировать разнообразные фильтры с АЧХ стандартной конфигурации. Для построения фильтров с АЧХ произвольной конфигурации можно воспользоваться функцией `fir2`:

```
b=fir2(n,f,m),
b=fir2(n,f,m,window),
```

где  $n$  – порядок фильтра, `window` – окно, а АЧХ задается векторами  $f$  и  $m$ :

$f$  – вектор, содержащий отсчеты частоты, значения которых нормализованы и лежат в пределах отрезка  $[0,1]$  и заполняют вектор в возрастающем порядке. Первый отсчет должен быть всегда равен нулю, а последний – единице;

$m$  – вектор, содержащий значения проектируемой АЧХ. Каждому значению вектора  $m$  должно соответствовать определенное значение частоты в векторе  $f$ ,  $f$  и  $m$  должны быть одной длины.

Ниже приведен пример проектирования КИХ-фильтра нижних частот с окнами Кайзера и Чебышева.

```
wd=1000; % Частота дискретизации
wd2=wd/2;
wp=250; % Частота среза полосы пропускания
ws=350; % Частота среза полосы непропускания
rp=3; % Пульсации в полосе пропускания, дБ
rs=45; % Ослабление в полосе непропускания, дБ
f=[wp ws]; % Вектор частот
a=[1 0]; % Вектор амплитуд
dev=[10^(-rp/20) 10^(-rs/20)]; % Вектор отклонения
% амплитуд
% Вычисление порядка фильтра
[n,wn,beta,ftype]=kaiserord(f,a,dev,wd)
% Нахождение вектора b для фильтра с окном Кайзера
b=fir1(n,wn,ftype,kaiser(n+1,beta),'noscale');
% Нахождение вектора b для фильтра с окном Чебышева
b1=fir1(n,wn,ftype,chebwin(n+1,rs),'noscale');
freqz(b,1); % Построение частотной характеристики,
% фильтр соответствует заданным требованиям
pause
freqz(b1,1); % В случае окна Чебышева необходимо
% изменить порядок фильтра и повторить расчет
% коэффициентов
```

## Проектирование оптимального КИХ-фильтра

Несмотря на существование большого числа способов сравнения получаемых фильтров, возможно, самый объективный (и самый полезный) состоит в сравнении значений ширины переходных полос фильтров, необходимых для удовлетворения заданных требований к пульсациям в полосе пропускания и непропускания. При таком сравнении лучшими будут оптимальные фильтры, поскольку из всех КИХ-фильтров с линейной фазой при одинаковых заданных параметрах именно они обладают минимальной переходной полосой. В системе MATLAB имеется возможность проектирования оптимальных фильтров методом Паркса – Макклеллана, который использует чебышевскую теорию аппроксимации и алгоритм Ремеза. Идея этого метода приводится ниже.

Если  $P(e^{jw})$  представляет собой комбинацию из  $N$  косинусных функций, т.е

$$P(e^{jw}) = \sum_{n=0}^{N-1} a(n) \cos(wn), \quad (2.1)$$

то  $P(e^{jw})$  будет наилучшей аппроксимацией со взвешиванием непрерывной функции  $\hat{D}(e^{jw})$ , в компактной подобласти  $A$  из области  $(0, \pi)$ , если взвешенная функция ошибки  $E(e^{jw})$  имеет, по крайней мере,  $N+1$  экстремальных частот, т.е. таких  $N+1$  точек  $w$ , что  $w_1 < w_2 < \dots < w_N < w_{N+1}$ ,

$$E(e^{jw_i}) = -E(e^{jw_{i+1}}) \quad i = 1, 2, \dots, n \text{ и}$$

$$|E(e^{jw_i})| = \max_{W \in A} E(e^{jw_{i+1}}).$$

Уравнение ошибки вида

$$E(e^{jw_i}) = \left| \hat{W}(e^{jw}) \left[ \hat{D}(e^{jw}) - P(e^{jw}) \right] \right|$$

можно использовать для синтеза фильтров любого вида, где  $\hat{W}(e^{jw})$  – положительная весовая функция. Для построения фильтров, имеющих ошибку с таким равноколебательным пове-

дением, используется алгоритм Ремеза. Идея его состоит в решении системы уравнений вида:

$$\hat{W}(e^{jw}) \left[ \hat{D}(e^{jwk}) - P(e^{jwk}) \right] = (-1)^k \delta, i = 0, 1, \dots, N - 2.$$

где  $\delta$  – заданная на проектирование ошибка. Сначала выбирают  $N+1$  значений  $W_k$  и решают систему уравнений, получают коэффициенты  $a(n)$  в выражении (2.1) и  $\delta$ .

Если  $E(e^{jw_i}) \leq \delta$ , то полученное решение будет оптимальным. Если  $E(e^{jw_i}) > \delta$ , то выбирают новую совокупность  $N+1$  экстремальных частот.

Проектирование оптимального фильтра осуществляется с помощью функции `remez`. Вариантами вызова этой функции являются:

```
b=remez(n, f, m),
b=remez(n, f, m, w),
b=remez(n, f, m, 'ftype'),
b=remez(n, f, m, w, 'ftype').
```

Вектор  $b$  будет содержать  $n+1$  коэффициентов КИХ-фильтра порядка  $n$ , параметр  $w$  задает весовую функцию, а параметр 'ftype' задает тип фильтра. Если `ftype=hilbert`, то будет спроектирован оптимальный фильтр, реализующий преобразование Гильберта, если `ftype=differentiator`, то – дифференцирующий фильтр, а АЧХ задается векторами  $f$  и  $m$  (см. описание функции `fir2`).

Фильтр, спроектированный данным методом, называют еще минимаксным фильтром с равновеликими пульсациями. Ниже приведен пример проектирования полосового КИХ-фильтра 17-го порядка, проектная АЧХ имеет нестандартную конфигурацию.

```
f=[0 0.3 0.4 0.6 0.7 1]; % Задаем вектор частот.
m=[0 0 1 1 0 0]; % Задаем проектную АЧХ.
b=remez(17, f, m); % Находим параметры фильтра.
[h, w]=freqz(b, 1, 512); % Находим комплексную частотную характеристику.
plot(f, m, w/pi, abs(h)) % Построение АЧХ.
```

## Структурные схемы цифровых фильтров

Цифровой фильтр с заданной передаточной функцией  $H(z)$  можно построить различными способами. В реальном ЦФ шумы и погрешности квантования существенно зависят от структуры системы.

Линейное разностное уравнение непосредственно определяет способ построения ЦФ.

Так, уравнение 1-го порядка

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1)$$

можно реализовать можно с помощью схемы (рис. 2.4):

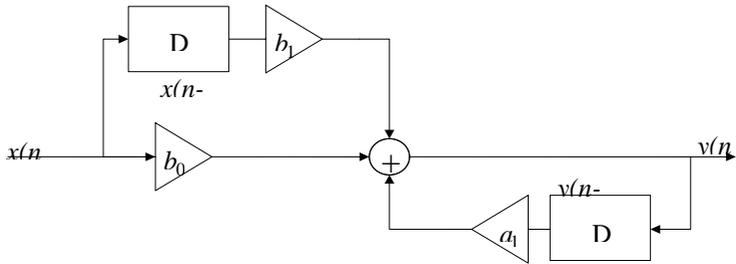
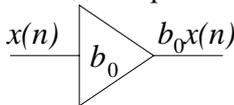
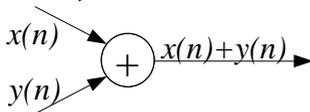


Рис. 2.4. Структурная схема ЦФ 1-го порядка

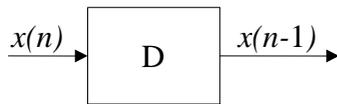
Здесь приняты следующие условные обозначения:



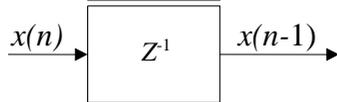
умножитель



сумматор



элемент единичной задержки



элемент единичной задержки (альтернативное обозначение)

В соответствии с разностным уравнением  $N$ -го порядка

$$y(n) = \sum_{i=0}^N b_i x(n-i) - \sum_{i=1}^N a_i y(n-i)$$

можно построить схему ЦФ в прямой форме 1 (рис. 2.5).

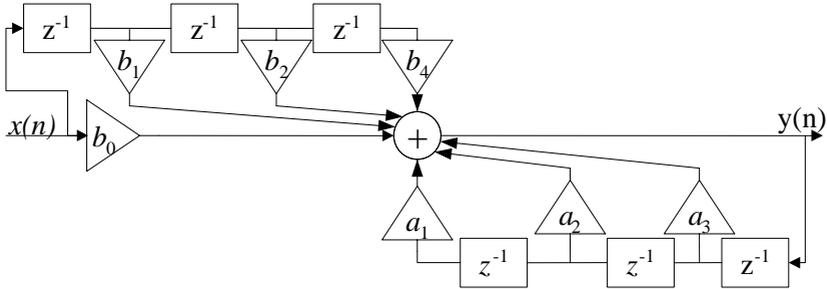


Рис. 2.5. Прямая форма 1

Особенности прямой формы 1: простота и непосредственная связь с разностным уравнением; трудности в физической реализации цифровой системы.

Перепишем  $H(z)$  в другом виде:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 + \sum_{i=1}^N a_i z^{-i}} \sum_{i=0}^N b_i z^{-i},$$

т.е. представим в виде двух последовательно соединенных звеньев

$$H_1(z) = \frac{W(z)}{X(z)} = \frac{1}{1 + \sum_{i=1}^N a_i z^{-i}},$$

$$H_2(z) = \frac{Y(z)}{W(z)} = \sum_{i=0}^N b_i z^{-i}.$$

Получим два разностных уравнения:

$$w(n) = x(n) - \sum_{i=1}^N a_i w(n-i),$$

$$y(n) = \sum_{i=0}^N b_i w(n-i),$$

которые можно реализовать, как показано на рис. 2.6.

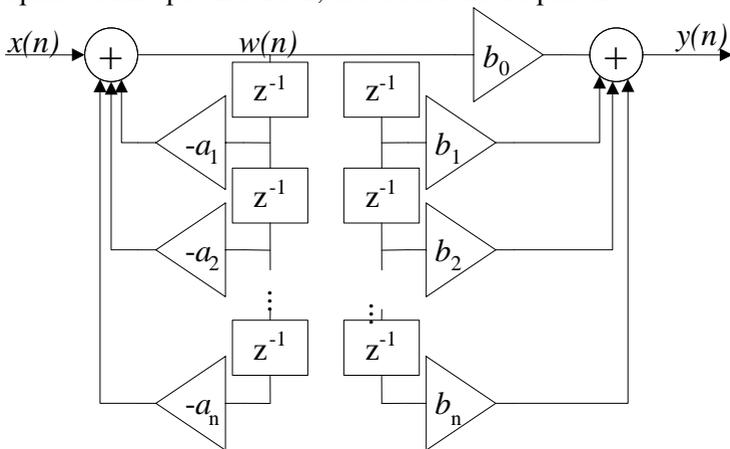


Рис. 2.6. Получение прямой канонической формы

Поскольку в цепях для  $H_1(z)$  и  $H_2(z)$  сигнал  $w(n)$  задерживается одинаково, то схему можно изобразить в виде, представленном на рис 2.7. Такая форма называется прямой канонической, или прямой формой 2.

Достоинство прямой канонической формы заключается в использовании минимального количества элементов задержки.

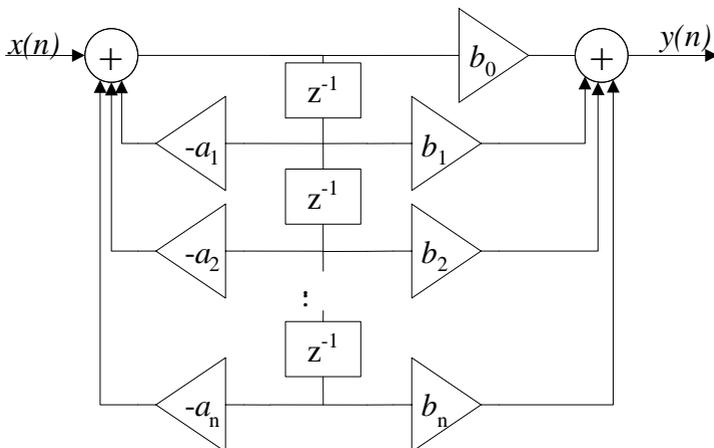


Рис. 2.7. Прямая каноническая форма (прямая форма 2)

Третья форма построения ЦФ – последовательная (каскадная) (рис. 2.8).

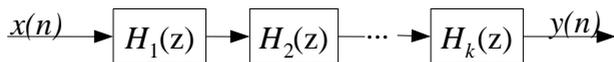


Рис. 2.8. Последовательная (каскадная) форма

Передаточная функция ЦФ представляется в виде произведения простых дробей

$$H(z) = b_0 \prod_{i=1}^k H_i(z),$$

где  $H_i(z)$  соответствует звеньям 1-го и 2-го порядка:

$$H_i(z) = \frac{1 + b_{1i}z^{-i}}{1 + a_{1i}z^{-1}},$$

$$H_i(z) = \frac{1 + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-2} + a_{2i}z^{-2}}.$$

В системе MATLAB последовательную форму ЦФ можно построить, используя функцию `tf2sos`.

Четвертая структурная схема ЦФ – параллельная (рис. 2.9).

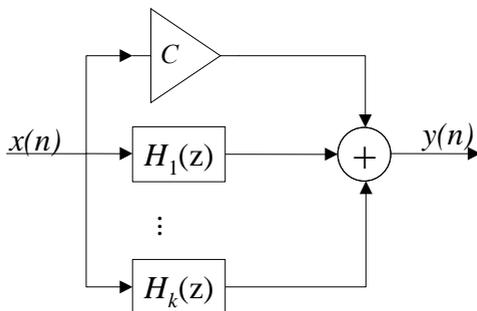


Рис. 2.9. Параллельная форма

Передающая функция ЦФ представляется в виде суммы простых дробей

$$H(z) = C + \sum_{i=1}^k H_i(z),$$

где  $H_i(z)$  соответствует звеньям 1-го и 2-го порядка:

$$H_i(z) = \frac{b_{oi}}{1 + a_{1i}z^{-1}},$$

$$H_i(z) = \frac{b_{0i} + b_{1i}z^{-1}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}}.$$

В системе MATLAB параллельную форму ЦФ можно построить, используя функцию `reseduez`.

Существуют и другие структурные схемы ЦФ.

### Задание к лабораторной работе

#### Часть 1

1.1. В системе MATLAB составить программу расчета БИХ-или КИХ-фильтра, параметры которого и требования к нему приведены в табл. 2.3.

1.2. Построить частотные характеристики фильтра (АЧХ, ФЧХ, ЛАЧХ, характеристика групповой задержки)

1.3. Построить импульсную и временные характеристики.

1.4. Получить реакцию фильтра на синусоидальный входной сигнал с частотами из полосы пропускания или полосы непропускания.

1.5. Получить коэффициенты последовательной и параллельной (только для БИХ-фильтра) форм реализации фильтра.

#### Часть 2

Программно реализовать на языке высокого уровня фильтр в прямой, канонической (БИХ-фильтр), последовательной и параллельной (БИХ-фильтр) формах при представлении коэффициентов разностного уравнения в форматах

а) с плавающей точкой,

б) с фиксированной точкой (16 бит)

и получить реакцию фильтра на единичный импульс, единичный ступенчатый и синусоидальный сигналы.

### Часть 3

Одну из форм фильтра реализовать на языке ассемблера процессора цифровой обработки сигнала.

Таблица 2.3

#### Варианты задания

Номер варианта	Тип фильтра	Характеристика фильтра
1	БИХ-фильтр нижних частот Баттерворта	$f_d = 2000$ Гц; $r_p = 1.35$ дБ; $r_s = 60$ дБ; $f_p = 400$ Гц; $f_s = 650$ Гц
2	БИХ-фильтр нижних частот Чебышева 2-го типа	$f_d = 2000$ Гц; $r_p = 1.35$ дБ; $r_s = 60$ дБ; $f_p = 400$ Гц; $f_s = 650$ Гц
3	БИХ-фильтр нижних частот эллиптический	$f_d = 2000$ Гц; $r_p = 1.35$ дБ; $r_s = 60$ дБ; $f_p = 400$ Гц; $f_s = 650$ Гц
4	БИХ-фильтр нижних частот Чебышева 1-го типа	$f_d = 2000$ Гц; $r_p = 1.35$ дБ; $r_s = 60$ дБ; $f_p = 400$ Гц; $f_s = 650$ Гц
5	БИХ-фильтр верхних частот Чебышева 2-го типа	$f_d = 1000$ Гц; $r_p = 0.2$ дБ; $r_s = 55$ дБ; $f_p = 220$ Гц; $f_s = 150$ Гц
6	БИХ-фильтр верхних частот Баттерворта	$f_d = 1000$ Гц; $r_p = 0.2$ дБ; $r_s = 55$ дБ; $f_p = 220$ Гц; $f_s = 150$ Гц
7	БИХ-фильтр верхних частот эллиптический	$f_d = 1000$ Гц; $r_p = 0.2$ дБ; $r_s = 55$ дБ; $f_p = 220$ Гц; $f_s = 150$ Гц
8	БИХ-фильтр верхних частот Чебышева 1-типа	$f_d = 1000$ Гц; $r_p = 0.2$ дБ; $r_s = 55$ дБ; $f_p = 220$ Гц; $f_s = 150$ Гц
9	Полосовой БИХ-фильтр эллиптический	$f_d = 1500$ Гц; $r_p = 0.55$ дБ; $r_s = 57$ дБ; $f_p = 540...640$ Гц; $f_s = 480...700$ Гц
10	Режекторный (полосозаграждающий) БИХ-фильтр эллиптический	$f_d = 2000$ Гц; $r_p = 1.2$ дБ; $r_s = 75$ дБ; $f_p = 600...850$ Гц; $f_s = 650...800$ Гц
11	БИХ-фильтр Юла – Уокера верхних частот	$f_d = 3000$ Гц, $f_c = 1000$ Гц

Окончание табл. 2.3

Номер варианта	Тип фильтра	Характеристика фильтра
12	БИХ-фильтр Юла – Уокера нижних частот	$f_d = 1000 \text{ Гц}; f_c = 300 \text{ Гц}$
13	Полосовой КИХ-фильтр (окно Кайзера)	$w_d = 2000 \text{ с}^{-1}; r_s = 45 \text{ дБ};$ $r_p = 0.2 \text{ дБ}; w_p = 400 \dots 600 \text{ с}^{-1}; w_s = 300 \dots 700 \text{ с}^{-1}$
14	КИХ-фильтр нижних частот (окно Кайзера)	$w_d = 1000 \text{ с}^{-1}; r_s = 40 \text{ дБ};$ $r_p = 0.25 \text{ дБ}; w_p = 150 \text{ с}^{-1};$ $w_s = 250 \text{ с}^{-1}$
15	КИХ-фильтр нижних частот (окно Чебышева)	$w_d = 1000 \text{ с}^{-1}; r_s = 40 \text{ дБ};$ $r_p = 0.25 \text{ дБ}; w_p = 150 \text{ с}^{-1};$ $w_s = 250 \text{ с}^{-1}$
16	Полосовой КИХ-фильтр (окно Чебышева)	$w_d = 2000 \text{ с}^{-1}; r_s = 45 \text{ дБ};$ $r_p = 0.2 \text{ дБ}; w_p = 400 \dots 600 \text{ с}^{-1};$ $w_s = 300 \dots 700 \text{ с}^{-1}$
17	КИХ-фильтр верхних частот (окно Кайзера)	$w_d = 100 \text{ с}^{-1}; r_s = 50 \text{ дБ};$ $r_p = 0.15 \text{ дБ}; w_p = 20 \text{ с}^{-1};$ $w_s = 15 \text{ с}^{-1}$
18	КИХ-фильтр верхних частот (окно Хэмминга)	$w_d = 100 \text{ с}^{-1}; r_s = 50 \text{ дБ};$ $r_p = 0.15 \text{ дБ}; w_p = 20 \text{ с}^{-1};$ $w_s = 15 \text{ с}^{-1}$
19	Оптимальный КИХ-фильтр верхних частот (функция remez)	$w_d = 100 \text{ с}^{-1}; r_s = 50 \text{ дБ};$ $r_p = 0.15 \text{ дБ}; w_p = 20 \text{ с}^{-1};$ $w_s = 15 \text{ с}^{-1}$
20	Оптимальный КИХ-фильтр (функция remez), частотная характеристика которого аппроксимирует дифференцирующий фильтр	Задаются самостоятельно

## Содержание отчета по работе

1. Цель работы.
  2. Особенности заданного типа фильтра.
  3. Текст программы в системе MATLAB для расчета параметров фильтра, его характеристик, а также разложения передаточной функции на произведение и сумму звеньев 2-го порядка.
  4. Полученные результаты (передаточная функция фильтра и графики его характеристик).
  5. Разностные уравнения и структурные схемы фильтра при его реализации в прямой, канонической, последовательной и параллельной формах.
  6. Текст программы на языке высокого уровня для программной реализации фильтра в различных формах при представлении коэффициентов разностного уравнения в форматах:
    - а) с плавающей точкой,
    - б) с фиксированной точкой (16 бит).
  7. Графики реакции фильтра на единичный импульс, единичный ступенчатый и синусоидальный сигналы при его программной реализации в различных формах и заключение о выборе одной из форм для реализации на языке ассемблера процессора цифровой обработки сигналов.
  8. Текст программы реализации выбранной формы фильтра на языке ассемблера процессора цифровой обработки сигналов и полученные результаты.
  9. Выводы.
- Пример выполнения лабораторной работы приведен в прил. 2.

