

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Матвеев Ю.Н., Симончик К.К., Тропченко А.Ю., Хитров М.В.

## **ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ**



**Санкт-Петербург**

**2013**

**Матвеев Ю.Н., Симончик К.К., Тропченко А.Ю., Хитров М.В. ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ Учебное пособие по дисциплине "Цифровая обработка сигналов". – СПб: СПбНИУ ИТМО, 2013. – 166 с.**

В учебном пособии рассматриваются основные методы теории цифровой обработки сигналов, используемые при предварительной обработке сигналов различной физической природы. Материал пособия разбит на 6 разделов. В каждом разделе, кроме шестого, приведены краткие теоретические сведения. Задания, приведенные в шестом разделе, имеют своей целью выработать у студентов практические навыки применения основных положений теории цифровой обработки сигналов и ее методов. Пособие может быть использовано при подготовке магистров по направлению 230400.68 «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ», а также магистров по направлению 230100.68 «ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА» и аспирантов.

Рекомендовано Советом факультета Информационных технологий и программирования 7 февраля 2013 г., протокол № 7



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики».

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2013

© Ю.Н. Матвеев, К.К. Симончик, А.Ю. Тропченко, М.В. Хитров. 2013

## Содержание

	стр.
Введение .....	5
<b>1. ОСНОВНЫЕ ПОНЯТИЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ .....</b>	<b>6</b>
1.1. Понятие о первичной и вторичной обработке сигналов .....	6
1.2. Технические средства комплекса обработки сигналов .....	7
<b>2. ПОНЯТИЕ СИГНАЛА. ВИДЫ СИГНАЛОВ .....</b>	<b>10</b>
2.1. Виды сигналов .....	10
2.2. Энергия и мощность сигнала .....	11
2.3. Представление периодических сигналов в частотной области .....	12
2.4. Представление в частотной области непериодических сигналов .....	14
2.4.1. Введение в теорию ортогональных преобразований .....	14
2.4.2. Интегральное преобразование Фурье .....	15
2.5. Свойства преобразования Фурье .....	19
2.5.1. Фурье-анализ неинтегрируемых сигналов .....	21
2.6. Интегральное преобразование Хартли .....	22
2.7. Случайные сигналы .....	24
2.7.1. Модели случайных процессов .....	24
2.7.2. Вероятностные характеристики случайного процесса .....	25
<b>3. КОРРЕЛЯЦИОННЫЙ АНАЛИЗ СИГНАЛОВ .....</b>	<b>29</b>
3.1. Корреляционная функция (КФ) .....	29
3.2. Взаимная корреляционная функция .....	30
3.3. Взаимный спектр сигналов .....	31
3.4. Корреляционные функции случайных процессов .....	32
3.4.1. Стационарные и эргодические случайные процессы .....	34
3.5. Спектральные характеристики случайных процессов .....	35
3.5.1. Теорема Винера-Хинчина .....	36
3.6. Комплексная огибающая сигнала .....	37
<b>4. ПЕРЕХОД ОТ НЕПРЕРВНЫХ СИГНАЛОВ К ЦИФРОВЫМ .....</b>	<b>39</b>
4.1. Дискретизация сигналов .....	39
4.1.1. Влияние формы дискретизирующих импульсов .....	41
4.1.2. Теорема Котельникова .....	41
4.1.3. Дискретизация при использовании квадратурных сигналов .....	44
4.1.4. Определение шага временной дискретизации при восстановлении сигнала полиномами 0-го порядка .....	45
4.1.5. Определение шага дискретизации по заданной автокорреляционной функции .....	47
4.2. Квантование непрерывных сигналов по уровню .....	48
<b>5. ОСНОВНЫЕ ТИПЫ ДИСКРЕТНЫХ АЛГОРИТМОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ .....</b>	<b>51</b>
5.1. Линейные и нелинейные преобразования .....	52
5.2. Характеристики линейных систем .....	54
5.3. Циклическая свертка и корреляция .....	55
5.4. Аперiodическая свертка и корреляция .....	55

5.5. Двумерная аperiodическая свертка и корреляция .....	57
5.6. Не рекурсивные и рекурсивные фильтры.....	59
5.7. Метод синхронного или когерентного накопления	62
5.8. Адаптивные фильтры .....	63
5.8.1. Оптимальный фильтр Винера-Хопфа	69
5.8.2. Оптимальный фильтр Калмана	71
6. ДИСКРЕТНЫЕ ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ	78
6.1. Дискретное преобразование Фурье .....	79
6.2.. Дискретное преобразование Хартли .....	81
6.3. Двумерные дискретные преобразования Фурье и Хартли.....	82
6.4. Ортогональные преобразования в диадных базисах .....	85
6.5. Дискретное косинусное преобразование	89
6.6. Оконное преобразование Фурье	93
6.7. Выполнение фильтрации в частотной области .....	97
6.7.1. Алгоритм Герцеля	100
7.ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ	103
7.1. Понятие о Wavelet-преобразованиях. Преобразование Хаара	104
7.2. Вейвлеты	106
7.2.1. Непрерывные вейвлет преобразования	107
7.2.2. Частотный подход к вейвлет преобразованиям	109
7.2.3. Вейвлет-ряды дискретного времени	109
7.2.4. Дискретное вейвлет-преобразование	111
7.2.4.1. Условия полного восстановления сигнала	113
7.2.5. Пакеты вейвлетов (алгоритм одиночного дерева)	114
7.2.6. Целочисленное вейвлет-преобразование	114
7.3. Применение вейвлет-преобразований для сжатия изображения	118
8. БЫСТРЫЕ АЛГОРИТМЫ ОРТОГОНАЛЬНЫХ ПРЕОБРАЗОВАНИЙ	123
8.1. Вычислительная сложность ДПФ и способы её сокращения.....	123
8.2. Запись алгоритма БПФ в векторно-матричной форме .....	124
8.3. Представление алгоритма БПФ в виде рекурсивных соотношений .....	128
8.4. Алгоритм БПФ с прореживанием по времени и по частоте .....	129
8.5. Алгоритм БПФ по основанию $r$ ( $N = r^m$ ) .....	130
8.6. Вычислительная сложность алгоритмов БПФ .....	131
8.7. Выполнение БПФ для случаев $N \neq r^M$ .....	133
8.8. Быстрое преобразование Хартли .....	135
8.9. Быстрое преобразование Адамара.....	137
8.10. Выбор метода вычисления свертки / корреляции.....	139
9. АЛГОРИТМЫ НЕЛИНЕЙНОЙ ОБРАБОТКИ СИГНАЛОВ .....	142
9.1. Ранговая фильтрация .....	143
9.2. Взвешенная ранговая фильтрация .....	150
9.3. Скользящая эквализация гистограмм.....	151
9.4. Преобразование гистограмм распределения .....	153
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ .....	156
ЛИТЕРАТУРА .....	162

## Введение

Бурный прогресс вычислительной техники в последние десятилетия привел к широкому внедрению методов цифровой обработки информации практически во всех областях научных исследований и народно-хозяйственной деятельности. При этом среди различных применений средств вычислительной техники одно из важнейших мест занимают системы цифровой обработки сигналов (ЦОС), нашедшие использование при обработке данных дистанционного зондирования, медико-биологических исследований, решении задач навигации аэрокосмических и морских объектов, связи, радиофизики, цифровой оптики и в ряде других приложений [2,5,8,11].

Цифровая обработка сигналов (ЦОС) – это динамично развивающаяся область ВТ, которая охватывает как технические, так и программные средства. Родственными областями для цифровой обработки сигналов являются теория информации, в частности, теория оптимального приема сигналов и теория распознавания образов. При этом в первом случае основной задачей является выделение сигнала на фоне шумов и помех различной физической природы, а во втором – автоматическое распознавание, т.е. классификация и идентификация сигнала.

В теории информации под сигналом понимается материальный носитель информации. В цифровой же обработке сигналов под сигналом будем понимать его математическое описание, т.е. некоторую вещественную функцию, содержащую информацию о состоянии или поведении физической системы при каком-нибудь событии, которая может быть определена на непрерывном или дискретном пространстве изменения времени или пространственных координат.

В широком смысле под системами ЦОС понимают комплекс алгоритмических, аппаратных и программных средств. Как правило, системы содержат специализированные технические средства предварительной (или первичной) обработки сигналов и специальные технические средства для вторичной обработки сигналов. Средства предварительной обработки предназначены для обработки исходных сигналов, наблюдаемых в общем случае на фоне случайных шумов и помех различной физической природы и представленных в виде дискретных цифровых отсчетов, с целью обнаружения и выделения (селекции) полезного сигнала, его пеленгования и оценки характеристик обнаруженного сигнала. Полученная в результате предварительной обработки полезная информация поступает в систему вторичной обработки для классификации, архивирования, структурного анализа и т.д. [8,9,11].

Основными процедурами предварительной обработки сигналов являются процедуры быстрых дискретных ортогональных преобразований (БДОП), реализуемых в различных функциональных базисах, процедуры линейной алгебры, линейной и нелинейной фильтрации. Указанные процедуры и быстрые алгоритмы их реализации рассматриваются в данном учебном пособии.

# 1. ОСНОВНЫЕ ПОНЯТИЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

## 1.1. Понятие о первичной и вторичной обработке сигналов

В задачах ЦОС выделяют этапы предварительной (первичной) и вторичной обработки сигналов. Это связано с тем, что в общем случае на входе системы ЦОС наблюдается смесь  $x(t)$  полезного сигнала  $s(t)$ , некоторого шума  $n(t)$  и различных помех разной природы  $p(t)$ :

$$x(t) = S(t) + p(t) + n, \quad (1.1)$$

где  $n(t)$  является характеристикой самого технического устройства, а  $p(t)$  – некоторое искажающее воздействие самой физической среды, в которой распространяется сигнал (например, затухание).

Различают следующие задачи цифровой обработки сигналов:

- 1) Обеспечение оптимального приёма сигналов, под которым понимается обеспечение максимально возможного подавления помех различной природы и шумов, т.к. в общем случае на вход приёмника попадает их смесь;
- 2) Определение числовых параметров сигналов – энергии, средней мощности, среднеквадратичного значения и т.д.
- 3) Разложение сигналов на некоторый набор элементарных составляющих для рассмотрения их в дальнейшем по отдельности или совместно, а так же решение обратной задачи синтеза сигнала.
- 4) Количественное измерение степени схожести или подобия сигналов.
- 5) Решение задач распознавания и идентификации сигналов.

При этом выделяют следующие основные этапы цифровой обработки сигналов:

- 1) *Предварительная обработка* – приём, успешное преобразование из аналоговой в цифровую форму представления.
- 2) *Первичная обработка* – оптимальный приём и анализ (см. задачи 1-3).
- 3) *Вторичная обработка* – выделение сигнала заданного вида, классификация, распознавание и т.д. (см. задачи 4-5).

Важнейшей задачей первичной обработки сигнала является подавление  $n(t)$  и  $p(t)$  (шума и помехи). Такая задача оптимального приема может быть решена только на основе использования избыточности представления исходного сигнала, а также имеющихся сведений о свойствах полезного сигнала, помехи и шума для увеличения вероятности правильного приема [11,19].

Вследствие того, что на вход приемного устройства системы поступает сумма полезного сигнала и помехи, вероятность правильного приема будет определяться отношением полезного сигнала к помехе. Для повышения вероятности правильного приема сигнала должна быть произведена предварительная обработка принятого сигнала, обеспечивающая увеличение отношения сигнал/помеха. Таким образом, средства цифровой обработки при приеме должны содержать два основных элемента (рис.1.1) : фильтр  $\Phi$ ,

обеспечивающий улучшение отношения сигнал/помеха, и решающее устройство РУ, выполняющее главные функции приема (обнаружения, различения и восстановления сигналов).

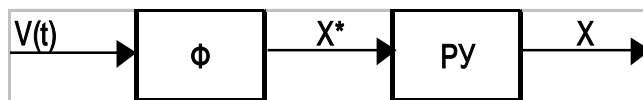


Рис.1.1. Структура оптимального приемного устройства

Известны следующие методы фильтрации, обеспечивающие улучшение соотношения сигнал/помеха:

- метод накопления;
- частотная фильтрация;
- корреляционный метод;
- согласованная фильтрация;
- нелинейная фильтрация.

Все эти методы основаны на использовании различий свойств полезного сигнала и помехи.

Кроме того, при первичной обработке решается задача обнаружения сигнала и определения местоположения его источника. На этом же этапе обработки в ряде случаев формируются также некоторые количественные оценки сигнала (амплитуда, частота, фаза).

Во входной смеси может и не быть полезного сигнала  $x(t)$ , поэтому на выходе системы предварительной обработки не будет никакого сигнала; следовательно, интенсивность потока данных на выходе будет ниже, чем на входе.

Система вторичной обработки сигнала предназначена для идентификации обнаруженного сигнала, его классификации и выдачи информации об обнаруженных сигналах оператору или формирования управляющего воздействия.

Характерной чертой первичной обработки сигнала является постоянство алгоритма обработки при его достаточно высокой вычислительной сложности. Этап вторичной обработки характеризуется большей гибкостью используемых алгоритмов, необходимостью поддержки обмена с другим техническим средством или диалога с оператором. Поэтому системы вторичной обработки чаще всего строятся на основе программируемых вычислительных средств. Системы же первичной обработки могут быть построены как на программируемых вычислительных средствах, так и на основе специальных вычислителей с жесткой логикой [11].

## 1.2. Технические средства комплекса обработки сигналов

Комплекс цифровой обработки сигналов содержит ЭВМ, специализированные устройства ввода и соответствующее программное

обеспечение. В общем случае подобный комплекс должен также обеспечивать ввод, вывод и передачу сигналов различной физической природы. Общие требования к системам ЦОС представлены в таблице 1.1.

При этом особый интерес представляет обработка двумерных сигналов – изображений, получаемых от различных приемных устройств.

Многие задачи обработки изображений могут быть решены на современных персональных ЭВМ, если к скорости обработки не предъявляются высокие требования. В этом случае те или иные процедуры обработки изображений на ПЭВМ реализуются путем создания специального программного обеспечения. Для обеспечения ввода изображения в реальном масштабе времени используются специализированные устройства ввода. К такому типу систем относятся системы IMAGE-3 и Microsight-2. Заметим, что в них обработка изображений производится на ПЭВМ не в реальном масштабе времени. Для обработки сигнальной информации в реальном масштабе времени требуется производительность, превышающая производительность ПЭВМ. В этом случае необходимы специализированные устройства обработки. В настоящее время, согласно литературе, известны два типа систем обработки сигналов [8,9,11,20].

Таблица 1.1.

**Основные требования к системам ЦОС**

<b>Параметр</b>	<b>Значение параметра</b>	<b>Область применения</b>
Динамический диапазон обрабатываемых данных (бит)	Фиксированная точка: 8-16	Сжатие изображений, Радиолокация, Связь
	Плавающая точка: 16-32	Гидроакустика, Обработка речи, Обработка изображений
Емкость памяти буферных устройств (слов)	64-128 не менее 1К более 4К до 16К	Сжатие изображений, Радиолокация, Гидроакустика, Обработка изображений
Производительность (оп/сек)	$10^8$ $10^9$ $10^9-10^{10}$ $10^{10}-10^{11}$	Системы связи Обработка акустических сигналов Обработка изображений Радиолокация и гидроакустика



Первый тип систем ЦОС предусматривает построение конструктивно законченного блока. Как правило, такой блок имеет модульную структуру и строится на базе специализированных СБИС (например, на основе БМК), что позволяет обеспечить аппаратную реализацию подлежащего исполнению алгоритма и оптимизировать структуру аппаратных средств под особенности алгоритма. К этому направлению можно отнести системы Series-151 и MaxVideo. В ряде случаев такие процессоры могут программироваться в целях выполнения тех или иных функций, как, например, WARP-процессор [9].

Отличительной чертой такой архитектуры является наличие отдельных магистралей ввода/вывода данных и возможность автономного функционирования. Блок со спецпроцессором при этом может быть выполнен в стандартном конструктиве типа VME, SAMAC, Multibys [8,9].

Такая система ЦОС допускает не только ввод, но и обработку изображений в реальном масштабе времени, поэтому подобный подход весьма эффективен при построении систем обработки видеоданных.