## Вопросы для самоконтроля

1. Какие характеристики цифровых фильтров вы знаете и как они связаны между собой?
2. Какой класс входных сигналов рассматривают при определении частотной характеристики цифрового фильтра?
3. Как по заданной передаточной функции цифрового фильтра определить его частотную характеристику?
4. Дать определение АЧХ, ФЧХ и групповой задержки цифрового фильтра.

5. Почему частотную характеристику цифрового фильтра рассматривают в диапазоне частот от 0 до половины частоты дискретизации? Как достроить частотную характеристику в случае расширения частотного диапазона?

6. Как называется форма цифрового фильтра, построенная непосредственно по его разностному уравнению? Какие структурные элементы необходимы для построения схемы фильтра?

7. Чем отличаются передаточные функции БИХ- и КИХ-фильтров?

8. Как получить импульсную и временную характеристики цифрового фильтра.

9. Каковы преимущества последовательной формы цифрового фильтра по сравнению с прямой?

10. Каковы преимущества параллельной формы цифрового фильтра по сравнению с прямой?

11. Каковы преимущества прямой канонической формы цифрового фильтра по сравнению с прямой?

12. Назовите этапы синтеза цифровых фильтров.

13. Изобразите АЧХ идеальных фильтров нижних и верхних частот. Почему идеальные фильтры физически нереализуемы?

14. Какие требования к частотной характеристике задаются при синтезе БИХ-фильтров?

15. На чем основан метод билинейного  $z$ -преобразования при синтезе БИХ-фильтров?

16. Каковы особенности фильтров Баттерворта, Чебышева, эллиптических?

17. Какие требования к частотной характеристике задаются при синтезе КИХ-фильтров?

## **ЛАБОРАТОРНАЯ РАБОТА 3** **Быстрое преобразование Фурье**

### **Теоретические сведения** **Дискретное преобразование Фурье**

Для описания аналоговых и дискретных сигналов в частотной области используется аппарат преобразования Фурье. Спектром  $X_a(j\omega)$  аналогового сигнала  $x_a(t)$  называют прямое преобразование Фурье:

$$X_a(j\omega) = \int_0^{\infty} x_a(t) e^{-j\omega t} dt.$$

В свою очередь, согласно обратному преобразованию Фурье:

$$x_a(t) = 1/2\pi \int_{-\infty}^{\infty} X_a(j\omega) e^{j\omega t} d\omega.$$

Преобразование Фурье существует и для дискретной последовательности. В теории цифровой обработки сигналов рассматривают дискретное преобразование Фурье.

Пусть  $x_p(n)$  – периодическая последовательность с периодом в  $N$  отсчетов, т.е.

$$x_p(n) = x_p(n + Nm), m = 0, \pm 1, \pm 2, \dots$$

ДПФ называют пару взаимнооднозначных преобразований

$$X_p(k) = \sum_{n=0}^{N-1} x_p(n) e^{-j\frac{2\pi}{N}kn}, k = 0, 1, \dots, N-1 \text{ (ДПФ)},$$

$$x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn}, n = 0, 1, \dots, N-1 \text{ (обратное ДПФ)}.$$

Введем обозначение  $W_N^{kn} = e^{-j\frac{2\pi}{N}kn}$  – поворачивающий множитель; тогда

$$X_p(k) = \sum_{n=0}^{N-1} x_p(n) W_N^{kn},$$

$$x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}.$$

$X_p(k)$  является периодической функцией с периодом  $N$ , так как

$$\begin{aligned} W_N^{kn} &= W_N^{(k+mN)n} = \\ &= e^{j\frac{2\pi}{N}(k+mN)n} = e^{j\frac{2\pi}{N}kn} \cdot e^{j\frac{2\pi}{N}mNn} = \\ &= e^{j\frac{2\pi}{N}kn} \cdot e^{j2\pi mn} = e^{j\frac{2\pi}{N}kn}, \\ &\text{т.к. } e^{j2\pi mn} = 1 \end{aligned}$$

ДПФ может быть использовано и для представления последовательности  $x(n)$  конечной длины, определенной при  $n = 0, 1, \dots, N-1$  и равной нулю вне интервала  $0..N-1$ .

Действительно,

$$\begin{aligned}x(n) &= x_p(n), n = 0, 1, \dots, N-1, \\x(n) &= 0, n = N, N+1, \dots \text{ и } n < 0,\end{aligned}$$

тогда

$$\begin{aligned}X(k) &= X_p(k), n = 0, \dots, N-1, \\X(k) &= 0, n = N, N+1, \dots \text{ и } n < 0.\end{aligned}$$

*Пример 1*

$x(n) = \delta(n)$  – функция единичного отсчета.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi}{N} nk} = e^{-j \frac{2\pi}{N} \cdot 0 \cdot k} = 1.$$

*Пример 2*

$x(n) = u(n)$  – единичная ступенчатая функция,  $0 \leq n \leq N-1$ .

$$\begin{aligned}X(k) &= \sum_{n=0}^{N-1} x(n) e^{j \frac{2\pi}{N} nk} = \\&= \sum_{n=0}^{N-1} e^{-j \frac{2\pi}{N} nk} = \begin{cases} N, k = 0, \\ 0, k \neq 0. \end{cases}\end{aligned}$$

Для выполнения ДПФ в примере 2 можно воспользоваться соотношением

$$\sum_{n=0}^{N-1} e^{-j \frac{2\pi}{N} n(k-m)} = \begin{cases} N, k = m, \\ 0, k \neq m. \end{cases}$$

**Свойства ДПФ:**

1. Линейность

Пусть  $x(n)$  и  $y(n)$  – последовательности длины  $N$  и

$$x(n) \rightarrow X(k),$$

$$y(n) \rightarrow Y(k),$$

тогда

$$ax(n) + by(n) \rightarrow aX(k) + bY(k).$$

## 2. Сдвиг

Пусть  $x_p(n)$  – периодическая последовательность с периодом  $N$  и

$$x_p(n) \rightarrow X_p(k),$$

тогда

$$x_p(n - n_0) \rightarrow X_p(k) \cdot e^{-j\frac{2\pi}{N}n_0k}.$$

## 3. Свойство симметрии

Если периодическая последовательность  $x_p(n)$  (с периодом  $N$ ) является вещественной, то ДПФ  $X_p(k)$  удовлетворяет условиям симметрии:

$$\begin{aligned}\operatorname{Re} X_p(k) &= \operatorname{Re} X_p(N - k), \\ \operatorname{Im} X_p(k) &= -\operatorname{Im} X_p(N - k), \\ |X_p(k)| &= |X_p(N - k)|, \\ \arg X_p(k) &= -\arg X_p(N - k).\end{aligned}$$

Это справедливо и для последовательности  $x(n)$  конечной длины  $N$ .

Если  $x_p(n) = x_p(N - n)$ , то  $X_p(k)$  – только вещественная.

**Круговая (циклическая) свертка.** Если  $x_p(n)$  и  $h_p(n)$  – две периодические последовательности с периодами  $N$  отсчетов и ДПФ равными

$$\begin{aligned}X_p(k) &= \sum_{n=0}^{N-1} x_p(n) e^{-j\frac{2\pi}{N}nk}, \\ H_p(k) &= \sum_{n=0}^{N-1} h_p(n) e^{-j\frac{2\pi}{N}nk},\end{aligned}$$

то  $N$ -точечное ДПФ последовательности  $y_p(n)$ , являющейся круговой (циклической) сверткой последовательностей  $x_p(n)$  и  $h_p(n)$

$$\begin{aligned}y_p(n) &= \sum_{m=0}^{N-1} x_p(m) h_p(n - m), \\ y_p(n) &= x_p(n) \otimes h_p(n),\end{aligned}$$

равна

$$Y_p(k) = H_p(k) X_p(k)$$

**Вычисление линейной свертки конечных последовательностей с использованием ДПФ.** Пусть последовательность  $x(n)$ ,  $n = 0, 1, \dots, N_1-1$  поступает на вход ЛДС с импульсной характеристикой  $h(n)$ ,  $n = 0, 1, \dots, N_2-1$ . Выходная последовательность  $y(n)$  вычисляется по формуле свертки

$$y(n) = \sum_{m=0}^n x(m)h(n-m)$$

и имеет длину  $N_1+N_2-1$

Для вычисления линейной свертки с помощью циклической нужно дополнить  $x(n)$  и  $h(n)$  нулевыми отсчетами до длины  $N_1+N_2-1$  и рассматривать их как периодические. Далее следует вычислить  $X_p(k)$ ,  $H_p(k)$ ,  $X_p(k)H_p(k)$  и выполнить обратное ДПФ.

Данный метод вычисления свертки называется *быстрым*, потому что можно использовать быстрые алгоритмы вычисления ДПФ (эффективно при  $N \geq 30$ ).

В пакете MATLAB существует функция `fft`, которая производит вычисление ДПФ по быстрым алгоритмам (БПФ):

```
y=fft(x);
y=fft(x,n);
```

где  $x$  – вектор, содержащий последовательность отсчетов в сигнале,  $y$  – вектор, который содержит результаты преобразования, причем  $x$  и  $y$  одинаковы по длине. Во втором выражении вычисляется  $n$ -точечное БПФ. Если длина вектора  $x < n$ , то вектор  $x$  дополняется нулями, пока его длина не будет равна  $n$ . Если  $x > n$ , то отсчеты, которые удлиняют вектор  $x$  по сравнению с  $n$ , будут отброшены.

Если  $x=[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$ , то при вычислении БПФ

```
y=fft(x)
```

на выходе будет результат:

```
y=
```

```
8 0 0 0 0 0 0 0.
```

Если  $n = 4$ , то при выполнении

```
y=fft(x,n)
```

на выходе будет результат

```
y=
```

```
4 0 0 0.
```

Функция `fft` является встроенной, т. е. ее нельзя увидеть среди внешних *m*-файлов пакета MATLAB.

Обратное БПФ вычисляет функция `ifft`:

```
y=ifft(x);  
y=ifft(x,n).
```

Известно, что свертка во временной области соответствует умножению в частотной области. Покажем это в примере, используя функции `fft` и `conv`.

```
x=[ 1 2 3 4 ]; %Задаем вектор x.  
y=[ 5 4 3 2 1 ]; %Задаем вектор y.  
zt=conv(x,y); %Свертка x и y.  
x=[ x 0 0 0 0 ]; %Удлиняем x на 4 нулевых отсчета.  
y=[ y 0 0 0 ]; %Удлиняем y на 3 нулевых отсчета.  
X=fft(x); %Вычисление БПФ вектора x.  
Y=fft(y); %Вычисление БПФ вектора y.  
XY=X.*Y; %Умножение векторов в частотной  
%области.  
zf=real(ifft(XY)); %Нахождение действительной  
%части от обратного БПФ вектора XY.  
zf=round(zf); %Округление.  
zf=  
5 14 26 40 30 20 11 4  
zt=  
5 14 26 40 30 20 11 4
```

ДПФ последовательности, как правило, комплексные функции, причем действительные части ДПФ – четные функции (симметричные последовательности)  $\text{Re } X(k) = \text{Re } X(N-k)$ , а мнимые части – нечетные функции (антисимметричные последовательности)  $\text{Im}X(k) = -\text{Im}X(N - k)$ . ДПФ симметричной последовательности  $x(n)=x(N - n)$  является вещественной функцией.

Число операций, необходимых для вычисления ДПФ: комплексных сложений  $(N-1)N$ , комплексных умножений  $N \times N$ .

Алгоритмы быстрого преобразования Фурье (БПФ) позволяют уменьшить количество операций до порядка  $N \log_2 N$  или даже  $N$ .

## Алгоритм БПФ с прореживанием по времени

Пусть задана последовательность  $x(n)$ ,  $n = 0, 1, 2, \dots, N - 1$ .  $N = 2^l$ . Разобьем ее на две части, выделяя четные и нечетные элементы:

$$\begin{aligned} \{x(2^*n)\} &= \{x(0), x(2), \dots, x(N-2)\} \text{ и} \\ \{x(2^*n+1)\} &= \{x(1), x(3), \dots, x(N-1)\}. \end{aligned}$$

Выражение для дискретного преобразования Фурье:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{(2n+1)k} = \\ &= \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1)W_N^{2nk}. \end{aligned}$$

Так как  $x(n)$  имела длину  $N$ ,  $X(k)$  имеет период  $N$ .

$$W_N^{2nk} = W_{N/2}^{nk}.$$

При  $k \geq \frac{N}{2}$  можно записать

$$\begin{aligned} W_N^{i+\frac{N}{2}} &= \left( e^{-j\frac{2\pi}{N}} \right)^{i+\frac{N}{2}}, \quad i = 0, \dots, \frac{N}{2} - 1; \\ W_N^{i+\frac{N}{2}} &= W_N^i W_N^{\frac{N}{2}} = -W_N^i. \end{aligned}$$

Обозначим

$$G(k) = \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk}, \quad H(k) = \sum_{n=0}^{N/2-1} x(2n+1)W_N^{2nk},$$

тогда

$$\begin{aligned} X(k) &= G(k) + W_N^k \cdot H(k), \quad 0 \leq k \leq \frac{N}{2} - 1, \\ X(k) &= G(k) - W_N^{(N/2-k)} \cdot H(k), \quad \frac{N}{2} \leq k \leq N - 1, \end{aligned}$$

т.е.  $N$ -точечное ДПФ вычисляется через два  $N/2$ -точечных.

**Пример.**  $N=8$ . 8-точечное ДПФ можно вычислить через два 4-точечных (рис. 3.1).

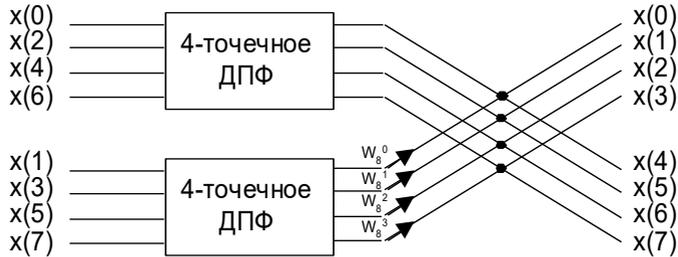


Рис. 3.1. Представление 8-точечного ДПФ через два 4-точечных

Операция «бабочки» для ДПФ с прореживанием по времени имеет вид (рис. 3.2):

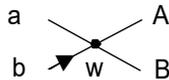


Рис. 3.2. Операция «бабочки» БПФ с прореживанием по времени,  
 $A = a + Wb$ ,  $B = a - Wb$ .

Соответственно 4-точечное ДПФ можно вычислить через два 2-точечных и т.д. (рис. 3.3).

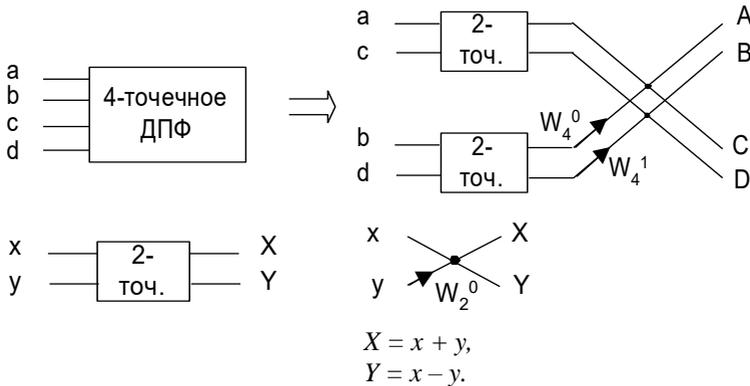


Рис. 3.3. Представление 4-точечного ДПФ через два 2-точечных

В результате многократного прореживания входные отсчеты располагаются в двоично-инверсном порядке. Поэтому двоичное реверсирование является первым шагом для ДПФ с про-

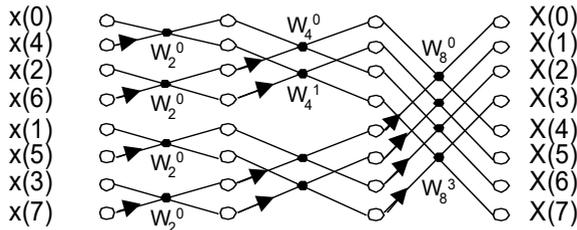
реживанием по времени (табл. 3.1).