Второй тип систем ЦОС представляет собой ПЭВМ со специализированным сопроцессором в виде платы, подключаемой к магистрали ПЭВМ и конструктивно встраиваемый в ее корпус. Примером такой архитектуры могут служить наборы модулей фирмы Data Translation на базе сигнальных процессоров типа TMS и платы-акселераторы типа B008 фирмы INMOS на базе транспьютеров T800 [24]. Указанные технические средства ориентированы на использование в качестве периферийных спецпроцессоров для построения систем на базе IBM PC/AT. Спецпроцессор, входящий в эту систему, имеет, как правило, конвейерную структуру и может выполнять процедуры обработки изображений, требующие больших вычислительных затрат, в реальном масштабе времени. Настройка на выполнение тех или иных конкретных алгоритмов обработки видеoinформации производится программированием спецпроцессора, что увеличивает функциональную гибкость подобных систем и расширяет области их возможного применения.

На практике первый тип систем ЦОС наиболее часто используется в составе средств предварительной обработки сигналов, причем соответствующие вычислительные средства строятся по принципу операционного автомата с жесткой логикой. Такой подход связан с автономностью функционирования средств предварительной обработки от управляющей ЭВМ при неизменном алгоритме обработки и высокой интенсивности входного потока данных.

Второй тип систем используется, как правило, для систем, сочетающих средства предварительной (спецпроцессоры) и вторичной (ПЭВМ) обработки, когда требуется достаточно интенсивный обмен с оператором.

## 2. ПОНЯТИЕ СИГНАЛОВ. ВИДЫ СИГНАЛОВ

Сигнал является физическим носителем информации. Информация передаётся сигналом за счёт изменения характеристик или параметров сигнала, например, во времени (в большинстве случаев) или как изменение (например, интенсивности света) от пространственных координат. Следовательно, с математической точки зрения, сигнал описывается как функция от одной или нескольких переменных.

### 2.1. Виды сигналов

Сигналы могут быть подразделены на следующие виды [19]:

1) *Аналоговые сигналы*  $x = x(t)$ , т.е. описываемые как функция (чаще всего действительная от одной или нескольких переменных, например, времени, пространственных координат. При этом значение сигнала может быть определено в любой момент времени как действительное число. Если например, аналоговый сигнал задан как изменение напряжения от времени  $u(t)$ , то размерность сигнала – вольт.

2) *Цифровые сигналы*  $X = [x_0, x_1, x_2, \dots, x_{N-1}]$ , т.е. заданные в виде набора значений или измерений в фиксированные моменты времени. Такие значения принято называть отсчетами сигнала. Каждый отсчет задан с определенной заданной точностью, т.е является числом фиксированной разрядности.

3) *Детерминированные сигналы* - значение сигнала в любой момент времени точно определённое.

4) *Случайные сигналы* -. Значение сигнала в любой момент времени является случайной величиной, которое принимает определённое конкретное значение с некоторой вероятностью.

5) *Сигналы с ограниченной энергией* (или интегрируемым квадратом)

$$\int_{-\infty}^{\infty} S^2(t) dt < \infty$$

б) *Периодические сигналы*:

$$S(t) = S(t + nT); \quad n = 1, 2, \dots$$

Где  $T$  – период сигнала;  $f = \frac{1}{T}$  – частота повторения сигнала ( $\text{с}^{-1}$ , Гц);

$\omega = 2\pi f = \frac{2\pi}{T}$  – круговая частота (рад/с). Отметим что периодические сигналы имеют бесконечную энергию.

7) *Финитные сигналы* или сигналы с конечной длительностью – если описывающая их функция отлична от «0» на конечном интервале. Если такая функция не имеет разрывов II рода (когда ветви сигнала уходят в бесконечность), то энергия таких финитных сигналов конечна.

**Более узкие виды сигналов.**

Как носители информации могут быть выделены:

а) *Потенциальный* сигнал (сигнал постоянного уровня, например, «0» → логический «0», 5В → логическая «1»). Допускает только амплитудную модуляцию.

б) *Гармонический* сигнал:

$$S(t) = A \cos(\omega t + \varphi)$$

$A$  – амплитуда;  $\omega = \frac{2\pi}{T} = 2\pi f$  – круговая частота;  $\varphi$  – начальная задержка или фаза. Допускает амплитудную, фазовую или частотную модуляцию или их одновременную комбинацию.

в) *Дельта-функция*  $\delta(t)$  или единичный импульс

$$\delta(t) = \begin{cases} 0, & t \neq 0 \\ \infty, & t = 0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

Фильтрующее свойство:

$$\int_{-\infty}^{\infty} f(t) \delta(t - t_0) dt = f(t_0)$$

$$\text{comb}(t) = \sum_{k=-\infty}^{\infty} \delta(t - t_k) \text{ – гребенчатая функция}$$

г) *Функция Хевисайда* (функция включения) или единичный скачок

$$\sigma(t) = \begin{cases} 0, & t < 0 \\ 1/2, & t = 0 \\ 1, & t > 0 \end{cases}$$

д) *Прямоугольный импульс*:

$$S(t) = A(\sigma(t) - \sigma(t - t_0))$$

$$\text{rect}\left(\frac{t}{T}\right) = \begin{cases} 1, & |t| \leq T/2 \\ 0, & |t| > T/2 \end{cases}$$

Сигналы могут описываться действительной функцией или, существенно реже, комплексной функцией:

$$S(t) = A \cdot e^{-j\omega_0 t}$$

## 2.2. Энергия и мощность сигнала

*Энергия сигнала*:

$$E = \int_0^T S^2(t) dt \quad (B^2 \cdot C) \quad (2.1)$$

*Мгновенная мощность* (instantaneous power):

$$P(t) = S^2(t)$$

*Средняя мощность* (average power):

$$P_{cp} = \frac{1}{T} \int_0^T S^2(t) dt \quad (B^2) \quad (2.2)$$

Если энергия сигнала бесконечна как, например, у периодических сигналов, то:

$$P_{cp} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} S^2(t) dt$$

Среднеквадратическое (действующее) значение сигнала (root mean square; RMS)

$$\sigma_S = \sqrt{P_{cp}} = \sqrt{\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} S^2(t) dt} \quad (2.3)$$

### 2.3. Представление периодических сигналов в частотной области

Периодические сигналы могут быть описаны в виде суммы (или суперпозиции) гармонических составляющих или гармоник, каждая из которых имеет определённую частоту, амплитуду и начальную фазу. Конкретный набор таких составляющих будет определяться видом сигнала  $S(t)$ . Для того, чтобы такое представление можно было бы осуществить, фрагмент сигнала на периоде  $T$  должен удовлетворять условиям Дирихле, т.е.:

- 1) не должно быть разрывов II рода;
- 2) число разрывов I-го рода (или скачков) должно быть конечным;
- 3) число экстремумов должно быть конечным.

При соблюдении этих требований периодический сигнал  $S(t)$  может быть представлен в виде ряда Фурье:

$$S(t) = \frac{A_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\omega_1 t) + b_k \sin(k\omega_1 t), \quad (2.4)$$

где  $\omega_1 = \frac{2\pi}{T}$  – круговая частота или период повторения сигнала;

$\frac{A_0}{2} = \frac{1}{T} \int_{-T/2}^{T/2} S(t) dt$  – постоянная составляющая сигнала;

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} S(t) \cos(k\omega_1 t) dt;$$

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} S(t) \sin(k\omega_1 t) dt;$$

$k\omega_1$  –  $k$ -я частотная составляющая сигнала или  $k$ -я гармоника.

Заметим, что пределы интегрирования могут быть и другими, например, от 0 до  $T$  – важно, чтобы охватывался бы лишь весь период сигнала  $S(t)$ .

Если  $S(t)$  – чётная функция (это значит, что  $S(t) \equiv S(-t)$ ), то все  $b_k \equiv 0$  и, наоборот, если  $S(t)$  – нечётная функция ( $S(t) = -S(-t)$ ), то все  $a_k \equiv 0$ .

Такое разложение можно записать в тригонометрической форме:

$$S(t) = \frac{A_0}{2} + \sum_{k=1}^{\infty} A_k \cos(k\omega_1 t + \varphi_k), \quad (2.5)$$

где  $A_k = \sqrt{a_k^2 + b_k^2}$  – амплитуда  $k$ -й гармоники;

$\varphi_k = \arctg(b_k/a_k)$  – начальная фаза.

При этом если  $S(t)$  – чётная, то  $\varphi_k \in [0; \pi]$  и если  $S(t)$  – нечётная, то  $\varphi_k \in [\pm \pi/2]$ .

Множество амплитуд гармоники называют амплитудным спектром, а множество фаз – фазовым спектром.

Если  $S(t)$  – действительная функция, то:

$$A_{-k} = A_k$$

$$\varphi_{-k} = -\varphi_k.$$

Пример: Пусть исходный сигнал представляет собой последовательность прямоугольных импульсов с постоянным периодом  $T$  (рис. 2.1).

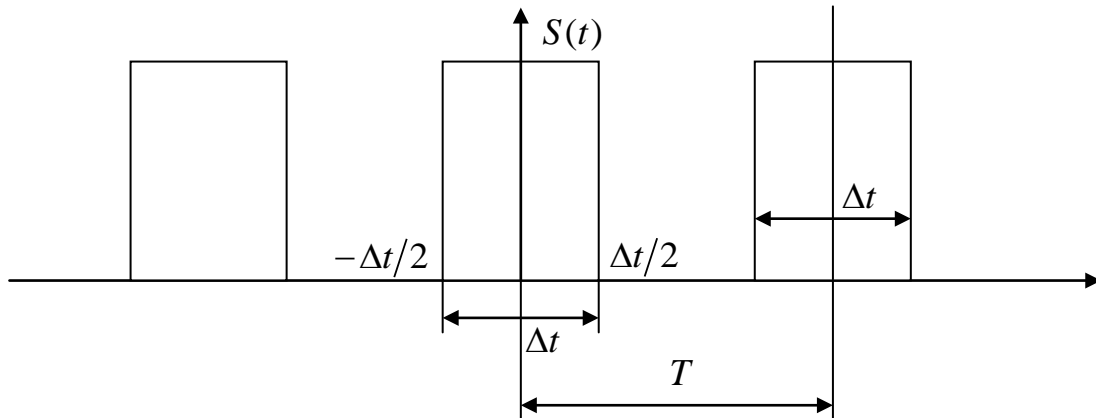


Рис.2.1. Последовательность прямоугольных импульсов

Т.к. такой сигнал чётный, то надо определить

$$a_k = \frac{2}{T} \int_{-\Delta t/2}^{\Delta t/2} A \cos\left(\frac{2\pi k}{T} t\right) dt = \frac{2A}{\pi k} \sin \frac{2\pi \Delta t}{T}$$

$q$  – скважность;  $q = T/\Delta t$ ;

$q^{-1} = \frac{\Delta t}{T}$  – коэффициент заполнения (duty cycle);

$$A_0 = \frac{A\Delta t}{T} = \frac{A}{q}$$

$$S(t) = \frac{A}{q} + \sum_{k=1}^{\infty} \frac{2A}{\pi k} \sin\left(\frac{\pi k}{q}\right) \cos\left(\frac{2\pi k}{T} t\right)$$

Амплитудный спектр подобного сигнала показан на рис.2.2..

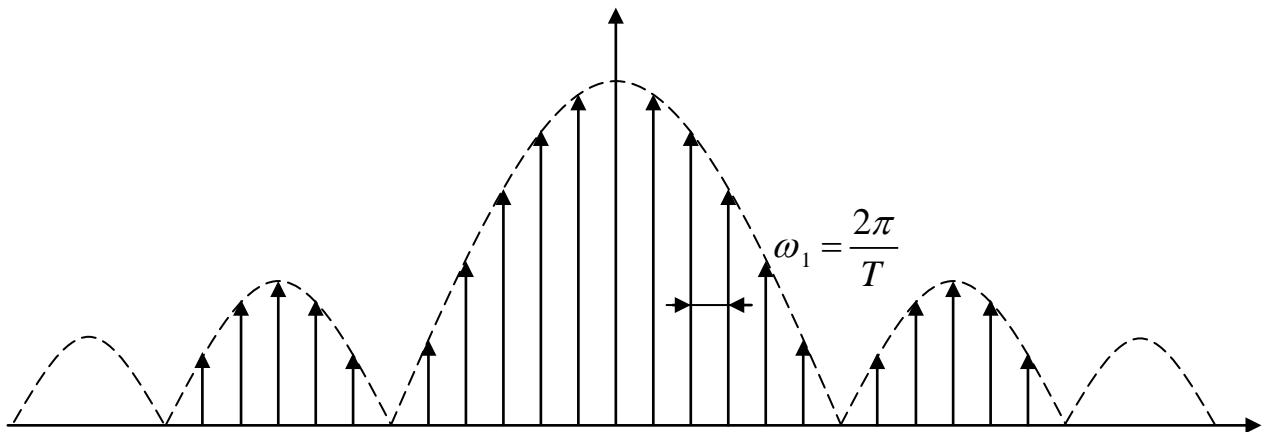


Рис.2.2. Амплитудный спектр последовательности прямоугольных импульсов.

Амплитудный и фазовый спектры периодических сигналов дискретны, т.е. определены на фиксированных частотах  $\omega_k = k\omega_1 = \frac{2\pi}{T}k$

## 2.4. Представление в частотной области непериодических сигналов

Реальный сигнал ограничен во времени и, следовательно, является непериодическим. Однако, условно его можно рассматривать как периодический с периодом  $T \rightarrow \infty$ . Тогда  $\omega_0 = 2\pi/T \rightarrow 0$ , а спектры амплитуд и фаз становятся непрерывными (сплошными), сумма в разложении Фурье превращается в интеграл. Такое интегральное преобразование относится к классу *ортogonalных интегральных преобразований*. Поэтому вначале рассмотрим основные особенности ортогональных преобразований.

### 2.4.1. Введение в теорию ортогональных преобразований

Две вещественные функции  $g(x)$  и  $h(x)$ , заданные на конечном или бесконечном интервале ( $a < x < b$ ), называются ортогональными друг другу на этом интервале, если

$$\int_a^b g(x)h(x)dx = 0$$

При этом функции предполагаются конечными либо бесконечными, но обязательно с абсолютно сходящимся интегралом. Интеграл называется абсолютно сходящимся если

$$\int_a^\infty |f(x)|dx < \infty.$$

Система функций называется ортогональной на некотором интервале, если каждые две функции из этой системы ортогональны друг другу на этом интервале.

Пусть задана система функций

$$g_1(x), g_2(x), \dots, g_n(x)$$

ортогональная на некотором интервале ( $a < x < b$ ).

Может возникнуть задача о разложении произвольной функции  $f(x)$  на этом интервале в ряд по функциям (2.1), т.е. в ряд вида

$$f(x) = a_1 g_1(x) + a_2 g_2(x) + \dots + a_n g_n(x) + \dots = \sum_{n=1}^{\infty} a_n g_n(x),$$

где  $a_n$  – числовые коэффициенты. При этом возникают вопросы: возможно ли разложение для любой функции  $f(x)$  и как найти коэффициенты  $a_n$ .

Будем считать для простоты все рассматриваемые функции, а также интервал конечным. Ответ на первый вопрос зависит от выбора системы, по которой мы будем производить разложение. Если разложение возможно для любой функции  $f(x)$ , то система функций называется полной [6].

Перейдем теперь к нахождению коэффициентов  $a_n$  разложения, причем будем считать, что ни одна из функций (2.1) не равна тождественно нулю. Для этого умножим обе части уравнения (2.2) на  $g_n(x)$  и проинтегрируем результат по интервалу ( $a < x < b$ ).

$$\int_a^b f(x) g_n(x) dx = a_1 \int_a^b g_1(x) g_n(x) dx + a_2 \int_a^b g_2(x) g_n(x) dx + \dots + a_n \int_a^b g_n^2(x) dx + \dots$$

В силу ортогональности системы (2.1), в правой части последнего равенства все интегралы равны нулю, за исключением интеграла от  $g_n^2(x)$ , и мы получаем формулу для коэффициентов

$$a_n = \frac{\int_a^b f(x) g_n(x) dx}{\int_a^b g_n^2(x) dx} \quad (n=1, 2, \dots)$$

### 2.4.2. Интегральное преобразование Фурье

Непериодический сигнал может быть в частотной области описан с помощью прямого интегрального преобразования Фурье, однако он для этого должен удовлетворять следующим требованиям:

- 1) должно выполняться условие Дирихле;
- 2) должен быть абсолютно интегрируемым, т.е.:

$$\int_{-\infty}^{\infty} |S(t)| dt < \infty.$$

**Прямое преобразование Фурье (Direct Fourier Transform) имеет вид:**

$$S(\omega) = \int_{-\infty}^{\infty} S(t) e^{-j\omega t} dt = \int_{-\infty}^{\infty} S(t) e^{-j2\pi ft} dt = \underbrace{\int_{-\infty}^{\infty} S(t) \cos(2\pi ft) dt}_{\text{Re}[S(\omega)]} - j \underbrace{\int_{-\infty}^{\infty} S(t) \sin(2\pi ft) dt}_{\text{Im}[S(\omega)]} \quad (2.6),$$

где  $S(\omega)$  – спектральная функция или спектральная плотность сигнала (в (2.6) и далее означает комплексную функцию). Иногда в задачах обработки сигналов ее называют фурье-образом или фурье-спектром сигнала.

В этом выражении для его преобразования использована формула Эйлера для записи комплексного числа в тригонометрической форме:

$$e^{jx} = \underbrace{\cos x}_{\text{Re } x} + j \underbrace{\sin x}_{\text{Im } x}$$

От спектральной плотности можно перейти к амплитудному спектру

$$A(\omega) = \sqrt{\operatorname{Re}^2[\dot{S}(\omega)] + \operatorname{Im}^2[\dot{S}(\omega)]} \quad (2.7)$$

и фазовому спектру

$$\varphi_\omega = \operatorname{arctg} \left[ \frac{\operatorname{Im}(\dot{S}(\omega))}{\operatorname{Re}(\dot{S}(\omega))} \right] \quad (2.8).$$

Для вещественной функции  $S(t)$  спектральная плотность на частотах  $\omega$  и  $-\omega$  является комплексно-сопряжённой, т.е.  $\dot{S}(-\omega) = \dot{S}^*(\omega)$ , тогда для амплитудного и фазового спектров справедливы соотношения:

$$A(\omega) = A(-\omega)$$

$$\varphi(\omega) = \varphi(-\omega).$$

Если  $S(t)$  – чётная, то спектральная плотность является *вещественной* и *чётной* и, наоборот, для нечётной  $S(t) \rightarrow \dot{S}(\omega)$  – чисто мнимая и нечётная.

**Обратное преобразование Фурье (Inverse Fourier Transform)** обеспечивает переход из частотной области во временную область заданного сигнала:

$$S(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \dot{S}(\omega) e^{j\omega t} d\omega; \quad (2.9)$$

Пример 1. Пусть сигнал является отдельным прямоугольным импульсом (рис.2.3.)

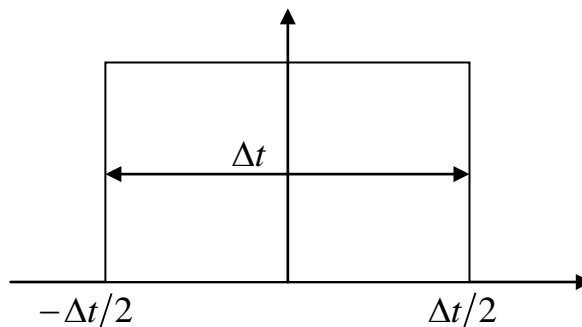


Рис 2.3. Прямоугольный импульс.

Такой сигнал может быть описан как (см. раздел 2.1):

$$S(t) = \begin{cases} A, & |t| \leq \Delta t/2 \\ 0, & |t| > \Delta t/2 \end{cases}$$

В этом случае спектральная плотность сигнала определяется следующим образом (с учетом выражения (2.6))

$$\dot{S}(\omega) = \int_{-\Delta t/2}^{\Delta t/2} A \cdot e^{-j\omega t} dt = \frac{2A}{\omega} \sin\left(\frac{\omega \Delta t}{2}\right) = A \Delta t \frac{\sin\left(\frac{\omega \Delta t}{2}\right)}{\frac{\omega \Delta t}{2}}.$$

Амплитудный и фазовый спектры такого сигнала представлены на рис. 2.4 и рис.2.5 соответственно.



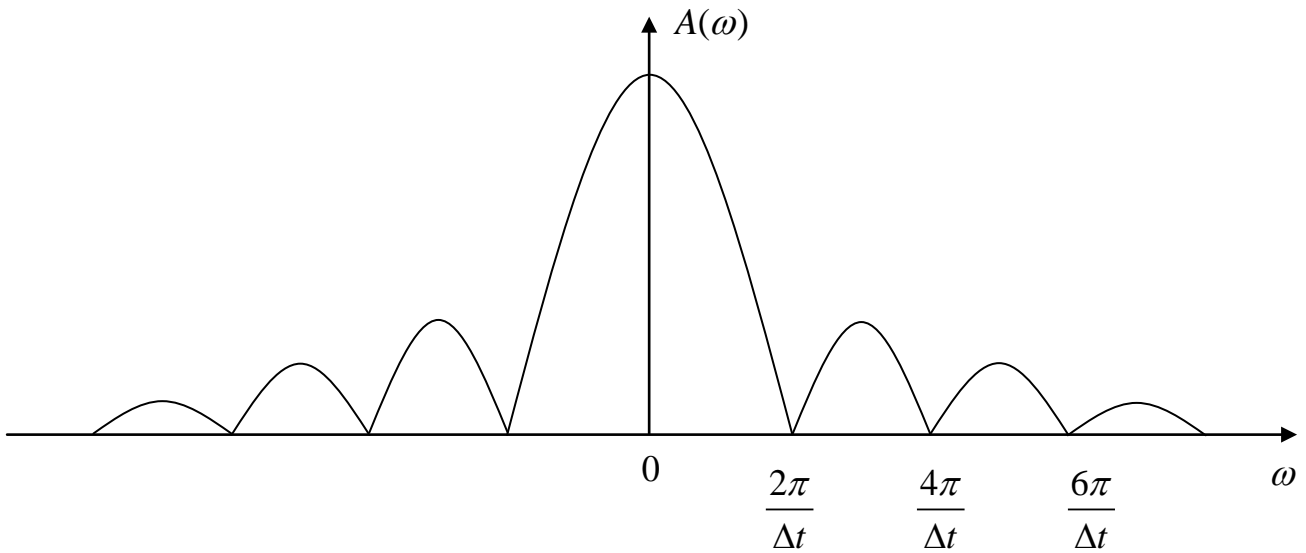


Рис. 2.4. Амплитудный спектр прямоугольного импульса.

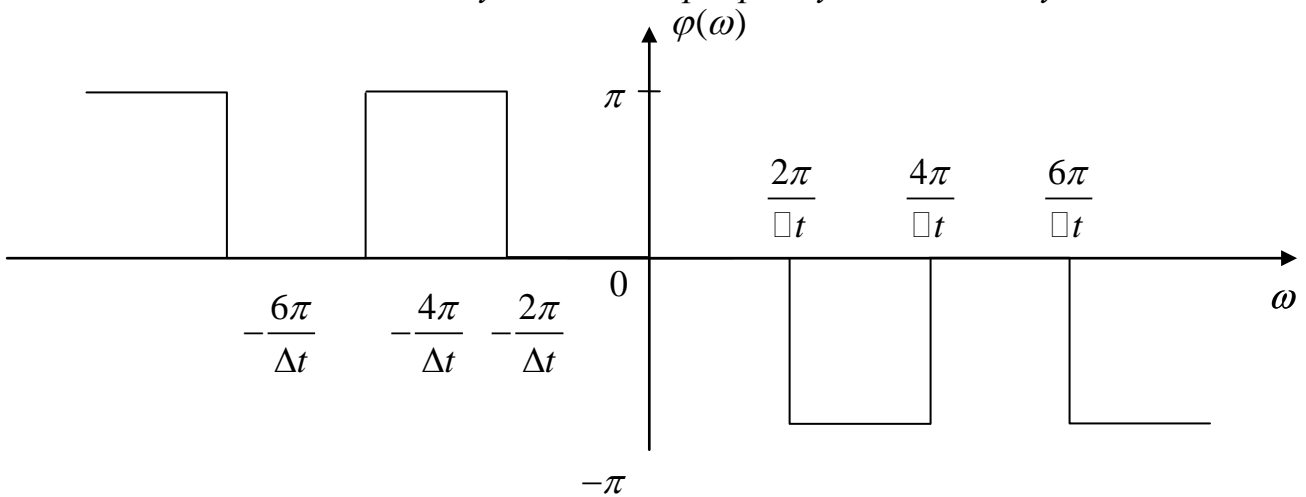


Рис. 2.5. Фазовый спектр прямоугольного импульса

Пример 2. Исходный сигнал является сдвинутым прямоугольным импульсом (рис.2.6).

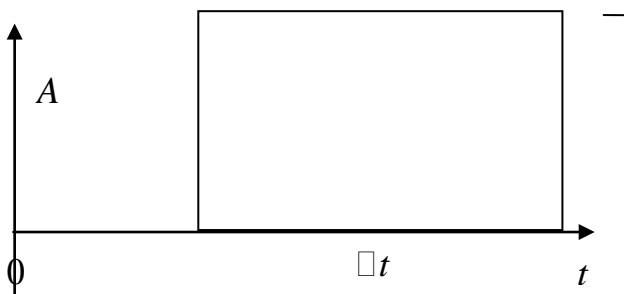


Рис. 2.6. Сдвинутый прямоугольный импульс.

Такой сигнал может быть описан как (см. раздел 2.1):

$$S(t) = \begin{cases} A, & 0 \leq t \leq \Delta t \\ 0, & t < 0; t > \Delta t \end{cases}$$

Тогда получаем выражение для его спектральной плотности:

$$\dot{S}(\omega) = \int_{-\Delta t/2}^{\Delta t/2} A \cdot e^{-j\omega t} dt = \frac{2A}{\omega} \sin\left(\frac{\omega\Delta t}{2}\right) = A\Delta t \frac{\sin\left(\frac{\omega\Delta t}{2}\right)}{\frac{\omega\Delta t}{2}}$$

На рис. 2.7 представлен амплитудный спектр сдвинутого прямоугольного импульса такого сигнала, а на рис. 2.8 – его фазовый спектр.

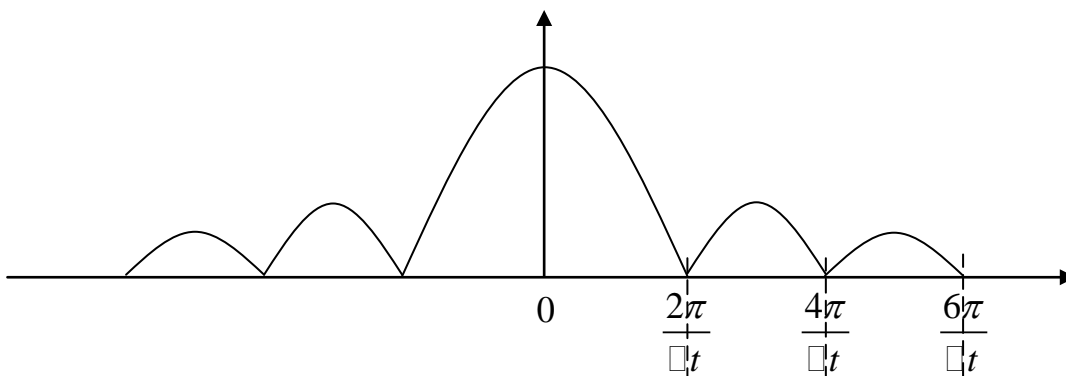


Рис.2.7. Амплитудный спектр сдвинутого прямоугольного импульса

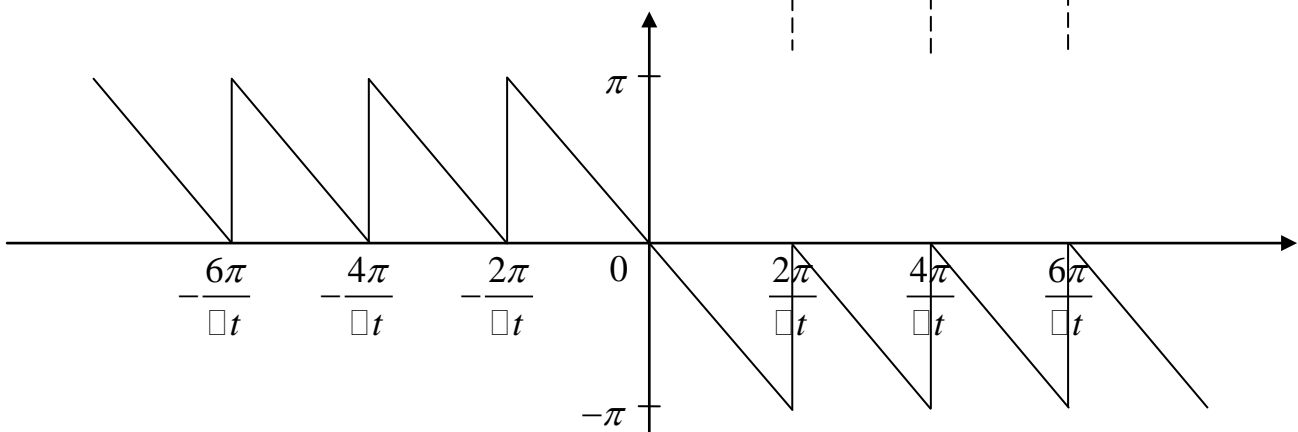


Рис. 2.8. Фазовый спектр сдвинутого прямоугольного импульса

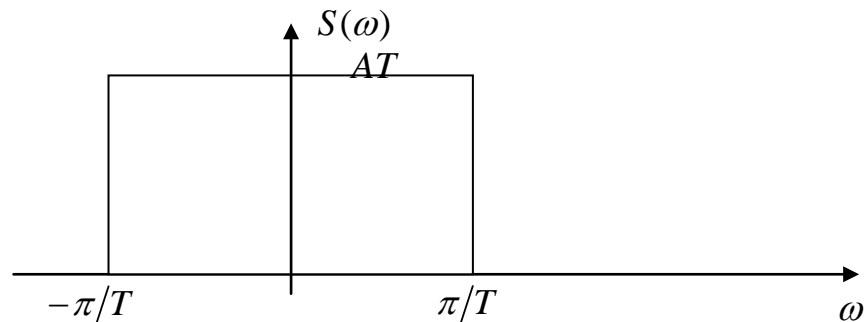
Отметим, что амплитудные спектры на рис.2.4 и рис. 2.7 совпадают, несмотря на сдвиг прямоугольного импульса.

Пример 3. Пусть исходный сигнал имеет вид:

$$S(t) = A \frac{\sin(\pi t/T)}{\pi t/T},$$

Отсюда не трудно получить, что его спектральная плотность имеет вид:

$$S(\omega) = \begin{cases} AT, & |\omega| \leq \pi/T \\ 0, & |\omega| > \pi/T \end{cases} .$$



Преобразование Фурье является одним из важнейших ортогональных преобразований, используемых в цифровой обработке сигналов. Действительно, вполне физически ясен смысл перехода от временного описания исходного сигнала к его частотному описанию. Кроме того, двумерное преобразование Фурье описывает не что иное, как дифракцию электромагнитных и упругих волн в дальней зоне (дифракцию Фраунгофера) – т.е. на большом (по сравнению с размерами источника и длиной волны) расстоянии от источника [14,16,25].