Таблица 3.1

Номера отсчетов на входе алгоритма БПФ после прореживания

Номера отсчетов			
исходные		после прореживания	
десятичные	двоичные	двоичные	десятичные
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Итоговый граф алгоритма имеет вид, представленный на рис. 3.4.



$$W_N^0 = 1, W_4^1 = W_8^2.$$

Рис. 3.4. Граф алгоритма БПФ с прореживанием по времени

Граф алгоритма имеет регулярную структуру и позволяет проводить вычисление с замещением, т.е. записывать промежуточные результаты на место исходных.

Оценим вычислительную сложность: комплексных сложений  $\log_2 N \frac{N}{2} 2$ ; комплексных умножений  $\log_2 N \frac{N}{2}$ .

## Матричная трактовка алгоритма БПФ

По определению ДПФ ( $N = 8$ ):

$$\bar{Y} = \bar{W}\bar{X},$$

$$\begin{bmatrix} y(0) \\ y(1) \\ \dots \\ y(7) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^8 & W^{10} & W^{12} & W^{14} \\ \dots & \dots \\ W^0 & W^7 & W^{14} & W^{21} & W^{28} & W^{35} & W^{42} & W^{49} \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ \dots \\ x(7) \end{bmatrix},$$

где  $W = e^{-j 2\pi / 8}$ ;  $W_8^{n+8k} = W_8^n$ ;  $W^{k+N/2} = -W^k$ .

В алгоритме БПФ матрица ДПФ разбивается на слабозаполненные матрицы:

$$\bar{W} = \bar{\Phi}_1 \cdot \bar{\Phi}_2 \cdot \bar{\Phi}_3 \cdot \bar{P},$$

где  $\bar{P}$  – матрица перестановки,

$\bar{\Phi}_1, \bar{\Phi}_2, \bar{\Phi}_3$  – матрицы вычислительного процесса на каждом шаге.

$$P = \begin{bmatrix} 10000000 \\ 00001000 \\ 00100000 \\ 00000010 \\ 01000000 \\ 00000100 \\ 00010000 \\ 00000001 \end{bmatrix}; \quad \Phi_1 = \begin{bmatrix} 1000 & 1 & 0 & 0 & 0 & 0 \\ 0100 & 0 & W^1 & 0 & 0 \\ 0010 & 0 & 0 & W^2 & 0 \\ 0001 & 0 & 0 & 0 & W^3 \\ 1000 & -1 & 0 & 0 & 0 \\ 0100 & 0 & -W^1 & 0 & 0 \\ 0010 & 0 & 0 & -W^2 & 0 \\ 0001 & 0 & 0 & 0 & -W^3 \end{bmatrix};$$

$$\Phi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & W^2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & W^2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -W^2 \end{bmatrix}; \quad \Phi_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Процедура разбиения матрицы на сомножители называется факторизацией. Алгоритм БПФ с матричной точки зрения является следствием факторизации матрицы ДПФ.

#### Алгоритм БПФ с прореживанием по частоте

В этом случае сначала все отсчеты делятся на части: первая часть – первые  $N/2$  отсчетов; вторая часть – остальные  $N/2$  отсчетов. Граф алгоритма соответствует алгоритму с прореживанием по времени, но с рассмотрением справа налево. Выходные отсчеты представляются в двоично-реверсивном виде.

Бабочка БПФ в этом случае имеет вид, представленный на рис. 3.5.



Рис. 3.5. Операция «бабочки» БПФ с прореживанием по частоте

#### Алгоритм БПФ Кули – Тьюки с произвольным основанием

Рассмотрим случай, когда число отсчетов входной последовательности  $N$  является составным, т.е.  $N = N_1 N_2$ . Преобразуем индексы по формуле

$$n = N_1 n_2 + n_1, \quad (1)$$

$$k = N_2 k_1 + k_2, \quad (2)$$

т.е. одномерную последовательность представим в виде двумерной таблицы:

$$n_1, k_1 = 0, \dots, N_1 - 1;$$

$$n_2, k_2 = 0, \dots, N_2 - 1.$$

Тогда ДПФ будет иметь вид:

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk} = X(N_2 k_1 + k_2) = \\
 &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(N_1 n_2 + n_1) W_{N_2}^{n_2 k_2} W_{N_1}^{n_1 k_1} W_{N_1}^{n_1 k_2}.
 \end{aligned}$$

Алгоритм можно представить в виде последовательности шагов:

- 1) отображение входной последовательности в двумерную таблицу в соответствии с (1);
- 2) вычисление  $N_1$  ДПФ длиной  $N_2$  точек (внутренняя сумма);
- 3)  $N$  умножений на так называемый поворачивающий множитель  $W_{N_1}^{n_1 k_1}$ ;
- 4) вычисление  $N_2$  ДПФ длиной  $N_1$  точек (внешняя сумма);
- 5) отображение таблицы в выходную последовательность в соответствии с (2).

Пример. Пусть  $N = 6 = 3 \cdot 2$ , тогда  $N_1 = 3$  – число столбцов;  $N_2 = 2$  – число строк в таблице.

Шаги алгоритма:

- 1) преобразование индексов на входе (рис. 3.6);
- 2) ДПФ каждого столбца (три 2-точечных);

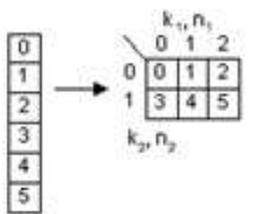


Рис. 3.6. Шаг 1. Отображение входной последовательности в двумерную таблицу

- 3) поэлементное умножение всех элементов таблицы на  $W_{N_1}^{n_1 k_1}$  (рис. 3.7);

	$n_1$	0	1	2
$k_2$	0	$W_6^0$	$W_6^0$	$W_6^0$
	1	$W_6^0$	$W_6^1$	$W_6^2$

Рис. 3.7. Шаг 3. Умножение на поворачивающий множитель

- 4) ДПФ каждой строки (два 3-точечных ДПФ);
- 5) преобразование индексов на выходе

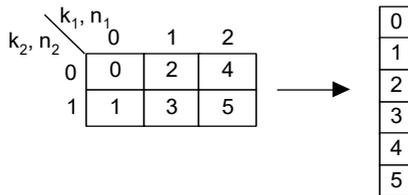


Рис. 3.8. Шаг 5. Отображение двумерной таблицы в выходную последовательность

Вся работа ведется с одной и той же таблицей.

Число операций комплексного умножения:  $N_1 N_2 (N_1 + N_2 + 1)$ ,  
 комплексного сложения:  $N_1 N_2 (N_1 + N_2 - 2)$ .

Чем больше  $N$ , тем больше выигрыш во времени в БПФ по сравнению с ДПФ.

Если ДПФ вычисляется с помощью алгоритмов БПФ, то число комплексных умножений –  $N_1 M(N_2) + N_2 M(N_1) + N$ , а число комплексных сложений –  $N_1 A(N_2) + N_2 A(N_1)$ , где  $M(N)$  и  $A(N)$  – число операций комплексного умножения и комплексного сложения в алгоритме БПФ.

Если множители  $N_1, N_2$  сами составные числа, то процедуру Кули – Тьюки можно продолжить рекурсивно. Например:  $N = 2^t$ . Если выберем  $N_1 = 2, N_2 = 2^{t-1}$ , то получим алгоритм с прореживанием по времени, а если выберем  $N_1 = 2^{t-1}, N_2 = 2$ , то получим алгоритм с прореживанием по частоте.

### Задание к лабораторной работе

Выполнить вариант задания, соответствующий порядковому номеру студента в группе.

**Варианты 1-15. Быстрые алгоритмы дискретного преобразования Фурье.** Программно реализовать БПФ и ДПФ по определению. Сравнить вычислительную сложность алгоритмов (число операций умножения и сложения). Выполнить обратное БПФ.

1. Алгоритм Кули – Тьюки по основанию 2.

2. Алгоритм Кули – Тьюки по основанию 4. Он может быть использован в случаях, когда длина преобразования  $N$  равна степени 4, и получается ее разложением в виде  $4 \cdot 4^{m-1}$ .

3. Алгоритм Кули – Тьюки двумерного БПФ. Двумерное БПФ можно вычислить с помощью одномерного БПФ в соответствии с выражением:

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) W_{N_1}^{n_1 k_1} W_{N_2}^{n_2 k_2} = \sum_{n_1=0}^{N_1-1} \left[ \sum_{n_2=0}^{N_2-1} x(n_1, n_2) W_{N_2}^{n_2 k_2} \right] W_{N_1}^{n_1 k_1} = \\ = \text{ДПФ}_{n_1} (\text{ДПФ}_{n_2} [x(n_1, n_2)]).$$

Итак, двумерное преобразование Фурье можно вычислить следующим образом: сначала выполняют одномерное БПФ по строкам (по столбцам), после чего выполняют одномерное БПФ со столбцами (строками) преобразованного массива.

4. Алгоритм 200-точечного БПФ на основе алгоритма Кули-Тьюки ( $200=25 \cdot 8$ ,  $25=5 \cdot 5$ ). 8-точечное БПФ вычислять с прореживанием по частоте, 5-точечное - по определению.

5. Алгоритм 24-точечного БПФ на основе алгоритма Кули-Тьюки ( $24=8 \cdot 3$ ). 8-точечное БПФ вычислять с прореживанием по времени, 3-точечное – по определению.

6. Алгоритм 320-точечного БПФ на основе алгоритма Кули-Тьюки ( $320=64 \cdot 5$ ). 64-точечное БПФ вычислять с прореживанием по времени, 5-точечное – по определению.

7. Алгоритм 56-точечного БПФ на основе алгоритма Кули-Тьюки ( $56=8 \cdot 7$ ). 8-точечное БПФ вычислять с прореживанием по времени, 7-точечное – по определению.

8. Алгоритм 15-точечного БПФ на основе алгоритма Кули-Тьюки ( $15=5 \cdot 3$ ). 5- и 3-точечные БПФ вычислять по определению.

9. Алгоритм 625-точечного БПФ на основе алгоритма Кули-Тьюки ( $625 = 25 \cdot 25$ ,  $25 = 5 \cdot 5$ ). 5-точечное БПФ вычислять по определению.

10. Алгоритм 640-точечного БПФ на основе алгоритма Кули-Тьюки ( $640=32 \cdot 20$ ,  $20=5 \cdot 4$ ). 32- и 4-точечные БПФ вычислять с прореживанием по времени, 5-точечное – по определению.

11. Алгоритм 400-точечного БПФ на основе алгоритма Кули-Тьюки (400=20·20, 20=5·4). 4-точечное БПФ вычислять с прореживанием по частоте, 5-точечное - по определению.

12. Алгоритм 720-точечного БПФ на основе алгоритма Кули-Тьюки (720=16·45, 45=5·9). 16-точечное БПФ вычислять с прореживанием по времени, 5- и 90-точечные – по определению.

13. **Алгоритм Гуда – Томаса.** Алгоритм Гуда – Томаса представляет собой способ отображения линейной последовательности из  $N = N_1N_2$  целых чисел в  $N_1N_2$ -таблицу, преобразующего одномерное преобразование Фурье в двумерное.

Числа  $N_1$  и  $N_2$  должны быть взаимно простыми, а основой отображения линейной последовательности в таблицу служит китайская теорема об остатках.

Сначала из линейного диофантова уравнения  $PN_1 + QN_2 = 1$  определяют два числа  $P$  и  $Q$ .

Входные индексы  $i$  вычисляются по формуле

$$i = i_1QN_2 + i_2PN_1 \pmod{N}.$$

В полученной таблице выполняется двумерное БПФ (см. задание 3). Выходные индексы  $k$  вычисляются по правилу

$$k = N_2k_1 + N_1k_2 \pmod{N}.$$

Пример.  $N = 6 = 3 \cdot 2$ ; тогда  $N_1 = 2$  – число строк в таблице,  $N_2 = 3$  – число столбцов. Из линейного диофантова уравнения  $2P + 3Q = 1$  находим  $P = 2$ ,  $Q = -1$ . Тогда переиндексация по методу Гуда – Томаса будет иметь вид, представленный на рис. 3.9.

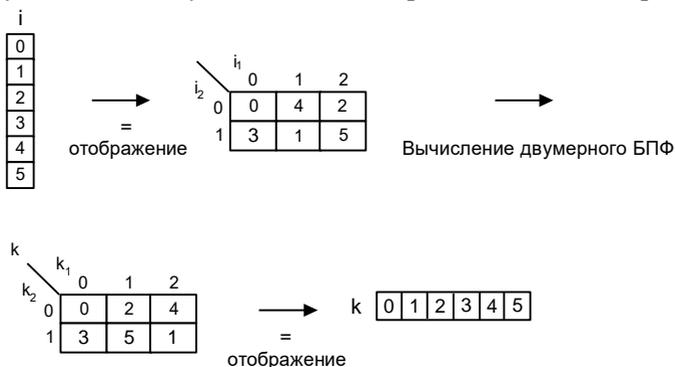


Рис. 3.9. Переиндексация элементов по алгоритму Гуда – Томаса

**14. Алгоритм Герцеля.** Он не принадлежит к алгоритмам БПФ, так как его сложность по-прежнему пропорциональна  $N^2$ . Алгоритм Герцеля полезен в тех случаях, когда требуется вычислить малое число компонент преобразования Фурье, как правило, не более чем  $\log_2 N$  из  $N$  компонент.

Для вычисления одной компоненты преобразования Фурье

$$X(k) = \sum_{n=0}^{N-1} x(n)W^{nk}$$

рассмотрим многочлен

$$p(z) = (z - W^k)(z - W^{-k}).$$

Он представляет собой многочлен наименьшей степени с вещественными коэффициентами, для которого элемент  $W^k$  является корнем. Это минимальный многочлен элемента  $W^k$  над полем вещественных чисел, и он равен

$$p(z) = z^2 - 2\cos(2\pi k/N)z + 1.$$

Пусть  $x(z) = \sum_{i=0}^{N-1} x(i)z^i$  и запишем

$$x(z) = p(z)Q(z) + r(z).$$

Многочлен частное  $Q(z)$  и многочлен остаток  $r(z)$  могут быть найдены с помощью алгоритма деления многочленов. Тогда величина  $X(k)$  может быть вычислена по остатку согласно равенству  $X(k) = x(W^k) = r(W^k)$ , так как по построению величина  $p(W^k)$  равна нулю.

**15. Алгоритм Блюстейна.** Алгоритм Блюстейна переводит  $N$ -точечное преобразование Фурье в  $N$ -точечную свертку и  $2N$  дополнительных умножений. Он описывается равенством

$$X(k) = \beta^{-k^2} \sum_{i=0}^{N-1} \beta^{(i-k)^2} (\beta^{-i^2} x(i)),$$

где  $\beta = \sqrt{e^{-j\frac{2\pi i}{N}}}$ .

Алгоритм содержит  $N$  поточечных умножений  $x(i)$  на  $\beta^{-i^2}$ , циклическую свертку с  $\beta^{i^2}$  и следующие за этим  $N$  поточечных

умножений на  $\beta^{-k^2}$ . Поэтому полное число операций по-прежнему имеет порядок  $N^2$ , так что алгоритм не эффективней прямого вычисления преобразования Фурье. Однако в некоторых приложениях он допускает более простую аппаратную реализацию. Кроме того, прямое вычисление свертки можно заменить алгоритмами быстрой свертки.

**Варианты 16-20.** Получить матрицу (или базисные функции) преобразования. Выполнить прямое и обратное преобразования.

16. **Дискретное преобразование Адамара** и быстрое преобразование Адамара. Матрицей Адамара называется ортогональная квадратная матрица порядка  $N$ , элементами которой являются действительные числа  $\pm 1$ . Матрица Адамара порядка  $N$  обозначается  $\mathbf{H}_N$ .

Простейшей матрицей Адамара является матрица второго порядка:

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Для построения матриц порядка  $N = 2^l$ ,  $l = 2, 3, \dots$  используется следующая теорема: если  $\mathbf{H}_N$  – матрица Адамара порядка  $N$ , то матрица

$$\mathbf{H}_{2N} = \begin{bmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{bmatrix}$$

является матрицей Адамара порядка  $2N$ .

Например, матрицы четвертого и восьмого порядков имеют соответственно следующий вид:

$$\mathbf{H}_4 = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & -\mathbf{H}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix};$$

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}_4 & -\mathbf{H}_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}.$$

Пусть  $\{s(n)\} = \{s(0), s(1), \dots, s(N-1)\}$  – совокупность равностоящих отсчетов сигнала. Выражения

$$b(k) = \sum_{n=0}^{N-1} s(n)(-1)^{(kn)}, \quad k=0, 1, \dots, N-1;$$

$$s(n) = N^{-1} \sum_{k=0}^{N-1} b(k)(-1)^{(kn)}, \quad n=0, 1, \dots, N-1$$

образуют пару дискретного преобразования Адамара в показательной форме. Используя матрицу Адамара порядка  $N$ , можно записать преобразование в матричной форме:

$$\mathbf{B} = \mathbf{H}\mathbf{S}; \quad \mathbf{S} = N^{-1}\mathbf{H}\mathbf{B},$$

где  $\mathbf{S} = [s(0), s(1), \dots, s(N-1)]^T$ ,  $\mathbf{B} = [b(0), b(1), \dots, b(N-1)]^T$  – векторы-столбцы отсчетов сигнала и спектральных коэффициентов.

Вычисление преобразования Адамара по определению требует выполнения  $N(N-1)$  операций сложения. Существуют быстрые алгоритмы, которые требуют только  $N \log_2 N$  операций. Для построения быстрого алгоритма рассмотрим матричное равенство

$$\mathbf{B} = \mathbf{H}_N \mathbf{S}.$$

Запишем его в виде

$$\begin{bmatrix} b(0) \\ b(1) \\ \cdot \\ \cdot \\ \cdot \\ b(N/2-1) \\ b(N/2) \\ \cdot \\ \cdot \\ \cdot \\ b(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{N/2} & \mathbf{H}_{N/2} \\ \mathbf{H}_{N/2} & -\mathbf{H}_{N/2} \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ \cdot \\ \cdot \\ \cdot \\ s(N/2-1) \\ s(N/2) \\ \cdot \\ \cdot \\ \cdot \\ s(N-1) \end{bmatrix},$$

что позволяет отдельно вычислить первую и вторую половины спектра. Получим

$$\begin{bmatrix} b(0) \\ b(1) \\ \cdot \\ \cdot \\ \cdot \\ b(N/2-1) \end{bmatrix} = \mathbf{H}_{N/2} \begin{bmatrix} s(0) + s(N/2) \\ s(1) + s(N/2+1) \\ \dots \\ s(N/2-1) + s(N-1) \end{bmatrix};$$

$$\begin{bmatrix} b(N/2) \\ b(N/2-1) \\ \cdot \\ \cdot \\ \cdot \\ b(N-1) \end{bmatrix} = \mathbf{H}_{N/2} \begin{bmatrix} s(0) - s(N/2) \\ s(1) - s(N/2+1) \\ \dots \\ s(N/2-1) - s(N-1) \end{bmatrix}.$$

Из этих выражений следует, что вычисление  $N$ -точечного преобразования сводится к предварительному суммированию (вычитанию) входных данных и последующему вычислению двух  $N/2$ -точечных преобразований

Так как  $N = 2^l$ , то процедуру снижения размерности преобразования можно продолжить до получения двухточечного пре-

образования. Для этого потребуется  $\log_2 N$  шагов. На каждом шаге производится  $N$  сложений, поэтому общее количество операций сложения равно  $N \log_2 N$ . Рассмотренный алгоритм называется *быстрым преобразованием Адамара* (БПА). Граф вычислительного процесса для  $N = 8$  имеет вид, представленный на рис. 3.10.