## 2.5. Свойства преобразования Фурье

### 1. *Линейность:*

если  $S(t) = af(t) + bg(t)$  то:

$$S(\omega) = aF(\omega) + bG(\omega) . \quad (2.10)$$

### 2. *Инвариантность к линейному смещению (задержке) сигнала:*

$\Delta t$  – время задержки:

$S_1(t) = S(t - \Delta t)$ ;  $S_1$  – задержанная на  $\Delta t$  копия сигнала  $S(t)$ , тогда:

$$\begin{aligned} S_1(\omega) &= \int_{-\infty}^{\infty} S(t - \Delta t) e^{-j\omega t} dt = \int_{-\infty}^{\infty} S(t - \Delta t) e^{-j\omega(t - \Delta t)} d(t - \Delta t) \cdot e^{-j\omega\Delta t} = \\ &= S(\omega) \underbrace{e^{-j\omega\Delta t}} \end{aligned} \quad (2.11)$$

линейный фазовый множитель.

Отсюда следует, что амплитудный спектр сигнала не изменится при любой его задержке (линейный сдвиг). Фазовый спектр приобретает дополнительное слагаемое  $-\omega\Delta t$ , линейно зависящее от частоты.

### 3. *Масштабируемость спектральной плотности*

Пусть  $S_1(t) = S(at)$ , где  $a$  – масштабирующий множитель, при  $|a| > 1$  сигнал сжимается, при  $|a| < 1$  – растягивается, кроме того если  $a < 0$ , то дополнительно происходит зеркальное отражение сигнала по вертикальной оси.

Для  $a > 0$ :

$$S_1(\omega) = \int_{-\infty}^{\infty} S(at)e^{-j\omega t} dt = \frac{1}{a} \int_{-\infty}^{\infty} S(at)e^{-j\frac{\omega}{a}t} dt = \frac{1}{a} S\left(\frac{\omega}{a}\right)$$

Для  $a < 0$ :

$$S_1(\omega) = -\frac{1}{a} S\left(\frac{\omega}{a}\right) \quad (2.12)$$

или  $S_1(\omega) = \frac{1}{|a|} S\left(\frac{\omega}{a}\right); a \neq 0$

И для  $a = -1$ :

$$S_1(\omega) = S^*(\omega)$$

4. *Дифференцирование сигнала:*

$$S_1(\omega) = \frac{dS}{dt} = \lim_{\xi \rightarrow 0} \frac{S(t + \xi) - S(t)}{\xi}$$

Тогда

$$S_1(\omega) = j\omega S(\omega) \quad (2.13)$$

При дифференцировании низкие частоты ослабляются, а высокие усиливаются. Фазовый спектр сдвигается на  $\pi/2$  для положительных частот и на  $-\pi/2$  – для отрицательных.

5. *Интегрирование сигнала*

$$S_1(\omega) = \frac{S(\omega)}{j\omega} \text{ – это справедливо для сигналов, не содержащих постоянных}$$

составляющих, т.е. если  $\int_{-\infty}^{\infty} S(t)dt = 0$ .

В противном случае появляется дополнительное слагаемое от постоянной составляющей в виде  $\delta$ - функции на частоте  $\omega = 0$ .

$$S_1(\omega) = \frac{S(\omega)}{j\omega} + \pi S(0)\delta(\omega) \quad (2.14)$$

При этом происходит ослабление высоких частот и усиление низкочастотных гармоник.

6. *Спектр свёртки двух сигналов:*

Свёртка двух сигналов определяется как:

$$S_1(t) = \int_{-\infty}^{\infty} S(t_1)g(t-t_1)dt_1, \text{ тогда спектральная плотность свёртки двух}$$

сигналов есть:

$$S_1(\omega) = S(\omega) \cdot G(\omega) \quad (2.15)$$

7. *Спектральная плотность от произведения двух сигналов*

Пусть  $S_1(t) = S(t) \cdot g(t)$

Тогда спектральная плотность такого сигнала равна:

$$S_1(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega_1) G(\omega - \omega_1) d\omega_1, \quad (2.16)$$

т.е. является свёрткой спектральных плотностей двух сигналов.

### 8. Эффект переноса спектра

Умножим исходный сигнал на гармоническую функцию:

$$S_1(t) = S(t) \cdot \cos(\omega_0 t + \varphi_0)$$

и попытаемся найти спектральную плотность такого сигнала:

$$S_1(\omega) = \int_{-\infty}^{\infty} S(t) \cdot \cos(\omega_0 t + \varphi_0) e^{-j\omega t} dt.$$

Представим  $\cos x$  в виде:

$$\cos x = \frac{1}{2} (e^{jx} + e^{-jx}) \text{ – на основе формулы Эйлера, тогда:}$$

$$\begin{aligned} S_1(\omega) &= \int_{-\infty}^{\infty} S(t) \cdot \frac{e^{j\omega_0 t + j\varphi_0} + e^{-j\omega_0 t - j\varphi_0}}{2} \cdot e^{-j\omega t} dt = \\ &= \frac{1}{2} \int_{-\infty}^{\infty} S(t) e^{j\varphi_0} e^{-j(\omega - \omega_0)t} dt + \frac{1}{2} \int_{-\infty}^{\infty} S(t) e^{-j\varphi_0} e^{-j(\omega + \omega_0)t} dt = \\ &= \frac{1}{2} e^{j\varphi_0} S(\omega - \omega_0) + \frac{1}{2} e^{-j\varphi_0} S(\omega + \omega_0) \end{aligned} \quad (2.17)$$

### 9. Равенство Парсеваля или закон сохранения энергии:

$$\begin{aligned} E_1(t) &= \int_{-\infty}^{\infty} S^2(t) dt \\ E_2(\omega) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} |S(\omega)|^2 d\omega \\ E_1 &\Leftrightarrow E_2 \end{aligned}$$

Однако на практике сигнал имеет конечную длительность, т.е. финитен, и  $E_1 = \int_0^T S^2(t) dt$  – является определённым интегралом, т.е. числом.

Величина же  $E_2(\omega)$  является неопределённым интегралом, т.е. функцией.

Зададим некоторую полосу частот  $[-\omega_{\max}, \omega_{\max}]$ , в пределах которой передаётся подавляющая доля энергии сигнала (до 90÷95). Ширина полосы частот  $\Delta\omega = 2\omega_{\max}$  называется практической шириной спектра сигнала.

Тогда  $E_2 = \frac{1}{2\pi} \int_{-\omega_{\max}}^{\omega_{\max}} |S(\omega)|^2 d\omega$  – также является определённым интегралом

Поэтому равенство Парсеваля приобретает вид:

$$\frac{1}{2\pi} \int_{-\omega_{\max}}^{\omega_{\max}} |S(\omega)|^2 d\omega = (1 - \xi) E_1$$

где  $\xi$  – величина, определяющая долю потери энергии вне пределов полосы практической ширины спектра.

### 2.5.1. Фурье-анализ неинтегрируемых сигналов

Спектральная плотность  $\delta$ -функции

$$S(\omega) = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1$$

или  $\delta(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega t} d\omega$ .

Потенциальный сигнал (константа)

$$S(\omega) = \int_{-\infty}^{\infty} A e^{-j\omega t} dt = 2\pi A \delta(\omega).$$

Спектральная плотность единичного скачка (формула Хэвисайда)

$$S(\omega) = \int_{-\infty}^{\infty} \sigma(t) e^{-j\omega t} dt = \pi \delta(\omega) - \frac{1}{j\omega}.$$

Спектральная плотность гармонического сигнала

$$S(t) = A \cos(\omega_0 t + \varphi_0).$$

Воспользуемся формулой (2.16)

$$\begin{aligned} S(\omega) &= \int_{-\infty}^{\infty} A \cos(\omega_0 t + \varphi_0) e^{-j\omega t} dt = \int_{-\infty}^{\infty} A \frac{e^{j\omega_0 t + j\varphi_0} + e^{-j\omega_0 t - j\varphi_0}}{2} \cdot e^{-j\omega t} dt = \\ &= \int_{-\infty}^{\infty} \frac{A}{2} e^{j\varphi_0} \cdot e^{-j(\omega - \omega_0)t} dt + \int_{-\infty}^{\infty} \frac{A}{2} e^{-j\varphi_0} \cdot e^{-j(\omega + \omega_0)t} dt = A\pi e^{j\varphi_0} \delta(\omega - \omega_0) + A\pi e^{-j\varphi_0} \delta(\omega + \omega_0) \end{aligned}$$

Спектральная плотность комплексной экспоненты:

$$S(t) = A \exp(j\omega_0 t) \text{ – сигнал комплексный}$$

$$S(\omega) = 2\pi A \delta(\omega - \omega_0) \text{ – спектральная плотность не симметрична!}$$

### 2.6. Интегральное преобразование Хартли

Для одномерного случая прямое преобразование Хартли может быть определено как [3]

$$H(\xi) = \int_{-\infty}^{\infty} f(x) [\cos 2\pi \xi x + \sin 2\pi \xi x] dx \tag{2.18}$$

и, соответственно, обратное преобразование Хартли

$$f(x) = \int_{-\infty}^{\infty} H(\xi) [\cos 2\pi \xi x + \sin 2\pi \xi x] d\xi \tag{2.19}$$

Сравним эти выражения с (2.6.) и (2.9), разложив ядро по формуле Эйлера на действительную и мнимую части (т.е.  $\sin$  и  $\cos$  компоненты):

$$F(\xi) = \int_{-\infty}^{\infty} f(x) [\cos 2\pi\xi x - j \sin 2\pi\xi x] dx \quad (2.20)$$

$$f(x) = \int_{-\infty}^{\infty} F(\xi) [\cos 2\pi\xi x + j \sin 2\pi\xi x] d\xi$$

Из анализа (2.18) - (2.20), можно сделать следующие выводы:

1) Преобразование Хартли является преобразованием с действительным ядром;

2) Прямое и обратное преобразование Хартли вычисляются идентично;

3) Квадрат модуля преобразования Фурье  $|F(\nu)|^2$  равен:

$$|F(\nu)|^2 = \frac{H^2(\nu) + H^2(-\nu)}{2} \quad (2.9)$$

4) Действительная и мнимая компоненты преобразования Фурье могут быть вычислены на основе преобразования Хартли весьма простым образом:

$$\operatorname{Re}\{F(\xi)\} = \frac{H(\xi) + H(-\xi)}{2} \quad (2.22)$$

$$\operatorname{Im}\{F(\xi)\} = \frac{H(\xi) - H(-\xi)}{2} \quad (2.23)$$

5) Если  $f(x)$  - четная (т.е.  $f(-x)=f(x)$ ), то:

$$H(\xi) \equiv \operatorname{Re}\{F(\xi)\}, \quad \operatorname{Im}\{F(\xi)\} \equiv 0.$$

Основные свойства преобразования Хартли соответствуют преобразованию Фурье:

1) Инвариантность к сдвигу (модуль  $H^2(\xi) + H^2(\xi)$  - неизменен).

2) Так же, как и для преобразования Фурье, для преобразования Хартли справедливы следующие соотношения согласно теореме масштабов:

$$f(x) \rightarrow H(\xi)$$

$$f(m_x x) \rightarrow H_1(\xi) = H(\xi/m_x)/m_x$$

3) Так же, как и для преобразования Фурье, для преобразования Хартли справедлива Теорема Парсеваля.

Отличие от Фурье - преобразования заключается в иной трактовке теоремы о свертке:

Если заданы функции  $f(x)$  и  $g(x)$ , причем  $H(\xi)$  и  $G(\xi)$  - соответственно их спектры Хартли:

$$f(x) \rightarrow H(\xi),$$

$$g(x) \rightarrow G(\xi),$$

то их свертка вычисляется следующим образом [3]:

1) вычисляются функции  $H(-\xi)$  и  $G(-\xi)$ ;

2) формируется функция:

$$\Phi(\xi) = \frac{[(H(\xi) + H(-\xi))G(\xi) + (H(\xi) - H(-\xi))G(-\xi)]}{2}$$

3) вычисляется преобразование Хартли от функции  $\Phi(\xi)$ .  
Очевидно, что если функция  $g(x)$  - четная, то:

$$\begin{aligned} G(\xi) &\equiv G(-\xi) \\ \Phi(\xi) &= H(\xi)G(\xi), \end{aligned}$$

Если и функция  $f(x)$  - четная, то:

$$\begin{aligned} H(\xi) &\equiv H(-\xi) \\ \Phi(\xi) &= G(\xi)H(\xi) \end{aligned}$$

Преобразование Хартли требует вычислений примерно вдвое меньшей сложности (поскольку его ядро действительная функция) и в то же время от его результата достаточно просто перейти к результату, эквивалентному результату преобразования Фурье. Поэтому на практике преобразование Хартли используется вместо преобразования Фурье в различных задачах ЦОС как некоторое искусственное синтетическое преобразование меньшей сложности, но обеспечивающее получение требуемого результата.

## 2.7. Случайные сигналы

В отличие от детерминированных сигналов, форма которых известна точно, мгновенные значения случайных сигналов не известны и могут быть предсказаны лишь с некоторой вероятностью. Характеристики таких сигналов являются статистическими.

В таком вероятностном описании нуждаются следующие два основных класса сигналов [17,19].

1. Шумы – хаотические изменяющиеся во времени сигналы, возникающие в различных физических системах из-за беспорядочного движения носителей.
2. Сигналы, несущие информацию, поскольку её смысл изначально неизвестен.

*Математическая модель* изменяющегося во времени случайного сигнала называется случайным процессом.

До регистрации (или приёма) случайный сигнал следует рассматривать как случайный процесс, представляющий собой множество или ансамбль функции  $x_i(t)$ , которые обладают некоторой статистической закономерностью.

Одна из таких функций, ставшая полностью известной после приёма сообщения, называется реализацией случайного процесса, она является уже не случайной, а детерминированной.



### 2.7.1. Модели случайных процессов

Для анализа свойств случайного процесса необходимо задать математическую модель такого процесса.

Рассмотрим примеры

1. *Гармонический сигнал со случайной начальной фазой*

$$x(t) = A \cos(\omega_0 t + \varphi)$$

$A$  – амплитуда – известна, т.е. детерминирована

$\omega_0$  – частота детерминирована

$\varphi$  – случайная начальная фаза, принимающая любое значение на интервале  $[0, 2\pi]$ .

При равномерном распределении такой начальной фазы на интервале  $0 \div 2\pi$  плотность вероятности:

$$P_\varphi(\varphi) = \frac{1}{2\pi}; 0 \leq 2\pi$$

2. *Случайный телеграфный сигнал*

Возможны значения  $\pm 1$ , переключение из одного состояния в другое происходит в случайные моменты времени.

Функция распределения вероятности, т.е. вероятности того, что за время  $t$  произойдет  $N$  переключений (случайная величина!) имеет вид:

$$P(N, t) = \frac{(\lambda t)^N}{N!} e^{-\lambda t},$$

где  $\lambda$  – параметр, определяющий среднюю частоту переключений.

Это выражение описывает закон Пуассона.

### 2.7.2. Вероятностные характеристики случайного процесса

#### Функциональные характеристики.

*Функция распределения вероятности* (cumulative distribution function CDF)

$$F(x, t_1) = P(X(t_1) \leq x) \quad (3.1)$$

Определить вероятность того, что в момент времени  $t_1$  значение случайного процесса  $X$  не превосходит  $x$

- а) CDF является неубывающей функцией;
- б) вероятность попадания значения случайного процесса в интервал  $[a, b]$ :

$$P(a < X(t_1) \leq b) = F(b, t_1) - F(a, t_1) \quad (3.2)$$

*Одномерная плотность вероятности* (probability density function, PDF)

$$P(x, t_1) = \frac{dF(x, t_1)}{dx} \quad (3.3)$$

Т.е. является производной от функции распределения и определяет характер скорости её изменения:

- а) плотность вероятности является неотрицательной функцией

б) вероятность попадания  $X(t_1)$  в произвольный интервал  $[a, b]$ :

$$P(a \leq X(t_1) \leq b) = \int_a^b P(x, t_1) dx \quad (3.4)$$

Очевидно, что:

$$\int_{-\infty}^{\infty} P(x, t_1) dx = P(-\infty < x(t_1) < \infty) = 1$$

### Числовые характеристики

*Математическое ожидание* (mean value) – это теоретическая оценка среднего взвешенного значения случайного процесса в момент времени  $t_1$ :

$$m_x(t_1) = M\{x(t_1)\} = \int_{-\infty}^{\infty} xp(x, t_1) dx \quad (3.5)$$

*Дисперсия* (variance) – характеризует среднюю мощность отклонений случайного процесса от его среднего  $m_x(t_1)$

$$D_x(t_1) = M\{[x(t_1) - m_x(t_1)]^2\} = M\{x^2(t_1)\} - m_x^2(t_1) = \int_{-\infty}^{\infty} x^2 p(x, t_1) dx - m_x^2(t_1) \quad (3.6)$$

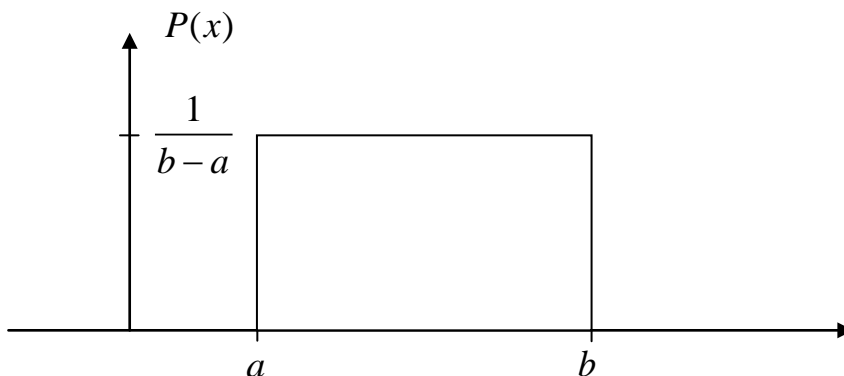
*Среднеквадратическое отклонение* (standard deviation):

$$\sigma_x(t_1) = \sqrt{D_x(t_1)} = \sqrt{M\{x^2(t_1)\} - m_x^2(t_1)} \quad (3.7)$$

### Примеры случайных процессов с различными законами распределения

1. *Равномерное распределение*: для такой случайной величины плотность вероятности является постоянной, т.е.

$$P_x = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & x < a; x > b \end{cases}$$



Функция распределения вероятности такой случайной величины на интервале  $[a, b]$  линейно возрастает от 0 до 1:

$$F(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases}$$

Математическое ожидание:

$$m_x = \int_a^b x \frac{1}{b-a} dx = \frac{a+b}{2};$$

Дисперсия: согласно (3.6) с учётом математического ожидания получаем:

$$D_x = \frac{(b-a)^2}{12}$$

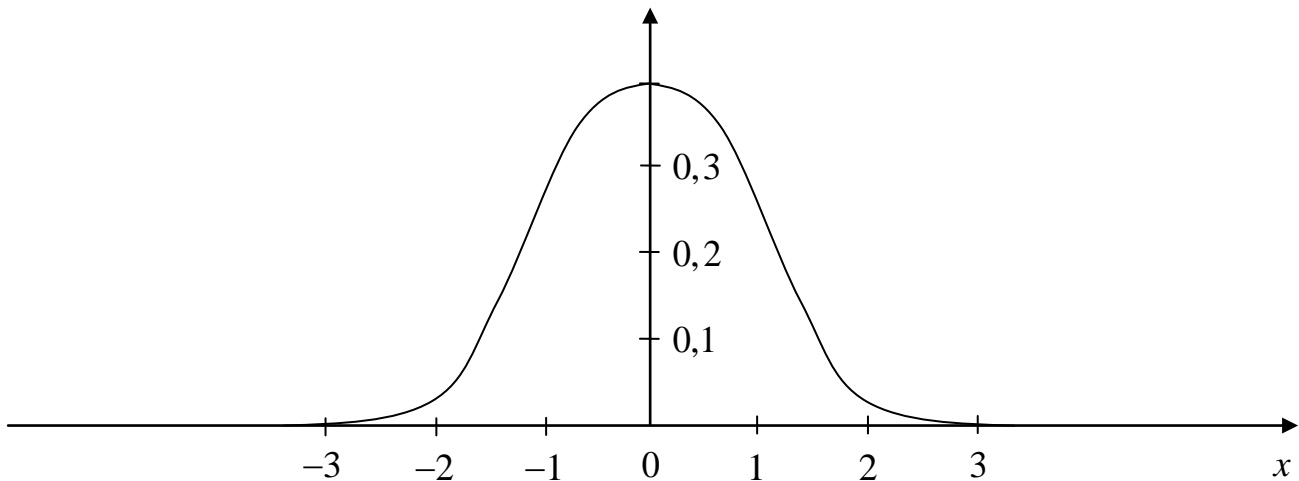
$$\text{СКО: } \sigma_x = \sqrt{D_x} = \frac{b-a}{2\sqrt{3}}.$$

2. *Нормативный закон распределения* – достаточно часто встречается на практике, например, он характерен для помех канала связи: при этом одномерная плотность вероятности нормальной случайной величины определяется как:

$$P(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{(x-m_x)^2}{2\sigma_x^2}},$$

где  $m_x$  и  $\sigma_x^2 = D_x$  – соответственно мат ожидание и дисперсия процесса (случайные величины).

График плотности вероятности  $P(x)$  в этом случае имеет вид (для  $m_x = 0$  и  $\sigma_x = 1$ ):



Функция распределения вероятности в этом случае обычно выражается через интеграл вероятности:

$$\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{(x')^2}{2}} dx'$$

$$F(x) = \Phi\left(\frac{x-m_x}{\sigma_x}\right)$$

В зарубежной литературе часто используется функция ошибок (error function):

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

между  $\Phi(x)$  и  $\operatorname{erf}(x)$  существует взаимосвязь:

$$\operatorname{erf}(x) = 2\Phi(x\sqrt{2}) - 1$$

$$\Phi(x) = \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)$$

С учётом этого функция распределения для нормального закона с математическим ожиданием  $m_x$  и дисперсией  $\sigma_x^2$ :

$$F(x) = \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{x - m_x}{\sigma_x \sqrt{2}}\right).$$

*Важное свойство:*

При суммировании достаточно большого числа равномошных статистических независимых случайных величин с произвольными плотностями распределения вероятности, плотность распределения вероятности суммы стремится к нормальной. Это положение носит название *центральной предельной теоремы*.

Кроме того, для математического анализа случайных величин полезным является то, что из некоррелированности гауссовых случайных величин следует их статистическая независимость.

### 3. КОРРЕЛЯЦИОННЫЙ АНАЛИЗ СИГНАЛОВ

Смысл корреляционного анализа состоит в определении количественной меры сходства различных сигналов. Для этого применяют корреляционные функции [15-17,19].

#### 3.1. Корреляционная функция (КФ):

$$B_s(t_1) = \int_{-\infty}^{\infty} S(t)S(t-t_1)dt \quad (3.1)$$

Эта функция определяет меру сходства между сигналом  $S(t)$  и его копией, имеющей произвольную задержку на время  $t_1$ . Чем больше величина (площадь) перекрытия сигнала  $S(t)$  с его копией, тем больше величина  $B_s(t_1)$ .

Свойства корреляционной функции:

1. При  $t_1 = 0$  значение КФ равно энергии сигнала:

$$B_s(0) = \int_{-\infty}^{\infty} S^2(t)dt = E$$

Размерность такой функции, если  $S(t)$  имеет размерность вольт, составляет  $B^2c$ .

2. КФ является чётной функцией:

$$B_s(t_1) = B_s(-t_1)$$

3. Значение КФ при  $t_1 = 0$  является максимально-возможным.

4. С увеличением аргумента  $|t_1|$  значение КФ убывает:

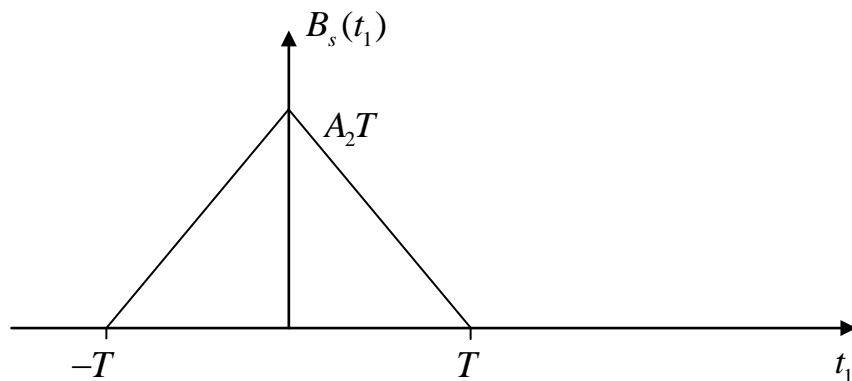
$$\lim_{t_1 \rightarrow \infty} B_s(t_1) = 0$$

5. Если  $S(t)$  не имеет разрывов, то и  $B_s(t_1)$  является непрерывной.

Пример:

Рассмотрим КФ прямоугольного импульса:

$$B_s(t_1) = \begin{cases} A^2 |T - |t_1||, & |t_1| \leq T \\ 0, & |t_1| > T \end{cases}$$



Для периодического сигнала, когда энергия его не ограничена, требуется рассмотреть значение КФ при сдвиге копии сигнала лишь в пределах одного периода  $T$  исходного сигнала:

$$B_s(t_1) = \frac{1}{T} \int_{-T/2}^{T/2} S(t)S(t-t_1)dt \quad (3.2)$$

Поэтому свойства КФ несколько изменяются, а именно, значение  $B_s(0)$  определяет *среднюю мощность сигнала*:

$$B_s(0) = \frac{1}{T} \int_{-T/2}^{T/2} S^2(t)dt = P_{cp} \text{ (размерность } B^2 \text{)}$$

Пример:

$S(t)$  – гармонический сигнал

$$S(t) = A \cos(\omega_0 t + \varphi_0); \quad T = \frac{2\pi}{\omega_0}$$

$$B_s(t_1) = \frac{\omega_0}{2\pi} \int_{-\pi/\omega_0}^{\pi/\omega_0} A \cos(\omega_0 t + \varphi_0) A \cos(\omega_0 (t - t_1) + \varphi_0) dt = \frac{A^2}{2} \cos(\omega_0 t_1)$$

### 3.2. Взаимная корреляционная функция

Определяет меру сходства между двумя различными сигналами  $S_1(t)$  и  $S_2(t)$ , сдвинутыми друг относительно друга на величину  $t_1$

$$B_{12}(t_1) = \int_{-\infty}^{\infty} S_1(t)S_2(t-t_1)dt \quad (3.3.)$$

Англоязычное название – CCF – cross-correlation function.

КФ является частным свойством ВКФ, когда  $S_1(t) = S_2(t)$ .

Свойства ВКФ:

1.  $B_{12}(t_1) = \sqrt{E_1 E_2}$ , где  $E_1$  и  $E_2$  – энергии сигналов  $S_1(t)$  и  $S_2(t)$ .

2.  $B_{12}(-t_1) = B_{21}(t_1)$ , т.е. изменение значения  $t_1$  эквивалентно взаимной перестановке сигналов.

3. Максимум  $B_{12}(t_1)$  может быть расположен при любом значении  $t_1$  (в зависимости от вида  $S_1(t)$  и  $S_2(t)$ ).

4. Для сигналов с конечной энергией  $\lim_{t_1 \rightarrow \infty} B_{12}(t_1) = 0$

5. Если  $S_1(t)$  и  $S_2(t)$  не имеют разрывов (скачков в виде  $\delta$ -функций), то ВКФ не имеет разрывов.

Для периодических сигналов ВКФ может быть определена, если  $S_1(t)$  и  $S_2(t)$  имеют одинаковый период.

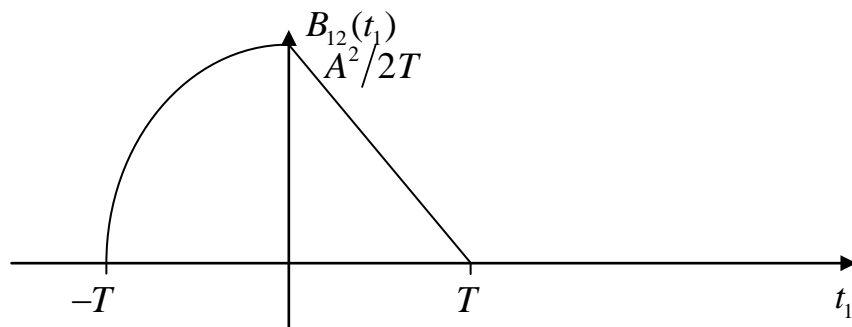
Пример:

Пусть

$$S_1(t) = \begin{cases} A, 0 \leq t \leq T \\ 0, t < 0; t > T \end{cases} \left| \text{прямоугольный импульс} \right.$$

$$S_2(t) = \begin{cases} At/T, 0 \leq t \leq T \\ 0, t < 0; t > T \end{cases} \left| \text{треугольный импульс} \right.$$

$$B_{12}(t_1) = \begin{cases} \frac{A^2}{2T}(T - t_1)^2; 0 \leq t_1 \leq T \\ \frac{A^2}{2T}(T^2 - t_1^2); -T \leq t_1 \leq 0 \\ 0; |t_1| > T \end{cases}$$



### 3.3. Взаимный спектр сигналов

Взаимный спектр сигналов определяется как преобразование Фурье от их ВКФ [17], т.е.

$$\begin{aligned} S_{12}(\omega) &= \int_{-\infty}^{\infty} B_{12}(t_1) e^{-j\omega t_1} dt_1 = \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} S_1(t) S_2(t - t_1) dt \right] dt_1 = \\ &= \int_{-\infty}^{\infty} S_1(t) \cdot e^{-j\omega t} \left[ \int_{-\infty}^{\infty} S_2(t - t_1) \cdot e^{j\omega(t - t_1)} d(t - t_1) \right] dt = S_1(\omega) \cdot S_2^*(\omega) \end{aligned} \quad (3.4)$$

Вывод: если спектры сигналов  $S_1(\omega)$  и  $S_2(\omega)$  не перекрываются (не имеют общей области по  $\omega$ ), то их взаимный спектр равен 0, и также равна нулю их ВКФ. Иначе говоря, сигналы с непересекающимися спектрами являются некоррелированными.

Для сигнала  $S(t)$  и его сдвинутой на величину  $t_1$  копии:

$$\begin{aligned} \int_{-\infty}^{\infty} B_s(t_1) e^{-j\omega t_1} dt_1 &= \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} S(t) S(t - t_1) e^{-j\omega t} dt \right] dt_1 = \\ &= \int_{-\infty}^{\infty} S(t) e^{-j\omega t} \left[ \int_{-\infty}^{\infty} S(t - t_1) e^{j\omega(t - t_1)} d(t - t_1) \right] dt = S(\omega) \cdot S^*(\omega) = |S(\omega)|^2 \end{aligned}$$

Вывод:

- 1) КФ связана как результат преобразования Фурье с энергетическим спектром сигнала.
- 2) КФ не зависит от фазового спектра сигнала  $S(t)$ .
- 3) По КФ нельзя восстановить исходный сигнал  $S(t)$  (из-за потери информации о фазе).

**Теорема Рэлея:**

$$\int_{-\infty}^{\infty} S_1(t)S_2(t)dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_1(\omega)S_2(\omega)d\omega \quad (3.5)$$

Откуда при  $S_1 = S_2 = S(t)$  получаем равенство Парсеваля (см. 1.6).