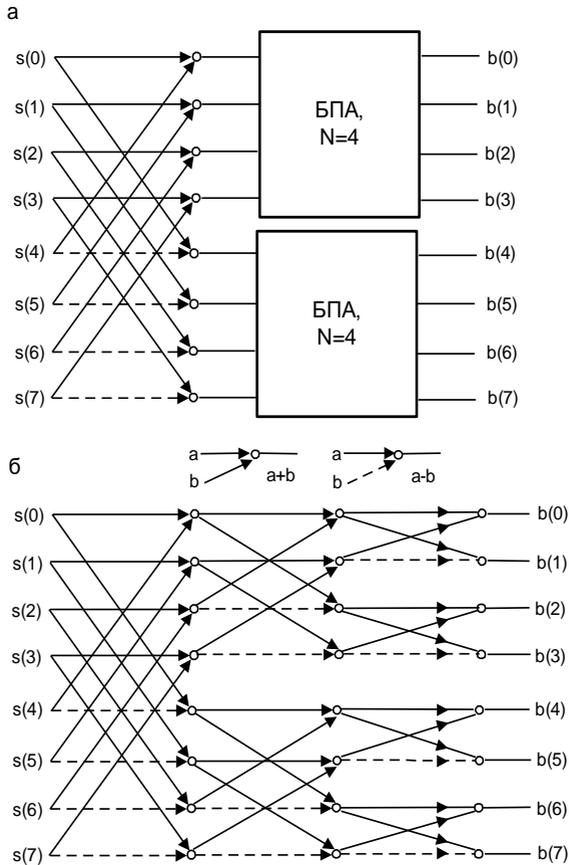


Рис. 3.10. Алгоритм быстрого преобразования Адамара

**17. Дискретное преобразование Хаара.** Функции Хаара  $\{\text{har}(r, k, t)\}$  образуют полную систему ортонормированных функций, определенных на интервале  $[0, N]$ ,  $N=2^l$ ,  $l = 1, 2, \dots$ , и

описываются следующими рекуррентными соотношениями:

$$\begin{aligned} \text{har}(0,0,t) &= 1, \quad t \in [0,N]; \\ \text{har}(r,k,t) &= \begin{cases} 2^{r/2}, N(k-1)2^{-r} \leq t < N(k-0,5)2^{-r} \\ -2^{r/2}, N(k-0,5)2^{-r} \leq t < Nk2^{-r} \\ 0, \text{ при остальных } t \in [0,N] \end{cases}; \end{aligned}$$

где  $0 \leq r < \log_2 N$  и  $1 \leq k < 2^r$ .

При дискретизации системы получают дискретные функции Хаара  $\{\text{har}(r,k,n)\}$ , определенные в целочисленных точках  $0, 1, 2, \dots, N-1$ .

Для  $N=8$  матрица этих функций имеет вид

$$\chi = \{\text{har}(r,k,n)\} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

С увеличением переменной  $r$  уменьшается интервал, на котором функция Хаара отлична от нуля. Это обстоятельство позволяет использовать разложение Хаара для получения спектральных коэффициентов, учитывающих локальное поведение функции.

Прямое и обратное дискретное преобразование Хаара вектора  $\mathbf{S}=[s(0), s(1), \dots, s(N-1)]^T$  определяется следующими соотношениями:

$$\mathbf{Y} = \chi \mathbf{S}; \quad \mathbf{S} = \chi^{-1} \mathbf{Y} = \mathbf{N}^{-1} \chi^T \mathbf{Y}.$$

Для вычисления этих преобразований существуют быстрые алгоритмы с числом операций сложения, равным  $2(N-1)$ , т.е. преобразование Хаара является самым быстрым среди всех рассмотренных ортогональных преобразований.

Граф преобразования для  $N=8$  имеет вид, представленный на рис. 3.11.

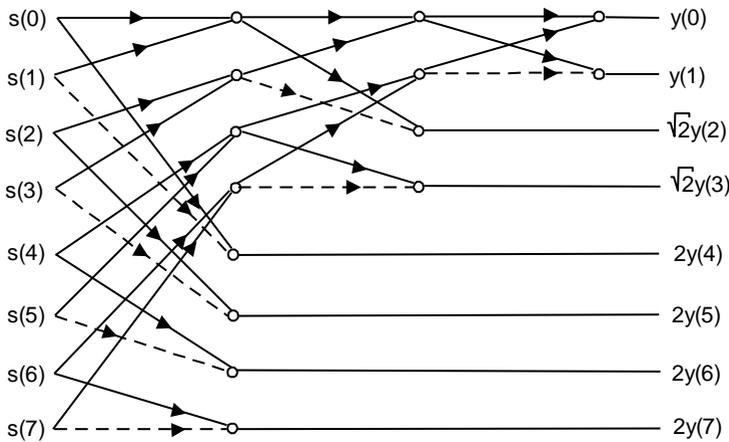


Рис. 3.11. Алгоритм быстрого преобразования Хаара

18. **Дискретное косинусное преобразование** (ДКП) определяется выражением:

$$y(k) = \sum_{n=1}^N 2x(n) \cos \left[ \frac{\pi}{2N} k(2n+1) \right], \quad k = 0, \dots, N-1,$$

где  $N$  – длина последовательности  $x$ ,  $x$  и  $y$  одинакового размера.

ДКП тесно связано с ДПФ. Двумерное ДКП применяется в алгоритмах сжатия изображений.

19. **Дискретное преобразование Хартли** (ДПХ). ДПФ отображает последовательность вещественных данных в комплексную область. Однако обработку вещественных данных желательно выполнять в вещественной области. Эту задачу решает ДПХ. Прямое и обратное ДПХ вещественной последовательности  $x(n)$  длины  $N$  определяются соотношениями:

$$H(k) = \sum_{n=0}^{N-1} x(n) \operatorname{cas} \left( \frac{2\pi nk}{N} \right); \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) \operatorname{cas} \left( \frac{2\pi nk}{N} \right),$$

где  $\operatorname{cas} \theta = \cos \theta + \sin \theta$ .

Между ДПФ и ДПХ существует простая прямая связь:

$$\operatorname{Re}\{X(k)\} = \frac{H(k) + H(-k)}{2}; \quad \operatorname{Im}\{X(k)\} = \frac{H(k) - H(-k)}{2}.$$

С другой стороны, по известным составляющим  $X(k)$  ДПФ можно получить ДПХ:  $H(k) = \operatorname{Re}\{X(k)\} - \operatorname{Im}\{X(k)\}$ .



4. Текст программы.
4. Результаты выполнения программы.
5. Выводы.

Тексты процедур БПФ с прореживанием по времени и частоте приведены в Приложении 3.

### **Вопросы для самоконтроля**

1. Дать определение прямого и обратного дискретного преобразования Фурье.
2. Для каких последовательностей определяют дискретное преобразование Фурье?
3. Описать свойства линейности, сдвига и симметрии ДПФ.
4. Дать определение круговой (циклической) свертки. Как связана круговая свертка с ДПФ?
5. Какой способ вычисления линейной свертки называется быстрым и почему?
6. Как от дискретно-временного преобразования Фурье (частотной характеристики) перейти к ДПФ?
7. Каково количество операций комплексного сложения и умножения при вычислении  $N$ -точечного ДПФ по определению?
8. Каков выигрыш по операциям комплексного умножения и сложения дают алгоритмы БПФ с прореживанием?
9. В чем суть операции двоичного реверсирования в алгоритмах БПФ с прореживанием?
10. На каком свойстве поворачивающих множителей основан алгоритм БПФ с прореживанием по времени?
11. Привести матрицу ДПФ для  $N = 8$ . Какова трактовка алгоритмов БПФ с матричной точки зрения?
12. В чем заключается операция бабочки в алгоритмах БПФ с прореживанием по времени и по частоте?
13. Какие этапы включает в себя алгоритм БПФ Кули – Тьюки?
14. Каков выигрыш по операциям комплексного умножения и сложения дает алгоритм БПФ Кули – Тьюки?

## ЛАБОРАТОРНАЯ РАБОТА 4 Цифровой спектральный анализ

### Теоретические сведения Задачи спектрального анализа

В задачах спектрального анализа считается априорно известным, что анализируемый сигнал  $x(nT)$  представляют собой сумму  $m$  синусоидальных составляющих и, возможно, белого шума  $x_m(nT)$

$$x(nT) = \sum_{j=1}^m a_j \sin(n\omega_j T + \varphi_j) + a_m x_m(nT), \quad (1)$$

где  $a_j$  – амплитуда  $j$ -й синусоидальной составляющей (при  $a_j = 0$  эта составляющая отсутствует);  $a_n = 0$  или 1 (при  $a_n = 0$  белый шум отсутствует).

Исходными данными для обработки являются  $N$  отсчётов сигнала  $x(nT)$ , т.е. время наблюдения этого сигнала равно  $\Theta = (N - 1)T$ .

Помимо  $N$  отсчётов обычно задаётся та или иная дополнительная информация о параметрах сигнала  $x(nT)$ : значение  $m$ , значения некоторых частот  $\omega_i$  или амплитуд  $a_i$  и т.д. Сформулируем ряд конкретных задач спектрального анализа, обнаружения и оценивания параметров сигнала.

Задача 1. Пусть в (1)  $m=1$ ,  $a_m = 0$ . Известна частота синусоидальной составляющей  $\omega_1$ . Требуется оценить неизвестные амплитуду  $a_1$  и фазу  $\varphi_1$  синусоидальной составляющей.

Задача 2. Пусть в (1)  $m > 1$ ,  $a_m = 0$ , известны частоты  $\omega_1, \omega_2, \dots, \omega_m$ . Требуется оценить неизвестные амплитуды  $a_1, a_2, \dots, a_m$  и фазы  $\varphi_1, \varphi_2, \dots, \varphi_m$ .

Задача 3. Пусть в (1)  $a_m = 1$ , т.е. в сигнале  $x(nT)$  имеется шумовая составляющая – белый шум с известной дисперсией  $\sigma^2$ ;  $m = 1$  известна  $\omega_1$ ,  $a_1 = 0$  или 1. Требуется определить  $a_1$  (т.е. принять решение о наличии или отсутствии синусоидальных составляющих).

Задача 4. Пусть в (1)  $m=2$ ,  $a_2 = ra_1, |\omega_1 - \omega_2| > \Delta\omega_1$ ,  $a_m = 0$ , заданы  $r \ll 1$ ,  $\omega_1$  и  $\Delta\omega_1$ , а величины  $\omega_2$ ,  $a_1$ ,  $a_2$ ,  $\phi_1$  и  $\phi_2$  неизвестны. Требуется оценить  $\omega_2$  и  $a_2$ .

Задачи 1-4 являются лишь примерами, существует много подобных задач. Общая черта: конечность времени наблюдения, для обработки доступны лишь  $N$  отсчётов сигналов.

Методы цифрового спектрального анализа можно разбить на 3 основные группы.

Метод цифровой фильтрации сводится к тому, что каждая из синусоидальных составляющих сигнала  $x(nT)$  выделяется отдельным фильтром. Выходной сигнал каждого фильтра подвергается дополнительной обработке, позволяющей принять решение или оценить амплитуду соответствующей  $\sin$  составляющей.

Метод, основанный на применении БПФ и оконных функций – один из самых распространённых методов цифрового анализа, сводится к вычислению ДПФ  $X(k\Omega)$  дискретной последовательности  $x(nT)$ ,  $k = 0, 1, \dots, N - 1$  и принятию соответствующего решения по значениям  $X(k\Omega)$ . Для улучшения качества обработки используются оконные функции  $w(n)$  – окно Хэмминга, Кайзера и т.д. (те же, что используются при проектировании КИХ-фильтров).

Методы линейного моделирования предусматривают вычисление параметров линейной модели БИХ-фильтра, квадрат АЧХ которого воспроизводит спектральную плотность мощности входного сигнала  $x(nT)$ . Эти методы используют в тех случаях, когда заранее почти ничего не известно о свойствах входного сигнала.

### **Дискретные случайные сигналы**

Ранее рассматривались детерминированные дискретные сигналы и системы. Они описываются с помощью  $z$ -преобразования и частотных характеристик (дискретно-временного преобразования Фурье).

Последовательности, которые имеют  $z$ -преобразование, должны обладать конечной энергией (требование сходимости  $z$ -преобразования).

Периодические последовательности, строго говоря,  $z$ -преобразования не имеют. Но такие последовательности могут быть однозначно определены параметрами одного периода (т.е. конечной длительности и, следовательно, конечной энергии), для представления периодической последовательности используют ДПФ.

Существует большое количество сигналов, которые либо не обладают конечной энергией, либо не являются периодическими. Например, сигналы систем связи имеют неограниченную длительность и лучше всего описываются с помощью сигналов бесконечной длительности и бесконечной энергии.

Другие примеры:

- речевые сигналы (для распознавания или сжатия);
- музыкальные сигналы (обработка для повышения качества).

Математически подобные сигналы представляются с помощью средних значений. Многие свойства подобных сигналов могут быть обобщены на основе последовательности с конечной энергией, называемой автокорреляционной или автоковариационной последовательностью, для которой часто существует  $z$ -преобразование или преобразование Фурье – спектральная плотность мощности. Обработка сигналов с бесконечной энергией в линейной дискретной системе может быть описана как результат воздействия системы на автоковариационную последовательность.

Рассмотрим основные понятия теории случайных сигналов. Множество случайных величин  $\{X(n)\}$  для  $-\infty < n < \infty$  вместе с вероятностным описанием каждой случайной величины определяет случайный процесс. При использовании случайного процесса в качестве модели случайного сигнала индекс  $n$  ассоциируется со временем или некоторой другой физической величиной (например, пространственной координатой). Данная последовательность  $\{X(n)\}$  при  $-\infty < n < \infty$  является реализацией случайного процесса. Множество реализаций случайного процесса называется ансамблем реализаций. Средним значением случайного процесса называется последовательность

$$m_x = E\{x(n)\},$$

автокорреляцией – последовательность

$$R_{xx}(n, n+m) = E\{x(n)x^*(n+m)\},$$

где  $E$  – оператор осреднения по ансамблю реализаций.

У стационарных процессов (в широком смысле) среднее значение  $m_x = \text{const}$ , а автокорреляционная последовательность  $R_{xx}(m) = E\{x(n)x^*(n+m)\}$ .

Автоковариационная последовательность стационарного случайного процесса определяется как

$$C_{xx}(m) = E\{(x(n) - m_x)(x^*(n+m) - m_x)\} = R_{xx}(m) - m_x^2$$

Физический смысл автокорреляции и автоковариации состоит в том, что они характеризуют меру зависимости между значениями случайного процесса в различные моменты времени.

Сигнал называется белым шумом, если  $R_{xx}(m) = \sigma_x^2 \delta(m)$ , т.е. отсчёты сигнала некоррелированы и  $m_x = 0$ .

Меру зависимости между двумя различными случайными сигналами дают взаимокорреляционная

$$R_{xy}(m) = E\{x(n)y(n+m)\}$$

и взаимоковариационная

$$C_{xy}(m) = E\{(x(n) - m_x)(y^*(n-m) - m_y)\} = R_{xy}(m) - m_x m_y$$

последовательности.

Свойства автокорреляционной и автоковариационной последовательностей для действительных случайных процессов:

$R_{xx}(0) = E\{x^2(n)\}$  – среднее значение квадрата случайного процесса;

$$C_{xx}(0) = \sigma_x^2 = E\{(x(n) - m_x)^2\} = E\{x^2(n)\} - m_x^2;$$

если  $m_x = 0$ , то  $\sigma_x^2 = E\{x^2(n)\}$ ;

$$|R_{xx}(m)| \leq R_{xx}(0); |C_{xx}(m)| \leq C_{xx}(0);$$

$$R_{xx}(m) = R_{xx}(-m); C_{xx}(m) = C_{xx}(-m);$$

$$R_{xy}(m) = R_{xy}(-m); C_{xy}(m) = C_{xy}(-m);$$

$$\lim_{m \rightarrow \infty} R_{xx}(m) = m_x; \lim_{m \rightarrow \infty} C_{xx}(m) = 0.$$

Корреляция и ковариация являются аperiодическими последовательностями, которые затухают при больших  $m$ . Поэтому часто их можно представить с помощью  $z$ -преобразования.  $Z$ -преобразование  $R_{xx}(m)$  и  $R_{xy}(m)$  существует только тогда, ко-

гда  $m_x = 0 (m_y = 0)$ . В таком случае  $z$ -преобразование  $R_{xx}(m)$  равно  $z$ -преобразованию  $C_{xx}(m)$ .

Определим величину

$$P_{xx}(z) = \sum_{m=0}^{\infty} C_{xx}(m) z^{-m}.$$

Тогда из свойств автоковариационной последовательности следует, что (теорема Винера – Хинчина)

$$\sigma_x^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{xx}(\omega) d\omega, \text{ где } P_{xx}(\omega) = P_{xx}(e^{i\omega})$$

Напомним, что когда  $m_x = 0$ , дисперсия равна среднему квадрату или средней мощности. Поэтому область под  $P_{xx}(\omega)$  для  $-\pi \leq \omega \leq \pi$  оказывается пропорциональной средней мощности сигнала. По этой причине функция  $P_{xx}(\omega)$  называется спектром плотности мощности или просто спектром. Для действительного сигнала функция  $P_{xx}(\omega)$  является действительной, чётной и положительной.

### Оценка спектра мощности

На практике обычно принято считать, что стационарный случайный процесс является эргодическим (т.е. временные средние равным средним по ансамблю). Все средние могут быть вычислены из одной реализации случайного процесса длиной  $N$  отсчётов. Тогда выражения для оценки автокорреляционной последовательности принимают вид:

$$\hat{R}_{xx}(m) = \frac{1}{N - |m|} \sum_{n=0}^{N-|m|-1} x(n)x^*(n+m), \text{ где } |m| < N;$$

$$\hat{R}_{xx}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n)x^*(n+m).$$

Сами оценки являются случайными последовательностями.

Математическое ожидание  $E\left\{\hat{R}_{xx}(m)\right\} = R_{xx}(m);$

$$E\left\{\hat{R}_{xx}(m)\right\} = \frac{N - (m)}{N} R_{xx}(m).$$

Смещение =  $R_{xx}(m) \frac{m}{N}$  – смещённая оценка

Дисперсия оценок при увеличении  $N$  уменьшается. Смещение оценки  $E\left\{\hat{R}_{xx}(m)\right\}$  также уменьшается при увеличении  $N$ .

Поэтому обе оценки являются состоятельными.

В качестве оценки плотности спектра мощности рассмотрим последовательность

$$I_n(\omega) = \sum_{m=-(n-1)}^{n-1} \hat{R}_{xx}(m) e^{-j\omega m}.$$

Можно показать, что  $I_n(\omega) = \frac{1}{N} |X(e^{j\omega})|^2$ ,

где  $|X(e^{j\omega})| = \sum_{n=0}^{n-1} x(n) e^{-j\omega n}$ .

Оценка  $I_m(\omega)$  называется периодограммой. Она является смещённой оценкой  $R_{xx}$  (то же при использовании  $\hat{R}(m)$ ).

Так как периодограмма не является состоятельной оценкой спектра мощности (дисперсия оценки не стремится к нулю при увеличении длины выборки  $N$ ) и её поведение при увеличении  $N$  имеет крайне нежелательный характер (дисперсия пропорциональна квадрату спектра), то необходимо получить другие оценки.

В работах Бартлетта и Уэлча были предложены процедуры усреднения модифицированных периодограмм, которые используют для вычисления оценки спектра мощности БПФ.

Последовательность данных  $x(n)$ ,  $0 \leq n \leq N-1$  делят на  $V$  отрезков по  $M$  выборок каждый, так что  $N=V \cdot M$ , т.е. формирует отрезки  $x^{(i)}(n) = x(n+iM-M)$ ,  $0 \leq m \leq M-1$ ,  $1 \leq i \leq V$ , и вычисляют модифицированные периодограммы

$$I_m^{(i)}(\omega) = \frac{1}{MU} \left| \sum_{n=0}^{M-1} x^{(i)}(n) \cdot W(n) e^{-j2\pi n} \right|^2, \quad i = 1, 2, \dots, V,$$

где  $U = \frac{1}{M} \sum_{n=0}^{M-1} W^2(n)$ ,  $W(n)$  – временное окно (то, которое используется и при проектировании КИХ-фильтров).

Оценка спектра определяется выражением

$$\hat{P}_{xx}^W(\omega) = \frac{1}{V} \sum_{i=1}^V I_M^{(i)}(\omega).$$

Нормирующий множитель  $U$  необходим, чтобы оценка  $\hat{P}_{xx}^W(\omega)$  была несмещённой.

Устранение периодограмм приводит к тому, что дисперсия оценки при увеличении  $V$  стремится к 0, т.е. оценка становится состоятельной. Нормированная случайная ошибка определяется выражением  $E \cong \sqrt{\frac{1}{V}}$ .

Взвешивание отрезков данных с помощью окна уменьшает дисперсию оценки, но в то же время сглаживает спектр и ухудшает разрешающую способность.

Мы рассмотрели неперекрывающиеся отрезки  $x^{(i)}(n)$ . Можно выбирать отрезки с перекрытием, например, на 50% или 75%. Если рассматриваются неперекрывающиеся отрезки без окна, то получается периодограмма Бартлетта, если перекрывающиеся с окном – периодограмма Уэлча.

БПФ даёт эффективное средство вычисления оценок спектра мощности на равноудалённых нормированных частотах

$$\omega_k = \frac{2\pi}{M}k, k = 0, 1, \dots, M - 1.$$

При представлении частот в герцах:  $f_k = \frac{k}{M} \frac{1}{T}$ , где  $T$  – период дискретизации в секундах.

Таким образом, необходимо вычислить

$$\hat{P}_{xx}^n\left(\frac{2\pi}{M}k\right) = \frac{1}{V} \sum_{i=1}^V I_M^{(i)}\left(\frac{2\pi}{M}k\right), k = 0, 1, \dots, M - 1,$$

где 
$$I_M^{(i)}\left(\frac{2\pi}{M}k\right) = \frac{1}{MU} \left| \sum_{n=0}^{M-1} x^{(i)}(n)W(n)e^{-j\frac{2\pi}{M}kn} \right|^2,$$

при  $i = 1, 2, \dots, V$  и  $k = 0, 1, \dots, M-1$ .

Это можно сделать следующим образом:

Шаг 1. С помощью БПФ вычислить

$$X_M^{(i)}(k) = \sum_{n=0}^{M-1} x^{(i)}(n)W(n)e^{-j\frac{2\pi}{M}kn}, \quad k=0, 1, \dots, M-1 \text{ для каждого отрезка.}$$

Шаг 2. Затем вычислить  $|X_M^{(i)}(k)|^2$  для каждого отрезка.

Шаг 3. Сложить эти результаты ( $V$  оценок).

Шаг 4. Разделить на  $VMU$ . Эта процедура (очень простая!) даёт оценку спектру мощности, причём всегда неотрицательную.

Если задан  $T$  – интервал дискретизации случайного процесса и  $df$  – необходимая разрешающая способность спектрального анализа, то интервал наблюдения  $T_{\text{набл.}}$  и число  $M$  обрабатываемых на этом интервале отсчётов определяются как

$$T_{\text{набл.}} = \frac{k_0}{\Delta f}, \quad M = \frac{f_{\text{дискр}}}{df} k_0,$$

где  $k_0$  – коэффициент, зависящий от вида окна  $w(n)$  (табл. 4.1).

Если по значениям  $\hat{P}_{xx}(k)$  принятие решения в задаче обнаружения невозможно, то следует дополнить последовательность нулями, увеличив  $N$  в 2, 4 или 8 раз, что добавит в спектр промежуточные составляющие.

Таблица 4.1

Значения коэффициента  $k_0$

окно	прямоугольное	треугольное	Хемминга	Кайзера
$k_0$	1,21	1,78	1,81	1,99 ( $\alpha = 2$ ) 2,20 ( $\alpha = 2,5$ ) 2,39 ( $\alpha = 3$ ) 2,57 ( $\alpha = 3,5$ )

## Вычисление корреляционных оценок

БПФ можно эффективно использовать для вычисления оценки автокорреляции

$$\hat{R}_{xx}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n)x(n+m),$$

где  $|m| < N$ ;  $0 \leq m \leq M-1$ ,  $M \leq N$ .

Если из  $x(n)$  вычесть  $\hat{m}_x$ , то получим оценку автоковариации.

По свойству автокорреляции последовательности  $\hat{R}_{xx}(-m) = \hat{R}_{xx}(m)$ , т.е. вычисления нужно производить для  $m \geq 0$ .

Введем понятие круговой (периодической) корреляции:

$$r(m) = \sum_{n=0}^{N-1} x_p(n)x_p(n+m),$$

для которой выполняется равенство

$$r(m) = \text{ОДПФ}\{x_p(k)x_p^*(k)\} = \text{ОДПФ}\{|x_p(k)|^2\},$$

в котором  $X_p(k) = \text{ДПФ}\{x_p(n)\}$ .

Дополняя последовательность  $x(n)$  нулями до длины  $\lambda = N + M - 1$  и вычисляя  $L$ -точечное ДПФ, можно добиться того, чтобы значения круговой корреляции совпадали с автокорреляцией на интервале  $0 \leq m \leq M - 1$ .

Таким образом, можно вычислить  $\hat{R}_{xx}(m)$  при  $0 \leq m \leq M - 1$  следующим образом:

1. Сформируем  $L$ -точечную последовательность, добавляя к  $x(n)$   $M - 1$  нулей.

2. Вычислим  $L$ -точечное ДПФ

$$X(k) = \sum_{n=0}^{L-1} x(n)e^{-j\frac{2\pi}{L}kn}, \quad k = 0, 1, \dots, L - 1.$$

3. Вычислим  $L$ -точечное обратное ДПФ

$$V(m) = \frac{1}{L} \sum_{k=0}^{L-1} |X(k)|^2 e^{j\frac{2\pi}{L}km}, \quad m = 0, 1, \dots, L - 1.$$

4. Тогда получим  $\hat{R}_{xx}(m) = \frac{1}{N}V(m)$ ,  $m = 0, 1, \dots, L - 1$ .

Если  $M$  мало, то наиболее эффективным методом может быть расчёт по прямой формуле (количество операций пропорционально  $MN$ ).

При  $M \sim 100$  и более эффективна процедура с использованием БПФ (количество операций пропорционально  $L \log_2 L = (N+M)\log_2(N+M)$  и зависит от реализации алгоритма БПФ).

Как и при оценке спектра мощности, оценку автокорреляции можно производить, разбивая все реализацию случайного процесса на отрезки, вычисляя оценки автокорреляции на каждом отрезке и затем усредняя их.

### Задание к лабораторной работе

**Варианты 1-10.** Непрерывная реализация длиной  $T$  дискретизируется с частотой  $f_d$  и полученный временной ряд используется для построения оценки спектральной плотности с разрешающей способностью  $df$  и нормированной случайной ошибкой  $E$ . Построить алгоритм, составить и отладить программу оценки, определить необходимое количество операций вещественного сложения и вещественного умножения. Индивидуальные параметры взять из табл. 4.2.

Таблица 4.2

Параметры к вариантам 1-10.

Номер варианта	$T$ , с	$f_d$ , Гц	$df$ , Гц	$E$	Вид периодограммы	Вид окна
1	4	4096	16		Бартлетта	-
2	4	4096	16	?	Уэлча, перекрытие 25%	Треугольное
3	4	4096	16	?	Уэлча, перекрытие 50%	Хэмминга
4	4	4096	16	10%	Уэлча, перекрытие ?	Прямоугольное
5	4	4096	16	15%	Уэлча, перекрытие ?	Кайзера, $\alpha = 2$
6	8	4096	16	8%	Уэлча, перекрытие ?	Кайзера, $\alpha = 3$

Окончание табл. 4.2

Номер варианта	$T, c$	$f_d, Гц$	$df, Гц$	$E$	Вид периодограммы	Вид окна
7	8	8192	8		Бартлетта	-
8	8	8192	8	?	Уэлча, перекрытие 25%	Треугольное
9	8	8192	16	?	Уэлча, перекрытие 30%	Хэмминга
10	8	4096	8	9%	Уэлча, перекрытие ?	Прямоугольное

11. Периодограмма Даньелла. Если для вычисления выборочного спектра  $P_{xx}(f)$  на сетке частот  $f_k = k/KT$ ,  $0 \leq k \leq K-1$ , используется алгоритм БПФ, то модифицированная оценка периодограммы на частоте  $f_i$  может быть получена посредством усреднения в  $P$  точках (значений) с каждой стороны от этой частоты:

$$P_D(f_i) = 1/(2P+1) \sum_{n=i-P}^{i+P} P_{xx}(f_n)$$

Построить алгоритм вычисления оценки спектральной плотности мощности по Даньеллу. Составить и отладить программу оценки, определить необходимое количество операций вещественного сложения и вещественного умножения.

**Варианты 12-17.** Оценка автокорреляционной функции строится по реализации стационарного случайного процесса, содержащей  $N$  отсчетов. Расчет предполагается вести косвенным методом с использованием БПФ и при максимальном сдвиге  $m$ . Построить 2 алгоритма расчета: 1) по всей реализации длиной  $N$  отсчетов; 2) по  $V$  отрезкам длиной по  $M$  отсчетов. Составить и отладить программу оценки. Определить необходимое число операций над вещественными числами. Какой алгоритм требует меньшего числа операций? Индивидуальные параметры взять из табл. 4.3.

Таблица 4.3

Параметры к вариантам 12-17

Номер варианта	$N$	$m$	$V$	$M$
12	4096	128	32	128
13	4096	64	16	256
14	8192	128	64	128
15	4096	128	8	256
16	8192	128	16	512
17	8192	64	32	256

**Варианты 18-20.** Автокорреляционная функция оценивается по реализации стационарного случайного процесса, содержащей  $N$  отсчетов, при максимальном сдвиге  $m$  отсчетов. Расчет можно вести двумя методами: 1) прямым и 2) косвенным (с использованием БПФ). Построить оба алгоритма расчета. Составить и отладить программу оценки. Определить во сколько раз быстрее по сложению и умножению выполняются вычисления по косвенному методу, чем по прямому.

Таблица 4.4

Параметры к вариантам 18-20

№ варианта	$N$	$m$
18	4096	128
19	4096	64
20	8192	128

### Порядок выполнения работы

1. Изучить теоретические сведения по работе.
2. Составить схему реализации задания, выписать все используемые формулы. Исходя из разрешающей способности спектрального анализа и частоты дискретизации исследуемого сигнала, рассчитать необходимые параметры анализа: число отсчетов в сегменте, число сегментов, степень перекрытия и т.д.
3. Составить и отладить программу оценки спектра мощности или автокорреляции.
4. Подготовить в системе MATLAB наборы тестовых данных, каждый из которых представляет сумму нескольких синусоидальных составляющих и белого шума.

## Содержание отчета

1. Цель работы.
2. Текст задания.
3. Схема реализации, используемые формулы, особенности метода и применяемых окон.
4. Текст программы.
5. Результаты работы программы на тестовых примерах (графики оценок СПМ и автокорреляционной последовательности).
6. Выводы по работе.

Пример выполнения лабораторной работы приведен в прил.4.

## Вопросы для самоконтроля

1. Какие исходные данные используются при цифровом спектральном анализе?
2. Дать определения понятий: случайный процесс, реализация случайного процесса, ансамбль реализаций.
3. Какие случайные процессы называют стационарными, какие – эргодическими?
4. Дать математическое определение автокорреляционной и автоковариационной последовательностей стационарного случайного сигнала.
5. Каков физический смысл автокорреляционной и автоковариационной последовательностей?
6. Привести формулы оценки автокорреляционной последовательности для эргодического случайного процесса. Являются ли эти оценки состоятельными?
7. Дать математическое определение спектра плотности мощности стационарного случайного сигнала.
8. Дать определение периодограммы. Является ли периодограмма состоятельной оценкой спектральной плотности мощности?
9. Какие меры предложены Бартлетом и Уэлчем для того, чтобы периодограмма стала состоятельной оценкой спектра мощности?
10. Почему метод модифицированных периодограмм ухудшает разрешающую способность оценки спектра мощности?

11. С какой целью используются оконные функции в методе модифицированных периодограмм для оценки спектра мощности?

12. С какой целью используется перекрытие отрезков в методе модифицированных периодограмм для оценки спектра мощности?

13. Как определить необходимую длину отрезка в методе модифицированных периодограмм для оценки спектра мощности, если заданы частота дискретизации случайного сигнала  $F_d$  и разрешающая способность спектрального анализа  $df$ ?

14. Если задана нормированная случайная ошибка оценки в методе модифицированных периодограмм для оценки спектра мощности, как определить необходимое количество отрезков?

15. Какой размерности нужно выполнять БПФ при вычислении корреляционной оценки, если известны длина случайной последовательности  $N$  и максимальный сдвиг корреляции  $m$ ?

16. В чем суть метода спектрального анализа на основе линейного моделирования и когда его применяют?

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Сергиенко А.Б. Цифровая обработка сигналов. 3-е изд. СПб.: БХВ-Петербург, 2013. 756 с.

2. Солонина А.И., Улахович Д.А., Арбузов С.М., Соловьева Е.Е. Основы цифровой обработки сигналов. 2-е изд. СПб: ВНУ-Петербург, 2012. 768 с.

3. Бендат Дж., Пирсол А. Прикладной анализ случайных данных / пер. с англ. М.: Мир, 1989. 540 с.

4. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов / пер. с англ. М.: Мир, 1989. 448 с.

5. Гольденберг Л.М., Матюшкин Б.Д., Поляк М.Н. Цифровая обработка сигналов: справочник. М.: Радио и связь, 1985. 312 с.

6. Лосев В.В. Микропроцессорные устройства обработки информации. Алгоритмы цифровой обработки. Минск: Вышэйш. шк., 1990. 132 с.

7. Марпл С.Л. Цифровой спектральный анализ и его приложения / пер. с англ. М.: Мир, 1990. 584с.

8. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. 3-е изд. /пер. с англ. М: Техносфера, 2012. 1048 с.

9. Сиберт У. Цепи, сигналы, системы: в 2 ч./ пер. с англ. М.: Мир, 1988. 696 с.

10. Signal Processing Toolbox User's Guide. The MathWorks, Inc., 1994. 364 p.

## Приложения

### Приложение 1. Пример выполнения лабораторной работы 1 Базовые операции цифровой обработки сигналов

*Цель работы.* Изучение основных операций цифровой обработки сигналов, базирующихся на теории линейных дискретных систем. Знакомство с функциями пакета Signal Processing Toolbox системы MATLAB.

*Текст задания.* Теоретически и в системе MATLAB вычислить и построить частотные характеристики линейной дискретной системы (АЧХ, ФЧХ) с заданной передаточной функцией

$$H(z) = \frac{1}{1 - \frac{1}{6}z^{-1} - \frac{1}{6}z^{-2}}.$$

*Теоретическое выполнение задания.*

$$H(e^{jw}) = \frac{1}{1 - \frac{1}{6}e^{-jw} - \frac{1}{6}e^{-2jw}}.$$

$$e^{jw} = \cos w + j \sin w.$$

$$\begin{aligned} H(e^{jw}) &= \frac{1}{1 - \frac{1}{6}(\cos w - j \sin w) - \frac{1}{6}(\cos w - j \sin w)^2} = \\ &= \frac{1}{1 - \frac{1}{6} \cos w + j \frac{1}{6} \sin w - \frac{1}{6}(\cos^2 w - j 2 \sin w \cos w - \sin^2 w)} = \\ &= \frac{1}{1 - \frac{1}{6} \cos w + j \frac{1}{6} \sin w - \frac{1}{6} \cos^2 w + j \frac{1}{6} \sin 2w + \frac{1}{6} \sin^2 w} = \end{aligned}$$

$$= \frac{1}{1 - \frac{1}{6} \cos w + \frac{1}{6} \sin^2 w - \frac{1}{6} \cos^2 w + j \left( \frac{1}{6} \sin w + \frac{1}{6} \sin 2w \right)}$$

*умножим на комплексно-сопряженное значение знаменателя  
обе части дроби*

$$= \frac{1 - \frac{1}{6} \cos w + \frac{1}{6} \sin^2 w - \frac{1}{6} \cos^2 w - j \left( \frac{1}{6} \sin w + \frac{1}{6} \sin 2w \right)}{\left( 1 - \frac{1}{6} \cos w + \frac{1}{6} \sin^2 w - \frac{1}{6} \cos^2 w \right)^2 + \left( \frac{1}{6} \sin w + \frac{1}{6} \sin 2w \right)^2}$$

Частотные характеристики:

- АЧХ:

$$|H(e^{jw})| = \frac{\sqrt{\left( 1 - \frac{1}{6} \cos w + \frac{1}{6} \sin^2 w - \frac{1}{6} \cos^2 w \right)^2 + \left( \frac{1}{6} \sin w + \frac{1}{6} \sin 2w \right)^2}}{\sqrt{\left( 1 - \frac{1}{6} \cos w + \frac{1}{6} \sin^2 w - \frac{1}{6} \cos^2 w \right)^2 + \left( \frac{1}{6} \sin w + \frac{1}{6} \sin 2w \right)^2}} =$$

$$= \frac{1}{\sqrt{\left( 1 - \frac{1}{6} \cos w + \frac{1}{6} \sin^2 w - \frac{1}{6} \cos^2 w \right)^2 + \left( \frac{1}{6} \sin w + \frac{1}{6} \sin 2w \right)^2}}$$

- ФЧХ:

$$\arg(H(e^{jw})) = \arctg \left( \frac{\frac{1}{6} \sin w + \frac{1}{6} \sin 2w}{-\left( 1 - \frac{1}{6} \cos w + \frac{1}{6} \sin^2 w - \frac{1}{6} \cos^2 w \right)} \right)$$

*Проверка результата в системе MATLAB*

```
>> b = 1;
>> a = [1 -1/6 -1/6];
>> [H,w] = freqz(b,a);
>> A1 = abs(H);
>> Ph1 = angle(H);
>> n = length(w);
>> w = 0:pi/n:pi-pi/n;
>> x = 1-1/6*cos(w)+1/6*(sin(w)).^2-1/6*(cos(w)).^2;
>> y = 1/6*sin(w)+1/6*sin(2*w);
>> A2 = 1./(x.^2 + y.^2).^0.5;
>> Ph2 = atan(y./(-x));
>> subplot(2,2,1), plot(w,A1);
>> grid on; title('АЧХ с помощью MATLAB');
>> subplot(2,2,2), plot(w,Ph1);
>> grid on; title('ФЧХ с помощью MATLAB');
>> subplot(2,2,3), plot(w,A2);
>> grid on; title('АЧХ вычисленное вручную');
>> subplot(2,2,4), plot(w,Ph2);
>> grid on; title('ФЧХ вычисленное вручную');
```

### Полученные результаты

Полученные графики частотных характеристик приведены на рис. П1.