### 3.4. Корреляционные функции случайных процессов

Важное значение имеет анализ поведения ансамбля, т.е. совокупности реализаций случайной величины, в различные моменты времени, например,  $t_1$  и  $t_2$ . Для такого анализа исследуются два сечения случайного процесса. Совокупность таких сечений приводит к двумерной случайной величине:  $\{X(t_1), X(t_2)\}$ , которая описывается двумерной плотностью вероятностей  $P(x_1, x_2, t_1, t_2)$ . Тогда произведение вида  $P(x_1, x_2, t_1, t_2)dx_1dx_2$  представляет собой вероятность того, что реализация случайного процесса  $X(t)$  в момент времени  $t_1$  попадает в бесконечно малый интервал шириной  $dx_1$  в окрестности  $x_1$ , а в момент времени  $t_2$  – в интервал  $dx_2$  в окрестности  $x_2$ :

$$P(x_1, x_2, t_1, t_2)dx_1dx_2 = P\left\{|X(t_1) - x_1| \leq \frac{dx_1}{2}, |X(t_2) - x_2| \leq \frac{dx_2}{2}\right\}$$

Задание двумерной плотности вероятности  $P(x_1, x_2, t_1, t_2)$  позволяет определить ковариационную функцию [19]:

$$K_x(t_1, t_2) = M\{x(t_1)x(t_2)\}$$

Ковариационная функция  $K_x(t_1, t_2)$  случайного процесса  $X(t)$  представляет собой статистически усреднённое произведение значений случайной функции  $X(t)$  в момент времени  $t_1$  и  $t_2$ . При этом для каждой реализации случайного процесса произведение  $x(t_1)x(t_2)$  является некоторым числом. С помощью двумерной плотности вероятности такое усреднение произведений по всему множеству реализаций описывается так:

$$K_x(t_1, t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1x_2p(x_1, x_2, t_1, t_2)dx_1dx_2$$

При анализе случайных процессов часто необходимо исследовать их флуктуационную составляющую. Для этого используется корреляционная функция, которая представляет собой статистически усреднённое произведение значений центрированной случайной функции  $X(t) - m_x(t)$  в моменты времени  $t_1$  и  $t_2$ :



$$\begin{aligned}
R_x(t_1, t_2) &= M\{[x(t_1) - m_x(t_1)][x(t_2) - m_x(t_2)]\} = \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [x(t_1) - m_x(t_1)][x(t_2) - m_x(t_2)]p(x_1, x_2, t_1, t_2)dx_1dx_2 = \\
&= K_x(t_1, t_2) - m_x(t_1)m_x(t_2)
\end{aligned}$$

Корреляционная функция случайного процесса характеризует степень статистической связи значений для реализаций случайного процесса в моменты времени  $t_1$  и  $t_2$ .

Если  $t_1 = t_2$ , то тогда:

$$R_x(t, t) = K_x(t, t) = m_x^2(t) = D_x(t)$$

Если случайный процесс центрирован, то  $m_x(x) = 0$  и тогда

$$R_x(t_1, t_2) = K_x(t_1, t_2).$$

### ***Некоррелированность и статистическая независимость***

Под *статистической независимостью* двух случайных величин  $X_1$  и  $X_2$  понимается, что плотность вероятности одной случайной величины не зависит от того, какое значение принимает другая величина. В таком случае двумерная плотность вероятности представляет собой произведение одномерных плотностей вероятностей:

$$P(x_1, x_2) = P_1(x_1)P_2(x_2),$$

что определяет *условие статистической независимости*.

При наличии статистической связи между случайными величинами статистические свойства каждой из них зависят от значения, принимаемого другой величиной.

Мерой *линейной статистической связи* между случайными величинами является *коэффициент корреляции*:

$$r_{12} = \frac{M\{x_1, x_2\} - M\{x_1\}M\{x_2\}}{\sqrt{D\{x_1\}D\{x_2\}}}$$

При этом  $|r_{12}| \leq 1$ . Предельные значения  $r_{12} = \pm 1$  достигаются, если реализации случайных величин  $x_1$  и  $x_2$  жестко связаны линейным соотношением вида  $x_2 = ax_1 + b$ , причём знак коэффициента и определяет знак  $r_{12}$ .

Отсутствие линейной статистической связи означает отсутствие коррелированности случайных величин  $x_1$  и  $x_2$ . При этом  $r_{12} = 0$ .

Таким образом для некоррелированных случайных величин:

$$M\{x_1, x_2\} = M\{x_1\}M\{x_2\}$$

Из *статистической независимости* следует *некоррелированность* двух случайных величин. Обратное неверно, т.е. некоррелированные случайные величины могут быть зависимыми.

*Пример.*

Случайные величины  $x_1 = \cos(\varphi)$  и  $x_2 = \sin(\varphi)$ , где  $\varphi$  – случайная величина. Очевидно, что  $x_1$  и  $x_2$  являются статистически зависимыми, однако,  $r_{12} = 0$ .

### 3.4.1. Стационарные и эргодические случайные процессы

Стационарный случайный процесс – это процесс, статистические характеристики которого одинаковы во всех временных сечениях.

Случайный процесс *строго стационарен* (или *стационарен в узком смысле*), если его многомерная плотность вероятности  $P(x_1, x_2, \dots, x_n, t_1, t_2, \dots, t_n)$  ( $n$  – произвольная размерность,  $n \geq 3$ ) не изменяется при одновременном сдвиге всех временных сечений  $t_1, t_2, \dots, t_n$  на одинаковую величину  $\Delta t$ :

т.е.:

$$P(x_1, x_2, \dots, x_n, t_1, t_2, \dots, t_n) = P(x_1, x_2, \dots, x_n, t_1 + \Delta t, t_2 + \Delta t, \dots, t_n + \Delta t)$$

для любого  $\Delta t$ .

Процесс *стационарен в широком смысле*, если такое свойство независимости от временного сдвига обеспечивается лишь для одномерной и двумерной плотности вероятности.

Для стационарного случайного процесс математическое ожидание и дисперсия не зависят от моментов времени  $t_1$  и  $t_2$ , а лишь от интервала  $\Delta t = t_2 - t_1$  между ними, т.е.

$$R_x(t_1, t_2) = R_x(t_2 - t_1) = R_x(\Delta t)$$

Также для стационарного процесса:

$$R_x(-\Delta t) = R_x(\Delta t)$$

и, кроме того,

$$|R_x(\Delta t)| \leq R_x(0) = D_x$$

Коэффициент корреляции в этом случае:

$$r_x(\Delta t) = \frac{R_x(\Delta t)}{D_x}$$

Стационарным является любой случайный процесс, реализации которого являются периодическими функциями. Например,  $x(t) = A \cos(\omega_0 t + \varphi)$  – стационарен,  $\varphi$  – случайная величина.

*Стационарный случайный процесс называется эргодическим (ergodic)*, если при определении любых его статистических характеристик *усреднение по множеству (ансамблю) реализаций эквивалентно усреднению времени одной, теоретически бесконечной реализации*.

*Математическое ожидание* эргодического случайного процесса равно *постоянной составляющей* любой его реализации, а *дисперсия* эргодического процесса – *смысл мощности флуактуационной составляющей*.

Достаточной условие эргодичности случайного процесса, стационарного в широком смысле, является стремление к нулю его корреляционной функции с

ростом  $\Delta t$ . Так, например, случайный процесс  $x(t) = A \cos(\omega_0 t + \varphi)$  – является стационарным и эргодическим.

### 3.5. Спектральные характеристики случайных процессов

Для каждой реализации случайного процесса можно определить свою спектральную плотность  $S_x(\omega)$ , выполнив прямое преобразование Фурье. Для множества (ансамбля) реализаций можно определить статистически усреднённую спектральную плотность  $\overline{S_x(\omega)}$ :

$$\overline{S_x(\omega)} = \overline{\int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt} = \int_{-\infty}^{\infty} \overline{x(t) e^{-j\omega t}} dt = \int_{-\infty}^{\infty} m_x(t) e^{-j\omega t} dt$$

Таким образом, усреднённая спектральная плотность случайного процесса представляет собой спектр его детерминированной составляющей (математического ожидания). Для центрированного случайного процесса  $m_x(t) = 0$  и  $\overline{S_x(\omega)} = 0$ .

*Вывод* – вычисление  $\overline{S_x(\omega)}$  не несёт информации о собственно случайной составляющей процесса, так как фазы спектральных составляющих в различных реализациях независимы и случайны.

Рассмотрим *спектральную плотность мощности* случайного процесса, так как мощность *не зависит* от соотношения фаз спектральных составляющих.

Пусть  $x(t)$  – центрированный случайный процесс и ограничим длительность его реализации конечным интервалом  $T = [-T/2; T/2]$ . Найдём для реализации  $x(t)$  на этом интервале спектральную плотность  $x_T(\omega)$  через прямое преобразование Фурье.

Согласно равенству Парсеваля:

$$E_T = \int_{-T/2}^{T/2} x^2(t) dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |x_T(\omega)|^2 d\omega$$

Теперь определим среднюю мощность  $P_T$  реализации на данном временном интервале:

$$P_T = \frac{E_T}{T} = \langle x^2(t) \rangle_{|t| \leq T/2} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{|x_T(\omega)|^2}{T} d\omega$$

При увеличении времени  $T$  энергия всей реализации неограниченно возрастает, а средняя мощность стремится к некоторому пределу.

Пусть  $T \rightarrow \infty$ , тогда получаем:

$$\langle x^2(t) \rangle = \frac{1}{2\pi} \int_{-\infty}^{\infty} \lim_{T \rightarrow \infty} \frac{|x_T(\omega)|^2}{T} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} W(\omega) d\omega,$$

где  $W(\omega) = \lim_{T \rightarrow \infty} \frac{|x_T(\omega)|^2}{T}$  – спектральная плотность средней мощности или спектральная плотность мощности – *power spectral density* (PSD).

Для центрированного эргодического процесса средняя мощность для любой реализации равна дисперсии процесса, т.е.:

$$D_x = \frac{1}{2\pi} \int_{-\infty}^{\infty} W(\omega) d\omega$$

Заметим, что  $W(\omega)$  – вещественная функция и не содержит информации о фазах спектральных составляющих, тем самым она не позволяет восстановить отдельные реализации случайного процесса.

### 3.5.1. Теорема Винера-Хинчина

Взаимосвязь между корреляционной функцией случайного процесса и его спектральной плотностью мощности устанавливает теорема Винера-Хинчина [16,19]:

$$R(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} W(\omega) e^{j\omega\tau} d\omega$$

Поскольку  $R(\tau)$  и  $W(\omega)$  являются вещественными и чётными функциями, то:

$$R(\tau) = \frac{1}{\pi} \int_0^{\infty} W(\omega) \cos(\omega\tau) d\omega;$$

$$W(\omega) = 2 \int_0^{\infty} R(\tau) \cos(\omega\tau) d\tau$$

*Интервал корреляции* – это числовая характеристика, которая служит для оценки «скорости» изменения реализаций случайного процесса. Эта величина определяется следующим образом:

$$\tau_k = \frac{1}{R(0)} \int_0^{\infty} |R(\tau)| d\tau = \int_0^{\infty} r(\tau) d\tau$$

Если имеется информация о поведении какой-либо случайной величины в прошлом, то возможен вероятностный прогноз случайного процесса на время порядка  $\tau_k$ .

Интервал корреляции и практическая ширина случайного сигнала связаны соотношением:

$$\Delta\omega_{\text{ш}} \cdot \tau_k \sim 2\pi$$

*Белый шум* – спектральная плотность мощности его постоянна:

$$W(\omega) = W_0 = \text{const}$$

Согласно теореме Винера-Хинчина:

$$R(\tau) = \frac{W_0}{2\pi} \int_{-\infty}^{\infty} e^{j\omega\tau} d\omega = W_0 \delta(\tau)$$

т.е. корреляционная функция белого шума равна 0, кроме точки  $\tau = 0$ , где является  $\delta$ -функцией. В несовпадающие моменты времени значения белого шума не коррелированы.

Такой шум является математической абстракцией и не имеет физической модели, что объясняется бесконечной дисперсией белого шума (т.е. средней мощности). Однако если исследуемая полоса пропускания существенно уже практической ширины спектра шума, воздействующего на некоторую систему, то можно для упрощения реальный случайный процесс заменить белым шумом.

### 3.6. Комплексная огибающая сигнала

Рассмотрим сигнал, у которого одновременно осуществляется по какому-либо закону изменение (модуляция) амплитуды и фазы:

$$S(t) = A(t) \cos(\omega_0 t + \varphi(t)) \quad (2.6)$$

В (2.6)  $A(t)$  называют *амплитудой огибающей*;  $\varphi(t)$  – *фазовой функцией* сигнала  $S(t)$ . *Полная фаза* сигнала  $S(t)$  определяется как:

$$\Psi(t) = \omega_0 t + \varphi(t)$$

Сигнал вида (2.6) можно представить как вещественную часть импульсной функции:

$$S(t) = \operatorname{Re} \left[ A(t) \cdot e^{j(\omega_0 t + \varphi(t))} \right] = \operatorname{Re} \left[ A(t) \cdot e^{j\omega_0 t} \cdot e^{j\varphi(t)} \right] \quad (2.7)$$

В (2.7)  $e^{j\omega_0 t}$  определяет собой несущий немодулированный гармонический сигнал, множители  $A(t)$  и  $e^{j\varphi(t)}$  несут информацию об амплитудной огибающей и фазовой функции сигнала. Их произведение называют комплексной огибающей сигнала:

$$\dot{A}_m(t) = A(t) \cdot e^{j\varphi(t)}. \quad (2.8)$$

Для отличия того, что эта функция комплексная, обозначим её с точкой.

Введём понятие комплексного сигнала  $\dot{S}_m(t)$  (иногда его называют аналитическим сигналом).

Произвольный сигнал  $S(t)$  представляет собой действительную (вещественную) часть сигнала  $\dot{S}_m(t)$ .

$$S(t) = \operatorname{Re} \left[ \dot{S}_m(t) \right].$$

Для того, чтобы было возможным определить как амплитуду, так и фазу сигнала, необходима мнимая часть исходного комплексного сигнала:

$$S_{\perp}(t) = \operatorname{Im} \left[ \dot{S}_m(t) \right],$$

которая называется сопряжённым сигналом или *квадратурным дополнением*.

Тогда:

$$\dot{S}_m(t) = \operatorname{Re} \left[ \dot{S}_m(t) \right] + j \operatorname{Im} \left[ \dot{S}_m(t) \right] = S(t) + jS_{\perp}(t).$$

Квадратурное дополнение  $S_{\perp}(t)$  можно получить из  $S(t)$  с помощью *преобразования Гильберта*, которое имеет вид:

$$S_{\perp}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{S(t_1)}{t - t_1} dt_1.$$

Точно также, с помощью обратного преобразования Гильберта может быть по  $S_{\perp}(t)$  получен сигнал  $S(t)$ :

$$S(t) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{S_{\perp}(t_1)}{t - t_1} dt_1.$$

## 4. ПЕРЕХОД ОТ АНАЛОГОВЫХ СИГНАЛОВ К ЦИФРОВЫМ

Переход от непрерывных аналоговых сигналов к цифровым осуществляется с помощью процедур дискретизации и квантования, выполняемых последовательно друг за другом. Их совместное применение называют аналого-цифровым преобразованием [19,20].

Процесс перехода от непрерывной области изменения аргумента (задания функции) к конечному множеству отдельных значений аргумента называется дискретизацией.

Процесс перехода от непрерывной области изменения функции к конечному множеству определенных значений называется квантованием.

### 4.1. Дискретизация сигналов

Дискретизация аналогового сигнала может осуществляться тремя способами:

- 1) как процедура выбора отсчетов сигнала в фиксированные моменты времени, следующие через равные промежутки времени  $\Delta t$  (интервал  $\Delta t$  называют шагом дискретизации, а обратная величина  $f_D = 1/\Delta t$  носит название частоты дискретизации), такой способ носит название равномерной дискретизации;
- 2) как процедура выбора отсчетов сигнала в моменты времени, следующие друг за другом через не равные интервалы времени (чаще всего величину интервала выбирают в зависимости от скорости изменения сигнала на различных интервалах), так называемая адаптивная дискретизация;
- 3) как процедура выбора отсчетов в фиксированные моменты времени, задаваемые случайным образом по тому или иному закону (стохастическая дискретизация).

Простейшим случаем является дискретизация с равномерным шагом, поэтому на практике такой способ применяется наиболее широко.

Переход от непрерывного сообщения к дискретному осуществляется с потерей информации. Восстановление непрерывного сигнала по дискретным значениям на приемной стороне и устранение потерь информации зависит от параметров дискретизации, способа восстановления сигнала и от свойств сигнала.

Любой гармонический сигнал может быть однозначно представлен дискретными значениями (отсчётами), если частота такого сигнала меньше  $f_D/2$ . Частота  $f_N = f_D/2$  носит название *частоты Найквиста*.