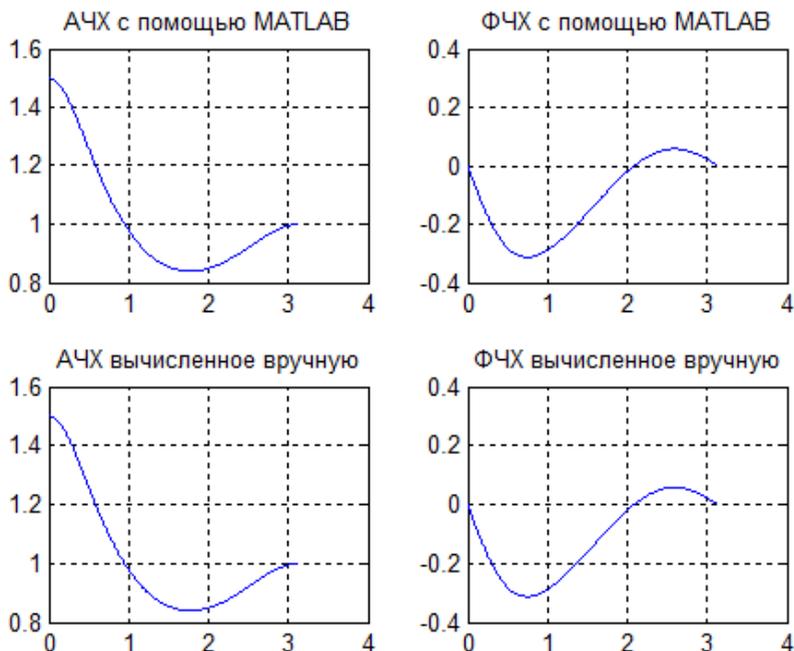


Рис. П1. Частотные характеристики

*Выводы.* Результаты, полученные аналитически и с помощью функций системы MATLAB, совпадают. Пакет Signal Processing Toolbox системы MATLAB предоставляет удобные функции для исследования процессов в системах цифровой обработки сигналов.

## Приложение 2. Пример выполнения лабораторной работы 2

### Проектирование, реализация и исследование характеристик БИХ-фильтра

*Цель работы.* Получение опыта проектирования и программной реализации цифровых фильтров.

#### Часть 1

*Задание.* Спроектировать БИХ-фильтр нижних частот Баттерворта. Характеристики фильтра:  $f_d = 2000$  Гц;  $r_p = 1.35$  дБ;  $r_s = 60$  дБ;  $f_p = 400$  Гц;  $f_s = 650$  Гц.

*Особенности заданного типа фильтра.* Фильтры Баттерворта характеризуются тем, что имеют максимально гладкую АЧХ в полосе пропускания.

*Текст программы для расчета параметров фильтра и его характеристик*

```
% Задаём характеристики фильтра
% частота дискретизации
Fd = 2000;
% частоты среза полосы пропускания и
% полосы непропускания
fp = 400;
fs = 650;
% величины затуханий
Rp = 1.35;
Rs = 60;
% Нормализуем частоты
fd2 = Fd / 2;
Wp = fp / fd2;
Ws = fs / fd2;

% Расчёт фильтра
% Вычисляем порядок фильтра и частоту среза
[n, Wn] = buttord(Wp, Ws, Rp, Rs);
disp('Порядок фильтра:');
disp('Частота среза');
% Вычисляем коэффициенты фильтра
[b, a]=butter(n, Wn);
disp('Коэффициенты b:');
disp('Коэффициенты a:');
```

```

% Построение частотных характеристик
% Вычисляем частотную характеристику
[h,w]=freqz(b,a);
% Задаем сетку частот
ff = w/pi*fd2;

% Построение АЧХ
figure(1)
plot(ff, abs(h));
title('Amplitude frequency characteristic');
xlabel('Frequency');
ylabel('Amplitude');

% Построение ФЧХ
figure(2)
plot(ff, unwrap(angle(h)));
title('Phase frequency characteristic');
xlabel('Frequency');
ylabel('Phase');

% Построение ЛАЧХ
figure(3)
plot(ff, 20 * log10(abs(h)));
title('Logarithmic amplitude frequency characteris-
tic');
xlabel('Frequency');
ylabel('Amplitude');

% Построение групповой задержки
figure(4)
gd = grpdelay(b, a);
plot(ff, gd);
xlabel('Frequency');
ylabel('delay');
title('Group delay');

% Построение импульсной и временной характеристик
% Импульсная характеристика
figure(5)
impz(b,a);

```

```

% Временная характеристика
figure(6)
stepz(b,a);

% Вычисление и построение реакции фильтра
% на синусоидальный сигнал в полосе пропускания
% и непропускания
% а) в полосе пропускания
nn = (0:99);
f1 = 300;
x1 = sin(2*pi*f1*nn/Fd);
y1 = filter(b, a, x1);
figure(7)
plot(nn,x1,nn,y1);
title('The bandwidth of the sinusoid');

% б) в полосе непропускания
f2 = 800;
x1 = sin(2*pi*f2*nn/Fd);
y1 = filter(b, a, x1);
figure(8)
plot(nn,x1,nn,y1);
title('The band nontransmitting sinusoid');

% Получаем коэффициенты последовательной и парал-
% лельной % форм реализации фильтра
% Построение последовательной формы фильтра
disp('Коэффициенты последовательной формы:')
[sos, g] = tf2sos(b,a)

% Построение параллельной формы фильтра
disp('Коэффициенты последовательной формы:')
[r p k] = residuez(b, a);
Bi=[ ; ];
Ai =[ ; ];

for i = 1: 2: floor(n/2)*2
    [bi ai] = residuez(r(i : i+1), p(i : i+1), 0);
    Bi((i+1)/2, :) = bi;
    Ai((i+1)/2, :) = ai;
end

```

### Полученные результаты

Коэффициенты  $b$  и  $a$  фильтра:

$b = [0.0010 \ 0.0099 \ 0.0447 \ 0.1193 \ 0.2088 \ 0.2506$   
 $0.2088 \ 0.1193 \ 0.0447 \ 0.0099 \ 0.0010];$

$a = [1.0000 \ -1.2664 \ 2.0182 \ -1.4914 \ 1.0924 \ -0.4860$   
 $0.1910 \ -0.0479 \ 0.0093 \ -0.0010 \ 0.0001].$

Порядок фильтра  $n = 10$ , частота среза  $\omega_n = 0.4346$ .

Коэффициенты последовательной формы:

$g = 0.0010$

$sos =$

1.0000	1.9993	0.9972	1.0000	-0.2016	0.0162
1.0000	2.0754	1.0776	1.0000	-0.2118	0.0676
1.0000	2.0296	1.0318	1.0000	-0.2344	0.1813
1.0000	1.9714	0.9736	1.0000	-0.2746	0.3841
1.0000	1.9242	0.9264	1.0000	-0.3440	0.7341

Коэффициенты параллельной формы:

$k = 17.715916510120348$

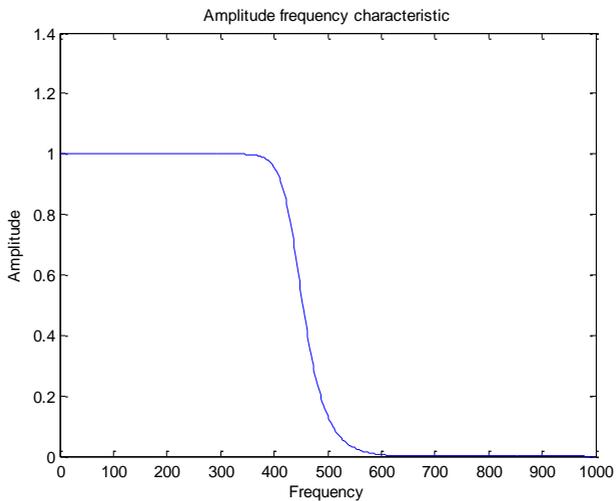
$B_i =$

-0.4294	0.4577	0.0000
3.3854	1.2055	0.0000
9.2900	-6.7739	0.0000
-44.4229	0.0671	0.0000
14.4620	8.5892	0.0000

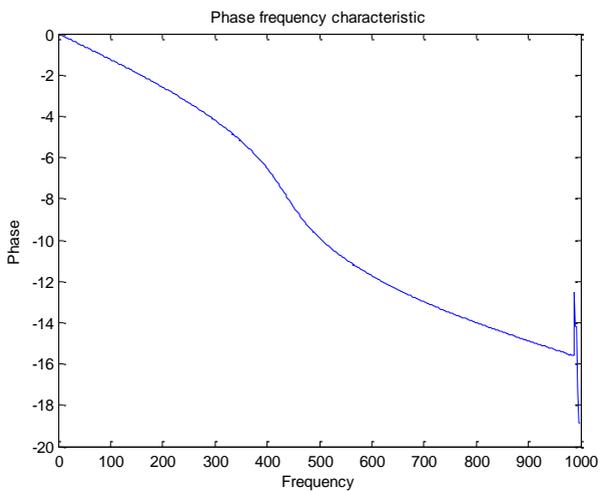
$A_i =$

1.0000	-0.3440	0.7341
1.0000	-0.2746	0.3841
1.0000	-0.2344	0.1813
1.0000	-0.2118	0.0676
1.0000	-0.2016	0.0162

Графики характеристик фильтра приведены на рис. П2.1 – П2.4, структурные схемы фильтров – на рис. П2.5 – П2.8.

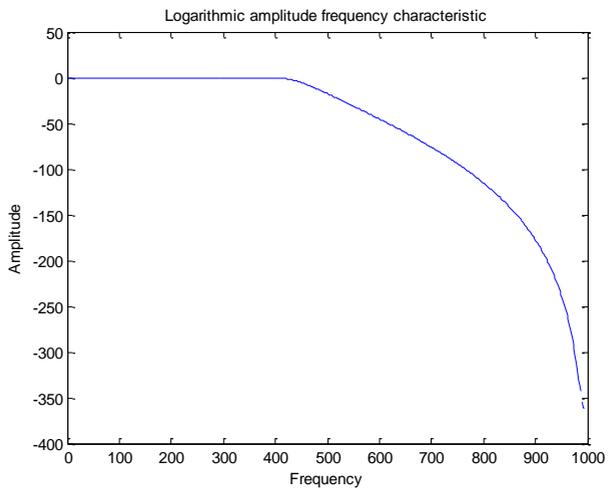


*a*

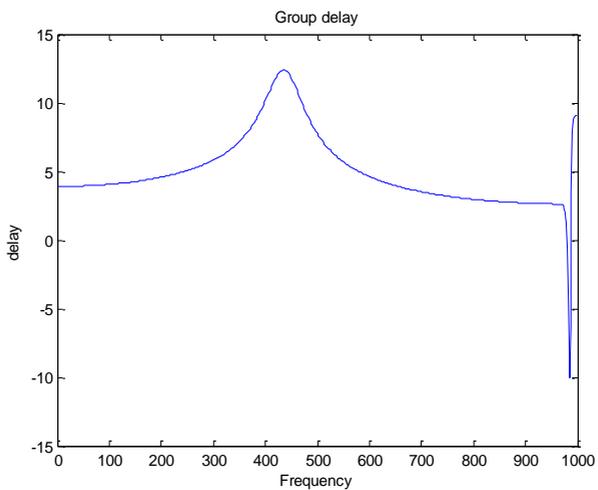


*б*

Рис. П2.1. Амплитудно-частотная (*a*) и фазочастотная (*б*) характеристики фильтра

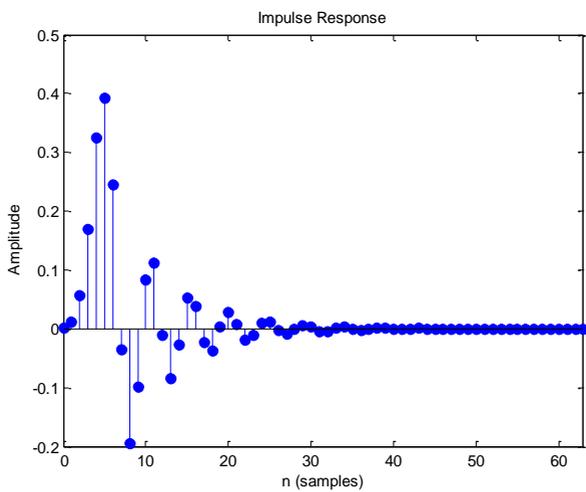


*a*

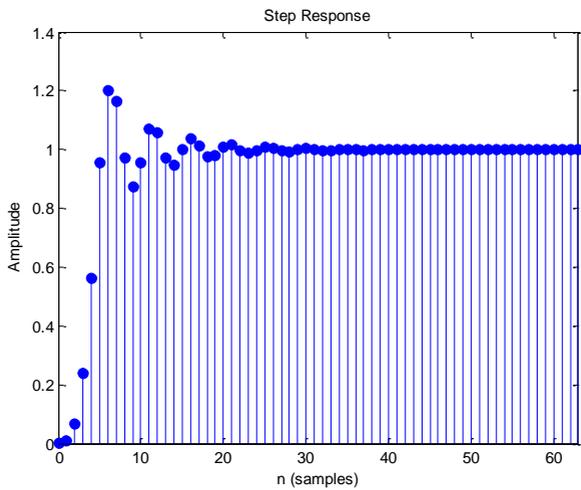


*b*

Рис. П2.2. ЛАЧХ (*a*) и групповая задержка (*b*) фильтра

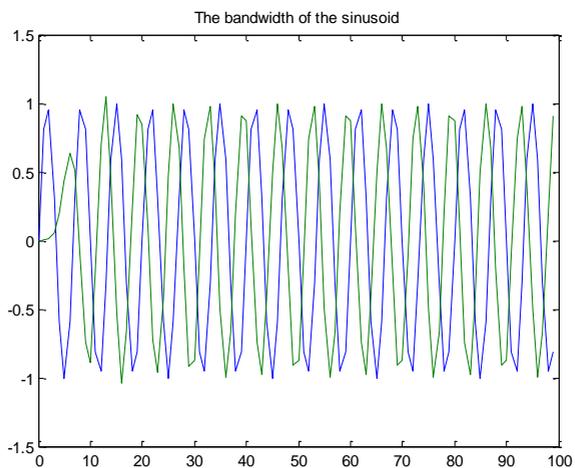


$a$

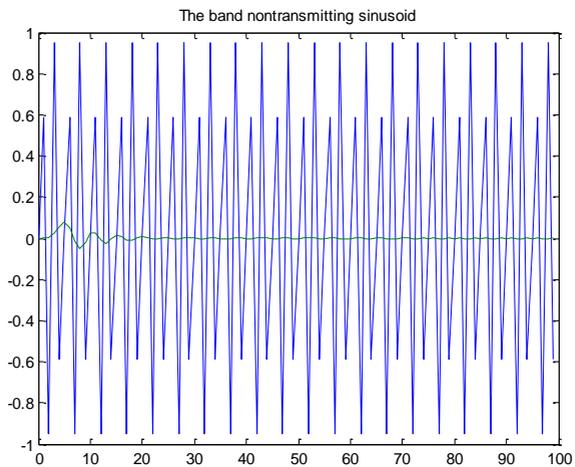


$b$

Рис. П2.3. Импульсная ( $a$ ) и временная ( $b$ ) характеристики фильтра



*a*



*б*

Рис. П2.4. Реакция фильтра на синусоидальный сигнал из полосы пропускания (*a*) и полосы непропускания (*б*)

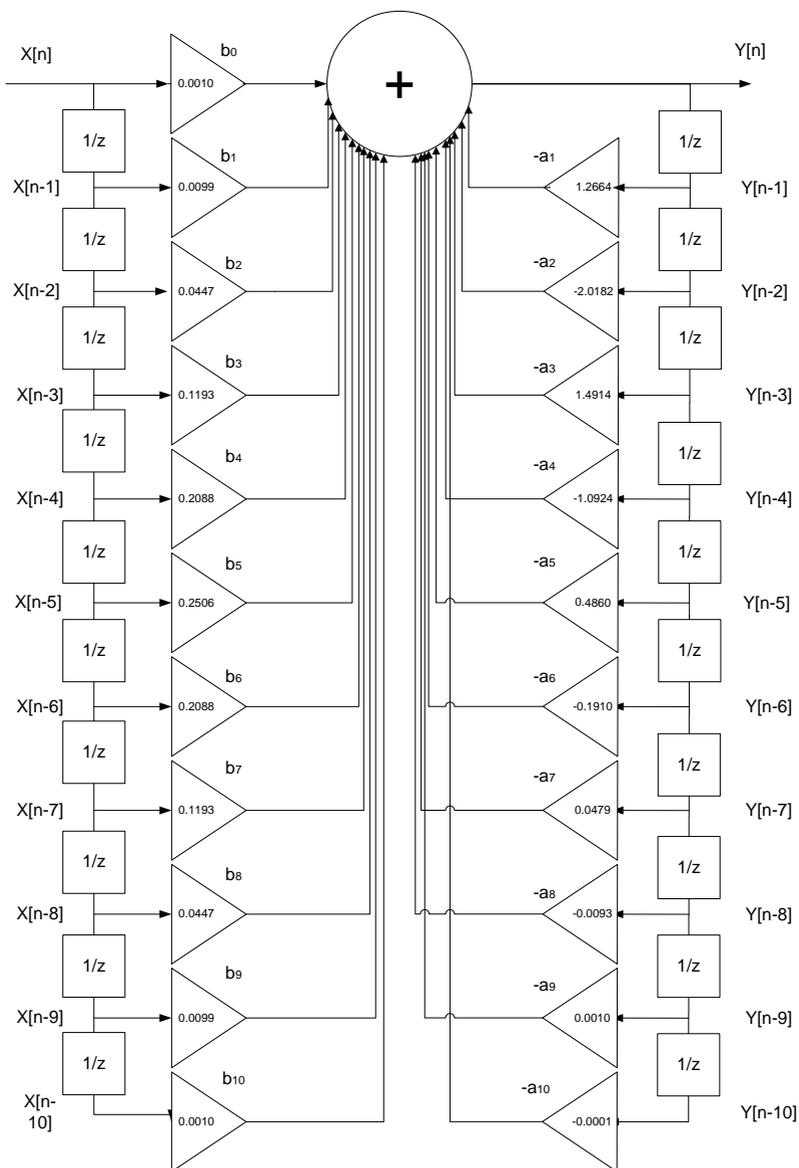


Рис. П2.5. Структурная схема фильтра в прямой форме

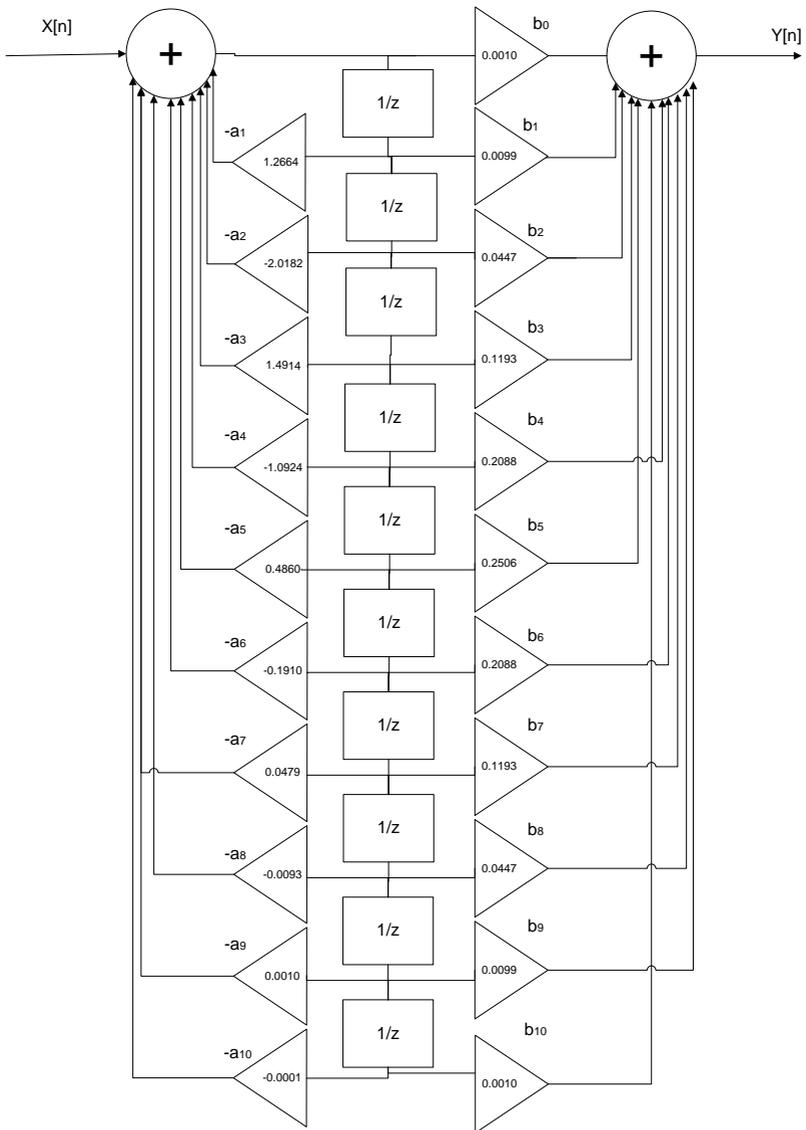


Рис. П2.6. Структурная схема фильтра в прямой канонической форме

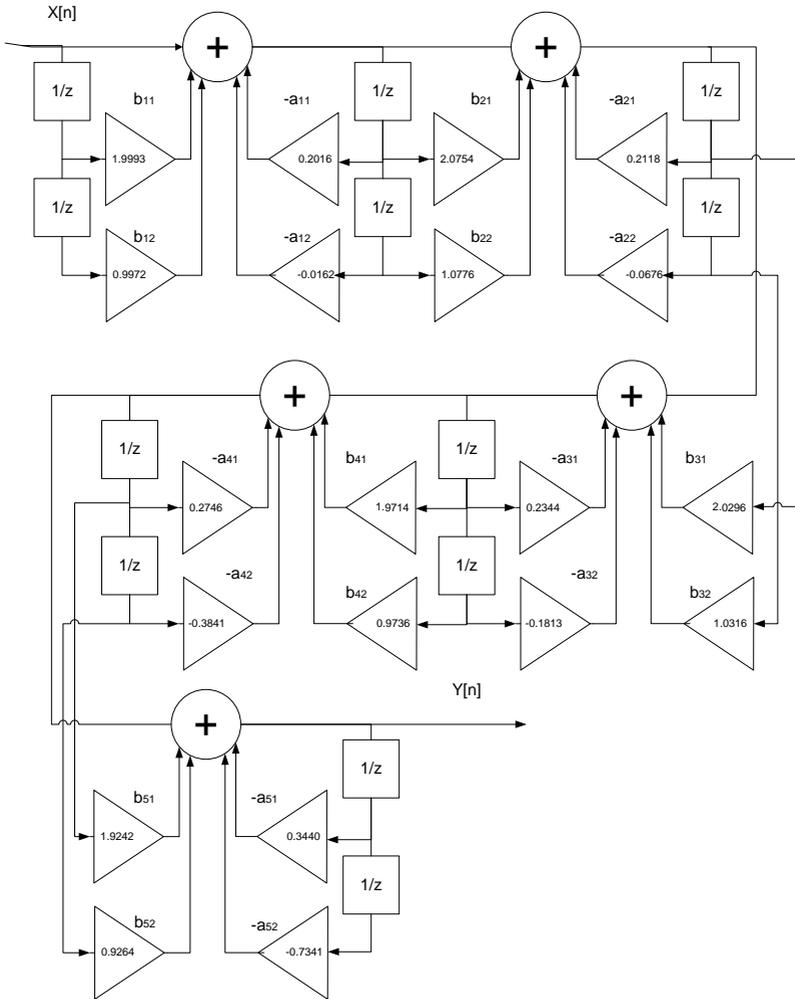


Рис. П2.7. Структурная схема фильтра в последовательной форме

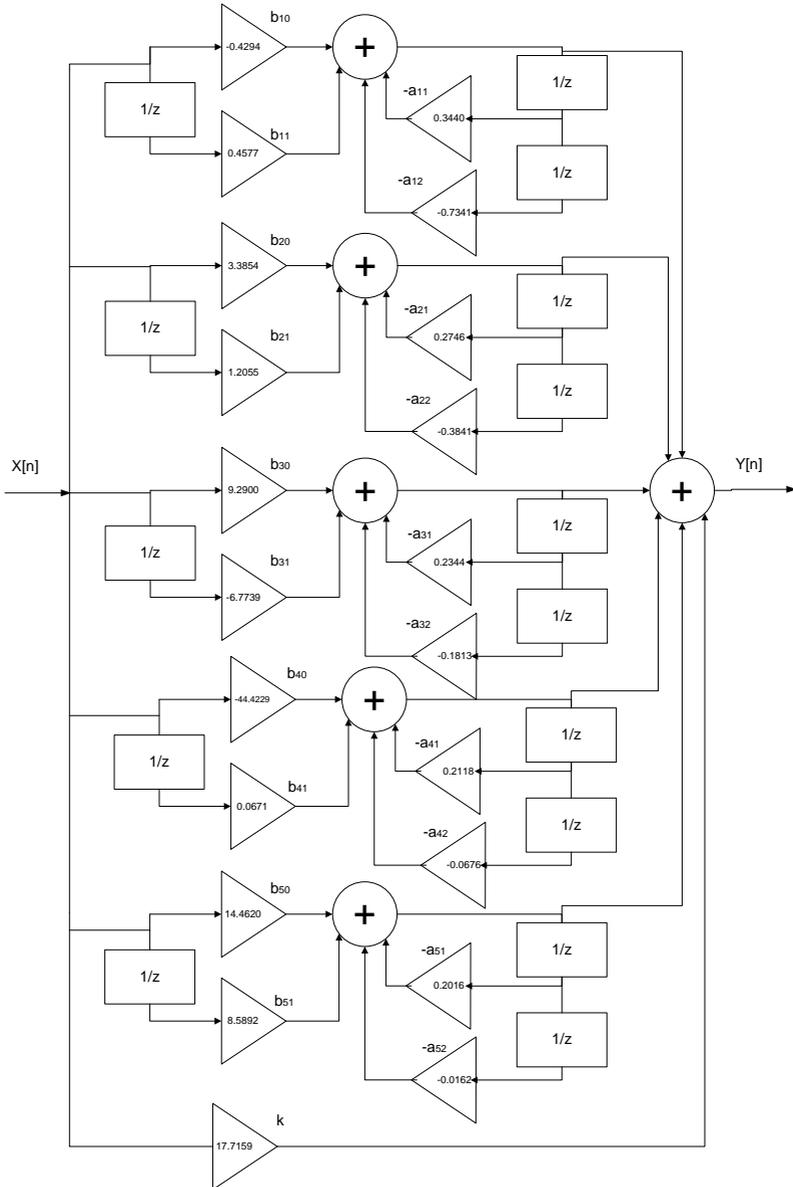


Рис. П2.8. Структурная схема фильтра в параллельной форме

## Часть 2

*Задание.* Программно реализовать ЦФ на языке высокого уровня фильтр в прямой, канонической, последовательной и параллельной формах при представлении коэффициентов разностного уравнения в форматах:

- а) с плавающей точкой,
- б) с фиксированной точкой (16 бит).

*Тексты программ реализации фильтров*

А) формат данных – плавающая точка

```
% Прямая форма
function y = myFilterDirect(x);
```

```
global a;
global b;
global oldx;
global oldy;
```

```
[s t] = size(b);%t=12
```

```
y=x*b(1);
for s=2:t
    y = y + oldx(s-1)*b(s);
    y = y - oldy(s-1)*a(s);
end
```

```
for s=t:-1:2
    oldx(s)=oldx(s-1);
    oldy(s)=oldy(s-1);
end
```

```
oldx(1)=x;
oldy(1)=y;
```

```
% Каноническая форма
function y = myFilterCanon(x);
global a;
global b;
global oldw;
[s t] = size(b);%t=12
w=x;
for s=2:t
```

```

        w = w - oldw(s)*a(s);
    end
    y=0;
    for s=1:t
        y = y + oldw(s)*b(s);
    end

    for s=t:-1:2
        oldw(s)=oldw(s-1);
    end
    oldw(1)=w;

% Последовательная форма
function y = myFilterSerial(x);

global oldx;
global oldy;
global sos;
global g;

y = g*x;

[t s] = size(sos);
for s = 1:t
    temp = y*sos(s,1)+oldx(2*s-1)*sos(s,2) +
oldx(2*s)*sos(s,3)-sos(s,5)*oldy(2*s-1) -
sos(s,6)*oldy(2*s);
    oldy(2*s) = oldy(2*s-1);
    oldy(2*s-1) = temp;
    oldx(2*s) = oldx(2*s-1);
    oldx(2*s-1) = y;
    y = temp;
end

% Параллельная форма
function y = myFilterParallel(x);

global Ai;
global Bi;
global oldx;
global oldy;

```

```

global k;

y = k*x;

[t s] = size(Ai);
for s = 1:t
    temp = x*Bi(s,1)+oldx*Bi(s,2)-Ai(s,2)*oldy(2*s-
1) - Ai(s,3)*oldy(2*s);
    oldy(2*s) = oldy(2*s-1);
    oldy(2*s-1) = temp;
    y = y+temp;
end
oldx = x;

% Получим реакцию фильтра на единичный импульс,
% единичный ступенчатый и синусоидальный сигналы

% Единичный импульс
max = 100;
nn=0:max-1;
x = eye(1,100);

%Прямая форма
[s t] = size(b);
oldx = zeros(1,t);
oldy = zeros(1,t);
y = zeros(1,max);
for temp=1:max
    y(temp) = myFilterDirect(x(temp));
end
figure(10);
plot(nn,x,'r',nn,y,'g');

% Синусоидальный сигнал
f = 300;
max = 100;
nn=0:max-1;
x=sin(2*pi*(f/Fd)*nn);

%Прямая форма
[s t] = size(b);
oldx = zeros(1,t);

```

```

oldy = zeros(1,t);
y = zeros(1,max);
for temp=1:max
    y(temp) = myFilterDirect(x(temp));
end
figure(10);
plot(nn,x,'r',nn,y,'g');

% Единичный ступенчатый сигнал
x = ones(1,100);

%Прямая форма
[s t] = size(b);
oldx = zeros(1,t);
oldy = zeros(1,t);
y = zeros(1,max);
for temp=1:max
    y(temp) = myFilterDirect(x(temp));
end
figure(10);
plot(nn,x,'r',nn,y,'g');

```

Вызов функций других форм реализации фильтра осуществляется аналогично.

*Полученные результаты.* Графики выходных сигналов фильтра совпадают с полученными в части 1. Программная реализация фильтра в формате с плавающей точкой выполнена верно.

Б) формат данных – фиксированная точка (16 бит)

Не удаётся подобрать такое значение коэффициента  $m$ , равного степени 2, чтобы результат умножения значений коэффициентов фильтра  $a$  и  $b$  на него можно было поместить в 16 битный формат. Поэтому не удастся реализовать фильтр в прямой и канонической форме.

Для параллельной и последовательной форм возьмём  $m = 512$  и  $1024$ .

```

% Последовательная форма
function y = myFilterSerialInt(x);

global oldx;
global oldy;
global sos;
global g;
global m;

y = round(g*x/m);
[t s] = size(sos);
for s = 1:t
    temp = round((y*sos(s,1)+oldx(2*s-
1)*sos(s,2)+oldx(2*s)*sos(s,3)-sos(s,5)*oldy(2*s-1)
- sos(s,6)*oldy(2*s))/m);
    oldy(2*s) = oldy(2*s-1);
    oldy(2*s-1) = temp;
    oldx(2*s) = oldx(2*s-1);
    oldx(2*s-1) = y;
    y = temp;
end

% Параллельная форма
function y = myFilterParallelInt(x);
global Ai;
global Bi;
global oldx;
global oldy;
global k;
global m;

y = round(k*x/m);
[t s] = size(Ai);
for s = 1:t
    temp = round((x*Bi(s,1)+oldx*Bi(s,2)-
Ai(s,2)*oldy(2*s-1) - Ai(s,3)*oldy(2*s))/m);
    oldy(2*s) = oldy(2*s-1);
    oldy(2*s-1) = temp;
    y = y+temp;
end
oldx = x;

```

```

% Основная программа
% представим коэффициенты в целом виде, умножив их
% на значение m
m = 512
sos = round(m*sos)
g = round(m*g)
Ai = round(m*Ai)
Bi = round(m*Bi)
k = round(k*m)

% Единичный импульс
max = 100;
nn=0:max-1;
x = eye(1,100);
x_int=m*x;

% параллельная форма
oldx = 0;
[s t] = size(Ai);
oldy = zeros(1,2*s);
y = zeros(1,max);
for temp=1:max
    y(temp) = myFilterParallelInt(x_int(temp));
end
figure(13);
plot(nn,x,'r',nn,y,'g');

% Синусоидальный сигнал
f = 300;
max = 100;
nn=0:max-1;
x=sin(2*pi*(f/Fd)*nn);
x_int=round(m*x);

% параллельная форма
oldx = 0;
[s t] = size(Ai);
oldy = zeros(1,2*s);
y = zeros(1,max);
for temp=1:max
    y(temp) = myFilterParallelInt(x_int(temp));
end

```

```

figure(13);
plot(nn,x,'r',nn,y,'g');
% Единичный ступенчатый сигнал
x = ones(1,100);
x_int = m*x;
% параллельная форма
oldx = 0;
[s t] = size(Ai);
oldy = zeros(1,2*s);
y = zeros(1,max);
for temp=1:max
    y(temp) = myFilterParallelInt(x_int(temp));
end
figure(13);
plot(nn,x,'r',nn,y,'g');

```

Вызов функций последовательной формы реализации фильтра осуществляется аналогично.

*Полученные результаты.* Графики выходных сигналов фильтра идентичны с полученными в части 1. Программная реализация фильтра в параллельной и последовательной формах в формате с фиксированной точкой выполнена верно.

*Выводы по работе.* По заданным требованиям к частотной характеристике в системе MATLAB получены коэффициенты БИХ-фильтра Баттерворта нижних частот. Выполнена программная реализация фильтра на языке высокого уровня в формате с плавающей и фиксированной точкой. В 16-битном целом формате удалось реализовать, подобрав соответствующие масштабированные множители, параллельную и последовательную формы фильтра.