Возможны три различных ситуации при восстановлении сигнала после дискретизации:

а)  $f_s < f_N$  – возможно правильное восстановление сигнала

б)  $f_s = f_N$  – восстановленный сигнал по дискретным отсчётам будет иметь ту же частоту, что и до дискретизации, однако фаза и амплитуда сигнала могут быть искажены.

в)  $f_s > f_N$  – восстановленный сигнал так же будет гармоническим, но его частота будет иной. Иначе говоря, проявляется эффект *появления ложных частот* (aliasing).

Отсюда следует, что при дискретизации возможны искажения, связанные с потерей информации о величине сигнала вне моментов измерения отсчетов.

Рассмотрим вначале спектр дискретного сигнала. Пусть сигнал имеет вид:

$$S(t) = \sum_{k=-\infty}^{\infty} x_k \delta(t - t_k) = \sum_{k=-\infty}^{\infty} x_k \delta(t - k\Delta t)$$

Такому дискретному сигналу соответствует спектральная плотность, описываемая как

$$S(\omega) = \sum_{k=-\infty}^{\infty} x_k e^{-j\omega k\Delta t}.$$

При этом учтены следующие факторы:

1) спектр  $\delta$ -функции равен 1

2) задержка сигнала во времени на  $k\Delta t$  приводит к умножению сигнала на комплексную экспоненту ( см. раздел 2.5).

Важнейшее свойство спектра дискретного сигнала является его периодичность с периодом  $2\pi$ , т.е.

$$S(\omega \pm 2\pi) = S(\omega)$$

$$S_\delta(\omega) = \frac{1}{\Delta t} \sum_{n=-\infty}^{\infty} S\left(\omega - \frac{2\pi n}{\Delta t}\right).$$

Это свойство иллюстрируется на рис.4.1.

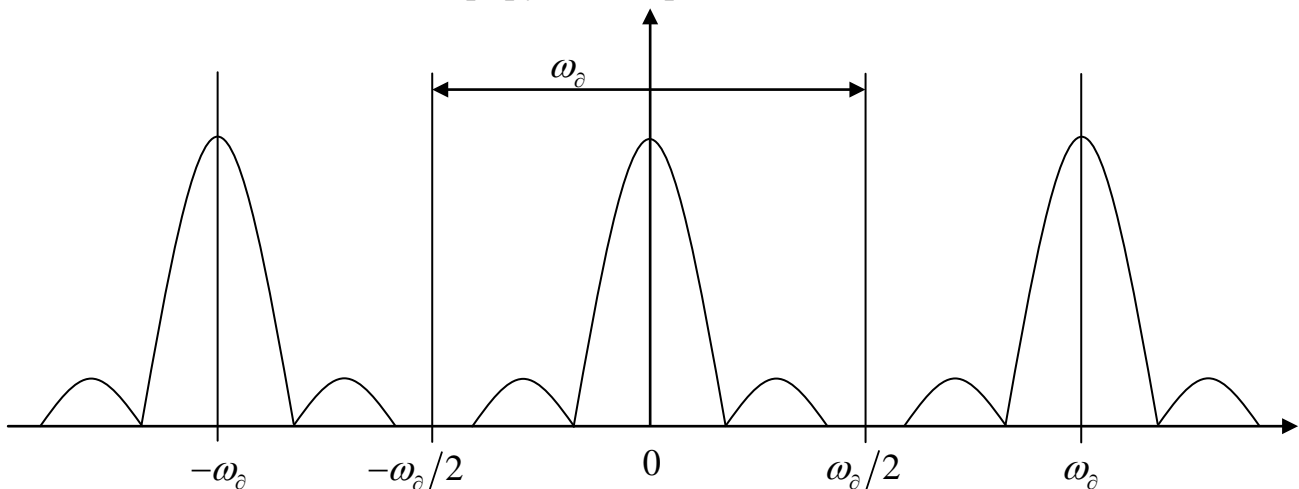


Рис.4.1. Амплитудный спектр дискретного сигнала.



### 4.1.1. Влияние формы дискретизирующих импульсов

При дискретизации мы полагаем, что отсчёты получены за счёт использования фильтрующего свойства  $\delta$ -функции. Однако в действительности при дискретизации каждый импульс имеет конечную длительность. Определим, как влияет вид импульса на спектр дискретного сигнала.

$$S_{\sigma}(t) = \sum_{k=-\infty}^{\infty} S(k\Delta t)S_0(t - k\Delta t),$$

где  $S_0(t - k\Delta t)$  – производный импульс, наблюдаемый в момент времени  $k\Delta t$ . В силу свойств спектральной плотности, результат преобразования Фурье от свёртки двух сигналов есть произведение их плотностей, т.е.

$$S_{\sigma}(\omega) = \frac{S_0(\omega)}{\Delta t} \sum_{n=-\infty}^{\infty} S\left(\omega - \frac{2\pi n}{\Delta t}\right)$$

т.е. при отличии формы импульсов при дискретизации приводит к искажению дискретного спектра.

Пусть  $S_0(\Delta t)$  – является единичным импульсом длительностью  $\Delta t$ .

В этом случае:

$$S_0(\omega) = \int_{-\infty}^{\infty} S_0(t)e^{-j\omega t} dt = \Delta t \frac{\sin(\omega \Delta t/2)}{\omega \Delta t/2} e^{-j\omega t} \quad (4,1)$$

### 4.1.2. Теорема Котельникова

Условие, при котором возможно восстановление сигнала без потерь, определяется из теоремы Котельникова [16,19,21].

**Прямая формулировка теоремы Котельникова.** Если сигнал  $S(t)$  имеет финитную спектральную плотность, локализованную в полосе частот  $\omega_{\max} = 2\pi f_{\max}$ , то он может быть без потерь представлен дискретными отсчётами  $S(k\Delta t)$ , удовлетворяющих условию:

$$\Delta t \leq \frac{1}{2f_{\max}} = \frac{\pi}{\omega_{\max}} \quad (4.2)$$

В зарубежной литературе теорему Котельникова чаще называют теоремой Найквиста или теоремой отсчётов.

Исходный сигнал в этом случае восстанавливается в следующем виде:

$$S(t) = \sum_{k=-\infty}^{\infty} S(k\Delta t) \frac{\sin\left(\pi \frac{t - k\Delta t}{\Delta t}\right)}{\pi \frac{t - k\Delta t}{\Delta t}} \quad (4.3).$$

В общем случае, можно записать, что сигнал восстанавливается с помощью системы восстанавливающих функций:

$$S(t) = \sum_{k=-\infty}^{\infty} S(k\Delta t)\varphi_k(t)$$

В случае теоремы Котельникова восстанавливающие функции имеют вид:

$$\varphi_k(t) = \frac{\sin\left(\pi \frac{t - k\Delta t}{\Delta t}\right)}{\pi \frac{t - k\Delta t}{\Delta t}}$$

Пусть сигнал имеет вид  $S(t) = A(t)\cos(n\omega_0 t)$  – согласно свойствам преобразования Фурье (см. раздел 2.5) для восстановления такого сигнала можно воспользоваться полосой частот шириной  $\omega_0$  и со средней частотой  $n\omega_0$ . В спектральной области такой полосе будут соответствовать две сдвинутые копии спектра  $S(\omega \pm 2\pi n/\Delta t)$ . Восстанавливающая функция будет иметь вид:

$$\varphi_k(t) = \frac{\sin\left(\pi \frac{t - k\Delta t}{\Delta t}\right)}{\pi \frac{t - k\Delta t}{\Delta t}} \cdot \cos\left(\frac{2\pi n t}{\Delta t}\right).$$

**Обратная формулировка теоремы Котельникова/** Если  $f(x)$  задана в ограниченной области  $-x_{\max} \leq x \leq x_{\max}$ , то ее спектр  $F(\nu)$  полностью определен набором отсчетов в точках, равноотстоящих друг от друга на расстоянии  $1/2 x_{\max}$ .

Поясним выбор шагов дискретизации по теореме Котельникова на рис. 4.2.

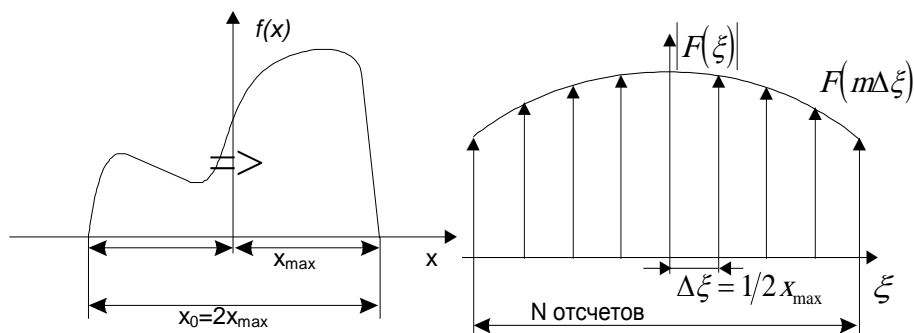


Рис.4.2. Выбор шага дискретизации по теореме Котельникова

При дискретизации согласно теореме Котельникова исходная функция  $f(x)$  может быть получена по ее дискретным значениям по формуле:

$$f(x) = \sum_{k=-\infty}^{\infty} f(k\Delta x) \frac{\sin 2\pi \nu \max(x - k\Delta x)}{\nu \max(x - k\Delta x)},$$

причем шаг дискретизации составляет

$$\Delta x = \frac{1}{2\nu_{\max}} = \frac{\pi}{\omega_{\max}}$$

Однако согласно теории Фурье-анализа конечной аperiodической функции  $f(x)$  соответствует бесконечный спектр и наоборот, конечный спектр соответствует бесконечной исходной функции.

Поэтому для реальных сигналов условия теоремы Котельникова в строгом смысле слова не выполняются.

1. Все реальные сигналы ограничены во времени и имеют неограниченный спектр, т.е.  $f_s = \infty$ .
2. В соответствии с рядом Котельникова восстановление осуществляется по бесконечному числу отсчетов ( $-\infty \leq k \leq \infty$ ).
3. Поскольку сигнал восстанавливается по бесконечному числу отсчетов функций, то его восстановление осуществляется с бесконечной задержкой во времени.

Несмотря на эти недостатки, теорема Котельникова имеет фундаментальное значение, так как позволяет определить предельные возможности дискретной передачи сигналов.

Пример.

Передается речевой сигнал в полосе частот  $\Delta F$  (от 0 до 3000 Гц). Время передачи  $\Delta t = 10$  сек. Каждый дискретный отсчет кодируется 5 двоичными разрядами.

Определить минимальный объем памяти, требуемый для хранения информации  $W_{\text{зу}}$ .

$$T = 1/2\Delta F = 1/6000 = 0,00016 \text{ с.},$$

следовательно, число отсчетов на интервале  $\Delta t$ :

$$N = \Delta t/T = 10/0,00016 = 60000.$$

$$\text{Объем ЗУ: } W_{\text{зу}} = 60000 \times 5 = 300000 \text{ бит.}$$

На практике теорему Котельникова для выбора шага дискретизации применяют следующим образом:

1. Определяют эффективную ширину спектра  $f_s$ .
2. Вычисляют шаг дискретизации  $T = 1/2f_s$ .
3. На приемной стороне восстанавливается сигнал по следующей формуле

$$x(t) \approx \sum_{k=1}^{\beta} x(kT) \frac{\sin(2\pi f_s(t - kT))}{2\pi f_s(t - kT)}, \quad (4.4)$$

где  $T$  - длительность сигнала;  $\Delta t$  - шаг дискретизации;  $\beta = \Delta t/T$  - база сигнала.

Итак, в результате дискретизации в соответствии с теоремой Котельникова от  $x(t)$  мы переходим к набору отсчетов или к вектору:

$$X = \{x_n\}; \quad X = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_{N-1}]. \quad n=0, N-1.$$

### 4.1.3. Дискретизация при использовании квадратурных сигналов

Пусть сигнал имеет вид:

$$S(t) = A(t) \cos(\omega_0 t + \varphi(t)),$$

где  $\omega_0$  – несущая частота,  $A(t)$  и  $\varphi(t)$  – законы амплитудной и фазовой аяции, которые задают медленно (по сравнению с  $\cos(\omega_0 t)$ ) меняющиеся функции.

Если непосредственно из теоремы Котельникова выбирать частоту дискретизации, то она должна быть не меньше, чем  $\omega_0 + \Delta\omega$ , где  $\Delta\omega$  – ширина спектра сигнала  $S(t)$ .

Однако такой сигнал можно представить, перейдя к комплексному виду, как:

$$\dot{S}(t) = A(t) \cdot e^{j\varphi(t)} \cdot e^{j\omega_0 t}$$

Информация, переносимая сигналом, описывается как:

$$\dot{A}(t) = A(t) \cdot e^{-j\varphi(t)}$$

Умножим  $S(t)$  на  $\cos(\omega_0 t)$  в одном канале и на  $\sin(\omega_0 t)$  – в другом (рис.4.3):

$$\begin{aligned} S(t) \cos(\omega_0 t) &= A(t) \cos(\omega_0 t + \varphi(t)) \cdot \cos \omega_0 t = \\ &= \frac{1}{2} A(t) \cos \varphi(t) + \frac{1}{2} A(t) \cos(2\omega_0 t + \varphi(t)) \end{aligned}$$

$\underbrace{\hspace{10em}}_{\text{Re}[\dot{A}(t)]}$

$$S(t) \sin(\omega_0 t) = -\frac{1}{2} \text{Im}(\dot{A}(t)) - \frac{1}{2} A(t) \cos(2\omega_0 t - \varphi(t))$$

В результате получаем частоту дискретизации  $\Delta\omega$ . Такой принцип дискретизации называют *гетеродинированием* сигнала [19].

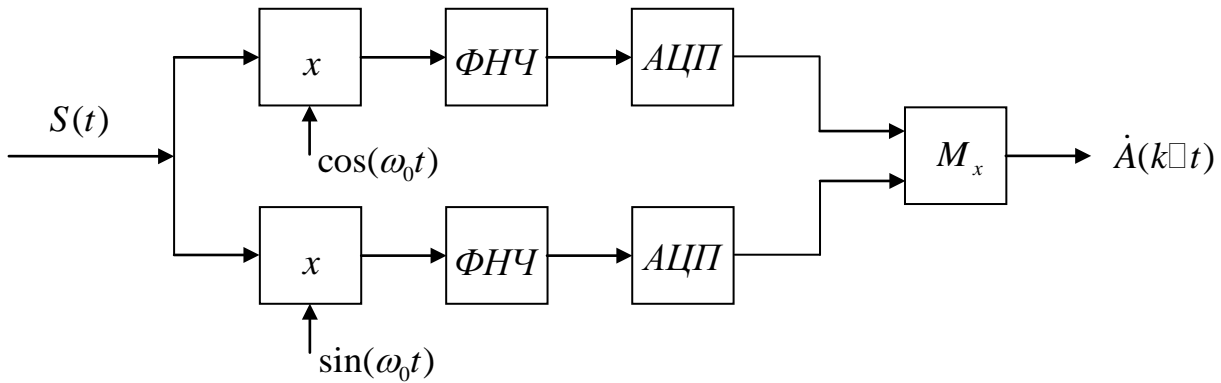


Рис.4.3. Принцип дискретизации с гетеродинированием сигнала

#### 4.1.4. Определение шага временной дискретизации при восстановлении сигнала полиномами 0-го порядка

Восстановление по непрерывного сигнала по теореме Котельникова связано с задержкой сигнала, что неприемлемо в системах работающих в реальном масштабе времени.

Для восстановления сигнала в реальном масштабе времени необходимы способы восстановления с экстраполяцией (предсказанием) сигнала. При этом могут использоваться полиномы различной степени. Простейший полином - 0-й степени. Рассмотрим этот случай (рис. 4.4.).