### Приложение 3. Тексты программ быстрого преобразования Фурье

#### Программа БПФ с прореживанием по времени на Паскале

```
CONST PI      = 3.141592653589793;
TYPE  COMPLEX = RECORD
        RE : REAL;
        IM : REAL
    END;

{ПРОЦЕДУРА БПФ}
{A - КОМПЛЕКСНЫЙ ВЕКТОР ДЛИНОЙ 2**M}
{ЕСЛИ INV > 0, ТО ПРЯМОЕ БПФ, ИНАЧЕ - ОБРАТНОЕ}
PROCEDURE FFT(VAR A : VECTOR; M, INV: INTEGER);
    VAR T: COMPLEX;
        I, J, IP, K, L, N, NM1, LE, NV2, LE1: INTE-
GER;
        IND, DIND, INDM: INTEGER;
    BEGIN
        N := STP(2, M);
        IF INV = 0 THEN {ОБРАТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ}
            FOR I := 1 TO N DO CONJG(A[I],A[I]);
        {ОПЕРАЦИЯ ДВОИЧНОГО РЕВЕРСИРОВАНИЯ}
        NV2:= ROUND(N/2);
        NM1:= N-1;
        J := 1;
        FOR I := 1 TO NM1 DO
            BEGIN
                IF I < J THEN
                    BEGIN
                        T := A[J];
                        A[J]:= A[I];
                        A[I]:= T
                    END;
                K := NV2;
                WHILE K < J DO
                    BEGIN
                        J:= J-K;
                        K:= ROUND(K/2)
                    END;
```

```

        J:= J+K
    END;
{КОНЕЦ ОПЕРАЦИИ ДВОИЧНОГО РЕВЕРСИРОВАНИЯ}

DIND := ROUND(N/2);
LE:= 1;
FOR L:= 1 TO M DO
    BEGIN
        LE1:= LE;
        LE := LE*2;
        IND:= 1;
        FOR J:= 1 TO LE1 DO
            BEGIN
                I := J;
                INDM:= ROUND((IND-1)*64/N+1);
                WHILE I <= N DO
                    BEGIN
                        IP:= I+LE1;
                        {'БАБОЧКА'}
                        {T := A[IP]*KOE[INDM]}
                        MUL_COMPLEX(A[IP], KOEF[INDM], T);
                        {A[IP] := A[I]-T}
                        SUB_COMPLEX(A[I], T, A[IP]);
                        {A[I] := A[I]+T}
                        ADD_COMPLEX(A[I], T, A[I]);
                        {КОНЕЦ 'БАБОЧКИ'}
                        I:= I+LE
                    END;
                    IND:= IND+DIND
                END;
                DIND:= ROUND(DIND/2)
            END;
        IF INV = 0 THEN
            FOR I:= 1 TO N DO
                BEGIN
                    CONJG(A[I], A[I]);
                    DIV_COMPLEX(A[I], N, A[I])
                END
            END;
        END;

```

```

{ФУНКЦИЯ ВОЗВЕДЕНИЯ ЧИСЛА DIGIT В СТЕПЕНЬ EX}
FUNCTION STP(DIGIT, EX: INTEGER): INTEGER;
VAR I, PR1: INTEGER;
BEGIN
  PR1:= DIGIT;
  FOR I:= 1 TO EX-1 DO PR1 := PR1*DIGIT;
  IF EX=0 THEN STP:= 1
    ELSE STP:= PR1
END;

{ПРОЦЕДУРА ПОЛУЧЕНИЯ КОМПЛЕКСНО-СОПРЯЖЕННОГО
ЧИСЛА}
{SOURCE - ИСХОДНОЕ КОМПЛЕКСНОЕ ЧИСЛО}
{DEST - ПОЛУЧЕННОЕ КОМПЛЕКСНО-СОПРЯЖЕННОЕ ЧИС-
ЛО}
PROCEDURE CONJG(VAR SOURCE, DEST: COMPLEX);
BEGIN
  DEST.RE:= +SOURCE.RE;
  DEST.IM:= -SOURCE.IM
END;

{ПРОЦЕДУРА СЛОЖЕНИЯ ДВУХ КОМПЛЕКСНЫХ ЧИСЕЛ}
{ A1, A2 -СЛАГАЕМЫЕ, СУММА ВОЗВРАЩАЕТСЯ В REZ}
PROCEDURE ADD_COMPLEX(VAR A1, A2, REZ: COMPLEX);
VAR BUF COMPLEX;
BEGIN
  BUF.RE:= A1.RE+A2.RE;
  BUF.IM:= A1.IM+A2.IM;
  REZ := BUF
END;

{ПРОЦЕДУРА ВЫЧИТАНИЯ ДВУХ КОМПЛЕКСНЫХ ЧИСЕЛ}
{S1 - УМЕНЬШАЕМОЕ, S2 - ВЫЧИТАЕМОЕ}
{ РАЗНОСТЬ ВОЗВРАЩАЕТСЯ В REZ}
PROCEDURE SUB_COMPLEX(VAR S1, S2, REZ: COMPLEX);
VAR BUF: COMPLEX;
BEGIN
  BUF.RE:= S1.RE-S2.RE;
  BUF.IM:= S1.IM-S2.IM;
  REZ := BUF
END;

```

```

{ПРОЦЕДУРА УМНОЖЕНИЯ ДВУХ КОМПЛЕКСНЫХ ЧИСЕЛ}
{M1, M2 - МНОЖИМЫЕ, РЕЗУЛЬТАТ ВОЗВРАЩАЕТСЯ В REZ}
PROCEDURE MUL_COMPLEX(VAR M1, M2, REZ : COMPLEX);
  VAR BUF: COMPLEX;
  BEGIN
    BUF.RE:= M1.RE*M2.RE-M1.IM*M2.IM;
    BUF.IM:= M1.RE*M2.IM+M2.RE*M1.IM;
    REZ    := BUF
  END;

{ПРОЦЕДУРА ДЕЛЕНИЯ КОМПЛЕКСНОГО ЧИСЛА НА ДЕЙСТВИ-
ТЕЛЬНОЕ}
{D1 - ДЕЛИМОЕ, D2 - ДЕЛИТЕЛЬ, ЧАСТНОЕ ВОЗВРАЩАЕТ-
СЯ В REZ}
PROCEDURE DIV_COMPLEX(VAR D1 : COMPLEX;
                      N      : INTEGER;
                      VAR REZ: COMPLEX);

  VAR BUF: COMPLEX;
  BEGIN
    BUF.RE:= D1.RE/N;
    BUF.IM:= D1.IM/N;
    REZ    := BUF
  END;

```

### **Программа БПФ с прореживанием по частоте на Си (среда Visual Studio)**

```

#include<complex>
#include<iostream>

using namespace std;
double PI = 4 * atan(1.0);
complex<double> WW[1024];

int ver_N(int n)
{
    int i, k = 1;
    while (k<n)
        k *= 2;
    return(k);
}

```

```

/*Процедура формирования комплексных коэффициентов
W для БПФ;
Wf - выходной массив комплексных коэффициентов;
T - период (интервал отсчетов в секундах;
N - порядок БПФ;
mode - режим вычисления БПФ: если mode=0, то вычисляются
коэффициенты для прямого БПФ; иначе вычисляются
коэффициенты для обратного БПФ.*/
void Wform(complex<double> *Wf, int N, int mode)
{
    int i;
    N = ver_N(N);
    double PI = 4 * atan(1.0), Tx = ((2 * PI) / (N));
    // период N-точечной входной последовательности
    // c1=exp(-j*T), c2=1;
    complex<double> c1 = complex<double>(cos(Tx),
-sin(Tx)), c2 = complex<double>(1.0, 0.0);
    if (mode != 0) c1 = conj(c1);
    for (i = 0; i<N / 2; i++) // W коэффициентов
    // в 2 раза меньше размерности БПФ
    {
        Wf[i] = c2;
        c2 = c2*c1;
        // WW[i]=Wf[i];

    }
}
} //end of Wform

//Процедура вычисления бабочки размером 2x2
// x \ / Y      Y=(x+y)
//   \ /      k*1
//   / \      X=Wn * (x-y)
// y / W \ X
//
// x1 - адрес x;
// x2 - адрес y;
// n - порядок БПФ,
// l - размер основной "бабочки", в которой
// вычисляется простейшая;
// k - номер простейшей "бабочки" в основной;
// Wf - массив комплексных коэффициентов.

```

```

void babochka (complex<double> *x1, complex<double>
*x2, int n, int l, int k, complex<double> *Wf)
{
    complex<double> y, z;
    y = (*x1) + (*x2);
    (*x2) = (*x1) - (*x2);
    if (k>0)
    {
        z = Wf[(n / l)*k];
        (*x2) = (*x2)*z;
    }
    (*x1) = y;
} //end of babochka

```

```

//Процедура вычисления бабочки размером l;
// vec - адрес входного массива;
// n - порядок вычисляемого БПФ;
// l - размер "бабочки";
// Wf - массив комплексных коэффициентов.
void FFT_eval (complex<double> *vec, int n, int l,
complex<double> *Wf)
{
    complex<double> *m1, *m2;
    int i;
    if (l>1)
    {
        // массив vec разбивается на 2 массива
        m1 = &vec[0]; //адрес первой половины
//массива vec
        m2 = &vec[l / 2]; //адрес второй полови-
ны массива
        // с каждым элементом 1-го и 2-го
// массива вычисляется бабочка размером 2
        for (i = 0; i<l / 2; i++)
            babochka (&m1[i], &m2[i], n, l, i,
Wf);
    }
} //end of FFT_eval

```

```

//Процедура двоичного реверсирования;
// vec - массив значений;
// n - размерность массива.
void Binary_Reverse (complex<double> *vec, int n)
{
    int i, count = -1, ni;
    unsigned short k, kr, ki;
    complex<double> z;

    ni = n;
    // устанавливается, какое количество разрядов
// занимают индексы массива
    while (ni>0)
    {
        ni >>= 1;
        count++;
    }
    // count - число разрядов
    for (k = 0; k<n; k++)
    {
        ki = k;
        kr = 0;
        for (i = 0; i<count; i++)
        {
            __asm{ mov ax, ki
                    mov bx, kr
                    shr ax, 1
                    rcl bx, 1
                    mov ki, ax
                    mov kr, bx
                }
        }
        if (kr>k)
        {
            z = vec[kr];
            vec[kr] = vec[k];
            vec[k] = z;
        }
    }
} //end of Binary_Reverse

```

```

void FFT(complex<double> *vec, complex<double> *W,
int n, int mode)
{
    int i = 1, k;
    double S = 1;
    n = ver N(n);
    while (i<n)
    {
        for (k = 0; k<i; k++)
            FFT_eval(&vec[(n / i)*k], n, n / i, W);
        i *= 2;
    }
    if (mode != 0) S = 1. / (n);
    for (k = 0; k<n; k++)
        vec[k] = vec[k] * S;
    Binary_Reverse(vec, n);
}
int main()
{
    setlocale(0, "rus");
    complex<double> vec[1024];
    int M;
    int mode = 0;
    cout << "Алгоритм прореживания по частоте" << endl;
    cout << "Введите X (длина=2**X, от нуля до
10)" << endl;
    cin >> M;
    int n = pow(2, M);
    Wform(WW, n, mode);
    //vec[0] = complex<double>(1.0);
    for (int i = 0; i<n; i++)
    {
        double zn1 = sin(2*PI*0.25*(i));
        double zn2 = sin(2*PI/3*(i));
        vec[i] = complex<double>(zn1 + zn2, 0);
        // vec[i]=complex<double>(1,0);
    }
    FFT(vec, WW, n, mode);
    cout << "Результат" << endl;
    for (int j = 0; j<n; j++)
        cout<< vec[j];
    system("pause");
    return 0;
}

```

## Приложение 4. Пример выполнения лабораторной работы 4

### Цифровой спектральный анализ. Оценка спектра мощности

*Цель работы.* Изучение алгоритмов оценки спектра мощности цифрового сигнала.

*Задание к лабораторной работе.* Непрерывная реализация длиной  $T$  дискретизируется с частотой  $f_d$  и полученный временной ряд используется для построения оценки спектральной плотности с разрешающей способностью  $df$  и нормированной случайной ошибкой  $E$ . Построить алгоритм, составить и отладить программу оценки, определить необходимое количество операций вещественного сложения и вещественного умножения. Индивидуальные параметры взять из таблицы.

$T$ , с	$f_d$ , Гц	$df$ , Гц	$E$	Вид периодограммы	Вид окна
8	1024	16	?	Уэлча, перекрытие 30%	Хэмминга

*Схема реализации, используемые формулы, особенности метода и применяемых окон.* Общее количество отсчетов, доступное для построения оценки спектра мощности

$$N = T \cdot f_d = 8 \cdot 1024 = 8192;$$

необходимое число отсчетов на одном отрезке, требуемое для достижения заданной разрешающей способности, округленное до ближайшей степени двойки

$$M = f_d / df \cdot k_0 = 1024 / 16 \cdot 1.81 = 128;$$

сдвиг между отрезками при перекрытии 30%

$$\text{сдвиг} = M - 0.3 M = 90;$$

количество отрезков

$$V = \frac{N - M}{\text{сдвиг}} = 89;$$

нормированная случайная ошибка

$$E = \sqrt{\frac{1}{V}} = 0.106.$$

Используемое окно Хэмминга описывается следующей функцией:  $w(k+1) = 0.54 - 0.46 \cos\left(2\pi \frac{k}{n-1}\right)$ , где  $k = 0, 1, \dots, n-1$ .

Достоинством окна является простота его математического описания и, как следствие, простота реализации.

*Текст программы*

```
f1 = 300;
f2 = 400;
k0=1.81;    %коэффициент, зависящий от вида окна
df=16;
fd=1024;
T=8;
N=fd*T      %количество отсчётов
M=2^ceil(log2(fd/df*k0)) %число обрабатываемых
на интервале отсчётов
Sdvig=floor(M-0.3*M)
V=(N-M)/Sdvig +1    %количество отрезков
E=sqrt(1/V)*100 %нормированная случайная ошибка
disp(E)
Td=1/fd;
t=(0:N-1)*Td;
x=sin(2*pi*f1*t)+sin(2*pi*f2*t)+randn(1,N);
n =0:N-1;
plot(n,x)
% Вычисление периодограммы Уэлча
Pw=zeros(1,M);
xp=zeros(1,M);
w = hamming(M);
for p=1:V,
    for n=1:M
        xp(n) = w(n)*x((p-1)*Sdvig+n);
    end
    Pw=Pw+abs(fft(xp)).^2; %fft - БПФ
end
Pw=Pw/(V*M);
m=0:M-1;
f=(m/M)*fd;
plot(f, Pw)
```

*Результаты работы программы* приведены на рис. П4.1 – исходный сигнал, представляющий собой сумму двух синусоид частотой 300 и 400 Гц и белого шума, рис. П4.2 – полученный спектр мощности сигнала, в котором четко выделяются две частотные составляющие.

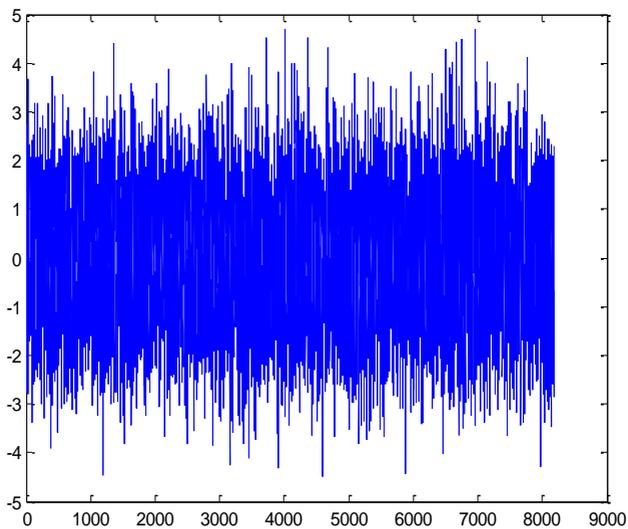


Рис. П4.1. Анализируемый сигнал

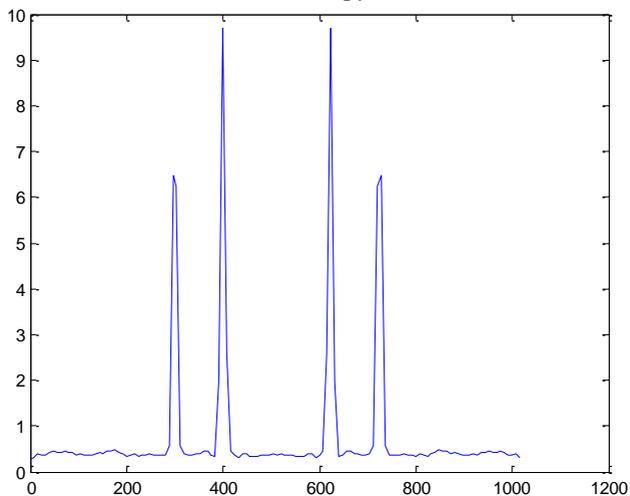


Рис. П4.2. Оценка спектра мощности

*Выводы по работе.* Оценка спектра мощности (периодограмма Уэлча) построена. Однако требуемая разрешающая способность спектрального анализа не достигнута.

## Оглавление

<b>Предисловие</b> .....	3
<b>Лабораторная работа 1. Базовые операции ЦОС</b> .....	3
<i>Теоретические сведения</i> .....	3
Разностные уравнения .....	3
Z-преобразование .....	9
Операция свертки.....	16
Частотная характеристика.....	21
<i>Задание к лабораторной работе</i> .....	24
<i>Порядок выполнения работы</i> .....	26
<i>Содержание отчета</i> .....	26
<i>Вопросы для самоконтроля</i> .....	27
<b>Лабораторная работа 2. Проектирование, реализация и исследование характеристик цифрового фильтра</b> .....	28
<i>Теоретические сведения</i> .....	28
Этапы проектирования цифрового фильтра.....	28
Проектирование БИХ-фильтра на основе аналогового прототипа ..	30
Проектирование БИХ-фильтра при помощи моделирования .....	37
Проектирование КИХ-фильтра при помощи оконных функций .....	39
Проектирование оптимальных КИХ-фильтров.....	45
Структурные схемы цифровых фильтров.....	47
<i>Задание к лабораторной работе</i> .....	52
<i>Содержание отчета</i> .....	54
<i>Вопросы для самоконтроля</i> .....	55
<b>Лабораторная работа 3. Быстрое преобразование Фурье</b> .....	57
<i>Теоретические сведения</i> .....	57
Дискретное преобразование Фурье.....	57
...Алгоритм БПФ с прореживанием по времени.....	62
Матричная трактовка алгоритма БПФ.....	65
...Алгоритм БПФ с прореживанием по частоте.....	66
...Алгоритм БПФ Кули-Тьюки с произвольным основанием.....	66
<i>Задание к лабораторной работе</i> .....	69
<i>Порядок выполнения работы</i> .....	78
<i>Содержание отчета</i> .....	79
<i>Вопросы для самоконтроля</i> .....	79
<b>Лабораторная работа 4. Цифровой спектральный анализ</b> .....	81
<i>Теоретические сведения</i> .....	81
Задачи спектрального анализа.....	81
Дискретные случайные сигналы.....	82
Оценка спектра мощности.....	89
Вычисление корреляционных оценок.....	90

<i>Задание к лабораторной работе</i> .....	92
<i>Порядок выполнения работы</i> .....	93
<i>Содержание отчета</i> .....	93
<i>Вопросы для самоконтроля</i> .....	93
<b>Список рекомендуемой литературы</b> .....	94
<b>Приложения</b>	
<i>Приложение 1. Пример выполнения лабораторной работы 1.</i> Базовые операции цифровой обработки сигналов.....	96
<i>Приложение 2. Пример выполнения лабораторной работы 2.</i> Проектирование, реализация и исследование характеристик БИХ-фильтра.....	99
<i>Приложение 3. Тексты программ быстрого преобразования Фурье</i> .....	121
<i>Приложение 4. Пример выполнения лабораторной работы 4.</i> Цифровой спектральный анализ. Оценка спектра мощности.....	128

*Учебно-практическое издание*

**АНДРЕЕВА** Антонина Аркадьевна

## **ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ**

Практикум

Редактор М.В. Перцева

Компьютерная верстка А.А. Андреевой

Согласно Закону № 436-ФЗ от 29 декабря 2010 года  
данная продукция не подлежит маркировке

Подписано в печать хх.хх.2015. Формат 60×84/16.  
Бумага газетная. Печать офсетная.  
Усл. печ. л. . Уч.-изд. л. ....Тираж 150 экз. Заказ № .

Издательство Чувашского университета  
Типография университета  
428015 Чебоксары, Московский просп., 15