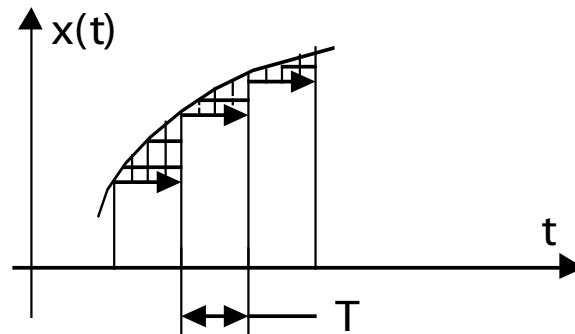
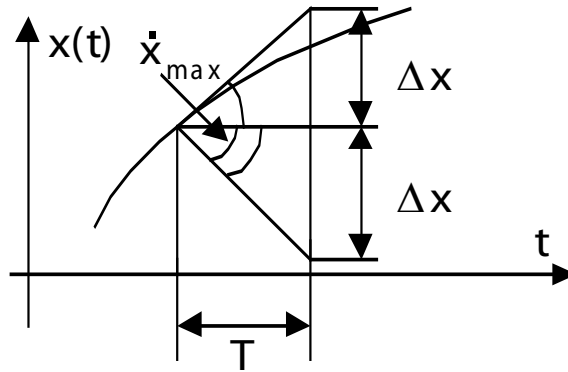


Рис. 4.4.

Требуется определить интервал времени  $T$ , при котором погрешность восстановления не превышает допустимых пределов, т.е.  $\sigma_B \leq (\sigma_B)_{\text{допуст.}}$ .

Для этого необходимо знать:

1. Наибольшее значение сигнала по модулю -  $|x|_{\text{max}}$ .
2. Эффективную ширину спектра сигнала  $f_3$ .



$\Delta x = \left| \dot{x} \right|_{\max} T$ , где  $\left| \dot{x} \right|_{\max}$  - модуль максимальной 1-й производной от сигнала.

Для ее оценки нужно оценивать  $f_3$ . Это можно сделать с помощью неравенства Бернштейна:

$$x(t) = \sin 2\pi f_3 t;$$

$$\left| \dot{x} \right| \leq \left| \dot{x} \right|_{\max} 2\pi f_3 \cos 2\pi f_3 t.$$

При  $t=0$  наихудшее значение  $\left| \dot{x} \right|$  составляет  $\left| \dot{x} \right|_{\max} = \left| \dot{x} \right|_{\max} 2\pi f_3$ .

С учетом неравенства Бернштейна можно получить

$$\Delta x = \left| \dot{x} \right|_{\max} 2\pi f_3 T;$$

Откуда искомое значение временного интервала:

$$T = \frac{\Delta x}{2 \left| \dot{x} \right|_{\max} \pi f_3}. \tag{4.5}$$

Надо на основании  $\Delta x$  определить  $\sigma_B$ .

$$\sigma_B = \sqrt{D_B};$$

$$D_B = \int_{-\infty}^{\infty} \omega_B(x) x^2 dx,$$

где  $D_B$  - дисперсия;  $\omega_B(x)$  - закон распределения плотности вероятности погрешности дискретизации;  $x$  - параметр интегрирования. Рассмотрим равномерный закон распределения. В этом случае (см. раздел

$$D_B = \int_{-\Delta x}^{\Delta x} \frac{1}{2\Delta x} x^2 dx = \frac{(\Delta x)^2}{3};$$

$$\sigma_B = \frac{\Delta x}{\sqrt{3}}. \tag{4.6}$$

Подставляя (4.6) в (4.5) получим

$$T = \frac{\sqrt{3}\sigma_B}{2|x|_{\max} \pi f_9}$$

### Методика выбора шага дискретизации T.

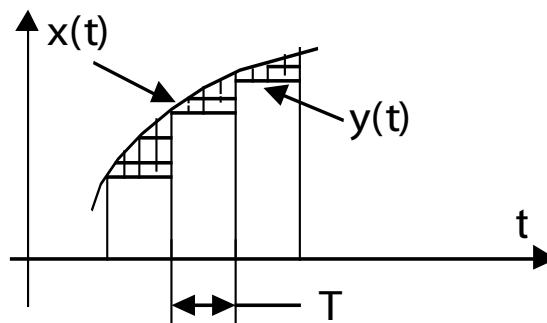
1. По заданному значению  $|x|_{\max}$  и  $f_9$  при помощи неравенства Бернштейна определяем  $\sigma_B$ .

2. По заданной величине  $(\sigma_B)_{\text{допуст}}$  определяем наибольшее отклонение сигнала за время T  $\Delta x = \sigma_B \sqrt{3}$ .

3. Определяем требуемый шаг  $T = \frac{\Delta x}{|x|_{\max}}$ .

### 4.1.5. Определение шага дискретизации при заданной автокорреляционной функции

Рассмотрим рисунок



Из рисунка видно, что ошибка  $e(t) = y(t) - x(t) = x(t-T) - x(t) = x(t^*) - x(t^* - T) = x(t) - x(t+T)$ . Введено обозначение  $t^* = t - T$ .

Дисперсия этой ошибки

$$\begin{aligned} D_e &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [e(t)]^2 dt = \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [x(t) - x(t+T)]^2 dt = \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [x(t)]^2 dt + \\ &+ \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [x(t+T)]^2 dt - \\ &- 2 \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [x(t)x(t+T)] dt. \end{aligned}$$

(4.7)

В выражении (4.7) первое слагаемое -  $Dx=Kx(0)$ , второе слагаемое -  $Dx=Kx(0)$ , третье слагаемое -  $2Kx(T)$ .

Пусть  $x(t)$  - стационарный сигнал. Он однороден во времени и его характеристики от него не зависят. Тогда

$$D_e = 2Kx(0) - 2Kx(T).$$

$$\sigma_\ell = \sqrt{D_e} = \sqrt{2(Kx(0) - Kx(T))} \quad (4.8)$$

Методика определения шага дискретизации состоит в разрешении уравнения (4.8) относительно  $T$ .

АКФ  $Kx(\tau)$  является более содержательной характеристикой, чем эффективная полоса частот. Поэтому, определение шага дискретизации по уравнению (4.8) является более точным, чем по неравенству Бернштейна.

**Изменение частоты дискретизации.** При решении различных задач обработки сигналов достаточно часто требуется изменение частоты дискретизации сигнала.

**Пример** – преобразование аудиозаписи из формата записи компакт-диска ( $f_{\text{дискр}} = 44,1 \text{ кГц}$ ) в формат цифровой магнитной записи R-DAT ( $f_{\text{дискр}} = 48 \text{ кГц}$ ) и обратно.

Возможны следующие варианты:

1. Интерполяция – повышение частоты дискретизации в целое число раз (interpolation).
2. Прореживание или сэмплирование (decimation) – понижение частоты дискретизации в целое число раз.
3. Передискретизация (resampling) – изменение частоты дискретизации в произвольное (в том числе дробное) число раз.

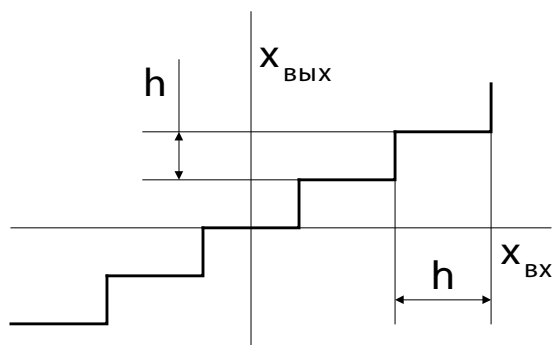
## 4.2. Квантование непрерывных сигналов по уровню

Квантование необходимо для того, чтобы от непрерывного сигнала перейти к цифровому. Процесс квантования сводится к округлению бесконечного множества значений сигнала до ближайших оцифрованных значений, называемых уровнями квантования.

В отличие от временной дискретизации, когда в соответствии с теоремой Котельникова погрешность дискретизации может быть  $\geq$ , квантование по уровню связано с возникновением неустраняемой погрешности квантования.

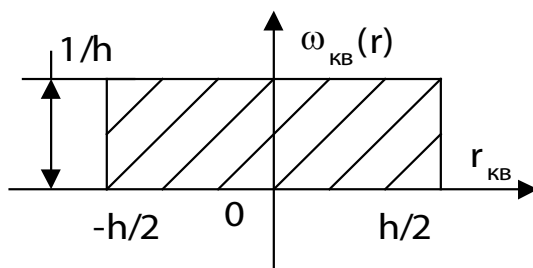
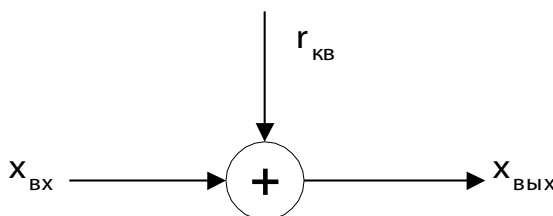
Рассмотрим рисунок





Доказано, что погрешность квантования представляет собой случайную величину, равномерно распределенную в пределах шага  $h$  и аддитивную по отношению к квантованному сигналу.

Поэтому, вместо квантователя можно использовать эквивалентную схему



Математическое ожидание  $m_r = 0$ . Дисперсия погрешности квантования  $D_r = h^2/12$ . СКО погрешности  $\sigma_r = h/2\sqrt{3}$ .

### Методика определения параметров квантования исходя из заданной

$\sigma_r = (\sigma_r)_{\text{доп}}$

Пусть известен диапазон амплитудных значений сигнала  $|x|_{\text{max}}$ .

Необходимо найти число уровней квантования  $K$  и число двоичных разрядов  $N$ .

1. По заданной погрешности квантования определяем шаг

$$h = 2\sigma_r\sqrt{3}.$$

2. Далее определяем число уровней квантования  $K = \lceil |x|_{\text{max}}/h \rceil$ .

3. И, наконец, находим число двоичных разрядов  $N = \lceil \lg K \rceil$ .

Вносимая в результате аналого-цифрового преобразования среднее квадратичное отклонение ошибки составит

$$\sigma = \sqrt{\sigma_{\epsilon}^2 + \sigma_r^2}.$$

Равномерное квантование гарантирует, что динамический диапазон (или размах) шума квантования не превосходит величины шага квантования.

Однако для минимизации среднеквадратичного шума квантование необходимо, чтобы набор уровней квантования зависел от статистических свойств сигнала, а именно, от плотности вероятности значений сигнала в фиксированные моменты времени.

Иначе говоря, уровни квантования должны располагаться плотнее друг к другу в тех областях, где сигнал принимает значения с наибольшей вероятностью. Тогда весь динамический диапазон сигнала  $D$  разбивается на  $N$  зон квантования  $a_0, a_1, \dots, a_N$ .

Если значение входного сигнала попадает в диапазон  $[a_{k-1}, a_k]$ , то ему соответствует значение  $b_k$ .

Пусть сигнал имеет плотность вероятности  $P(x)$  и мы хотим минимизировать дисперсию шума квантования.

Среднее значение ошибки квантования  $\bar{e}$ :

$$\bar{e} = \sum_{k=1}^N \int_{a_{k-1}}^{a_k} (x - b_k) P(x) dx = \bar{x} - \sum_{k=1}^N b_k \int_{a_{k-1}}^{a_k} P(x) dx = \bar{x} - \sum_{k=1}^N b_k P_k,$$

где  $\bar{x}$  – математическое ожидание сигнала  $x$ , а  $P_k$  – вероятность попадания значения сигнала в  $k$ -й диапазон квантования.

$$\bar{e}^2 = \sum_{k=1}^N \int_{a_{k-1}}^{a_k} (x - b_k)^2 P(x) dx = \bar{x}^2 - 2 \sum_{k=1}^N b_k \int_{a_{k-1}}^{a_k} x P(x) dx + \sum_{k=1}^N b_k^2 \int_{a_{k-1}}^{a_k} P(x) dx$$

$\bar{e}^2$  – средний квадрат ошибки.

Для минимизации  $\bar{e}^2$  необходимо, чтобы частное произведение этого выражения по  $a_k$  и  $b_k$  были бы равны 0.

В результате можно получить выражения для определения оптимальных значений параметров

$$b_k = \frac{\int_{a_{k-1}}^{a_k} x P(x) dx}{\int_{a_{k-1}}^{a_k} P(x) dx}; \quad a_k = \frac{b_{k-1} + b_k}{2}$$

Это же выражение обеспечивает и нулевое среднее значение шума квантования.

## 5. ОСНОВНЫЕ ТИПЫ ДИСКРЕТНЫХ АЛГОРИТМОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Математические методы обработки сигналов можно подразделить на три группы, если в основу классификации положить принцип формирования отдельного элемента (отсчета) результата по некоторой совокупности элементов (отсчетов) исходного сигнала [16].

**Точечные преобразования** – в таких преобразованиях обработка каждого элемента исходных данных производится независимо от соседнего. Иначе говоря, значение каждого отсчета результата определяется как функция от одного отсчета исходного сигнала, причем номера отсчетов сигнала и результата одинаковы.

Иначе говоря, пусть требуется обработать вектор из  $n$  отсчетов сигнала:

$$X = [x_0 \ x_1 \ x_2 \ \dots \ x_n] \quad (5.2)$$

и получить последовательность чисел:

$$Y = [y_0 \ y_1 \ y_2 \ \dots \ y_n], \quad (5.3)$$

причем

$$y_i = f(x_i) \quad (5.4)$$

Точечные преобразования достаточно просты и наименее громоздки с точки зрения вычислительных затрат. Если обрабатывается матрица размером  $N \times N$  элементов отсчетов исходного двумерного сигнала, то вычислительная сложность процедуры точечных преобразований составит

$$Q_m = N^2 \text{ (БО)},$$

где под базовой операцией (БО) понимается операция вида (5.4).

**Локальные преобразования** - при локальных преобразованиях обеспечивается формирование каждого элемента матрицы или вектора результата как функции от некоторого множества соседних элементов матрицы или вектора отсчетов исходного сигнала, составляющих некоторую локальную окрестность. При этом полагается, что местоположение вычисляемого отсчета результата (или текущий индекс элемента) задается координатами (или текущими индексами) центрального элемента локальной окрестности. Для формирования следующего элемента матрицы результата выполняется смещение окрестности вдоль строки матрицы исходных данных или вдоль исходного вектора. Такая перемещаемая окрестность часто носит название окна сканирования. При обработке матрицы исходных данных после прохождения всей строки матрицы исходных данных окно сканирования смещается на одну строку и возвращается в начало следующей строки, после чего продолжается обработка. Просматриваемая при перемещении окна сканирования полоса строк матрицы носит название полосы сканирования. Иначе говоря, при такой обработке

$$y_i = F(\hat{X}); \hat{X} = \{x_{i-m/2}, x_{i-m/2+1}, \dots, x_i, \dots, x_{i+m/2-1}, x_{i+m/2}\}, \quad (5.5)$$

где  $i = 0, N-1$  - индекс отсчета результата,  $m$  - размер окна сканирования. Если  $i < m/2$  или  $i > N-m/2$ , что имеет место на практике при обработке начальных и конечных отсчетов вектора исходного сигнала, то элементы вектора исходных данных с "недостающими" индексами полагаются равными нулю.

Вычислительная сложность локального преобразования составляет

$$Q_1 = N^2 m^2 \text{ (БО)},$$

где под базовой операцией понимается выполнение заданного преобразования для отдельного отсчета исходных сигналов. Примером локальных преобразований могут служить аperiodическая свертка или корреляция, а также процедуры ранговой фильтрации.

**Глобальное преобразование** предусматривает формирование каждого отсчета результата как функции от всей совокупности отсчетов исходного сигнала и некоторого множества меняющихся от одного отсчета результата к другому по определенному правилу коэффициентов, составляющих так называемое ядро преобразования. В случае обработки одномерного исходного сигнала глобальное преобразование можно определить как

$$Y_i = F(G_i, X); \quad X = [x_0 \ x_1 \ x_2 \ \dots \ x_n], \quad i = 0, N-1 \quad (5.6)$$

где  $G_i$  – изменяемое ядро преобразования. Вычислительная сложность глобального преобразования в общем случае для случая обработки двумерного сигнала составляет

$$Q_2 = N^4 \text{ (БО)},$$

где под базовой операцией понимается выполнение заданного преобразования вида (1.6) для отдельного элемента исходных данных. Примером подобных преобразований могут служить дискретные ортогональные преобразования типа преобразования Фурье, Хартли, Адамара.

### 5.1. Линейные и нелинейные преобразования

Все преобразования ЦОС могут быть подразделены по своему типу на линейные и нелинейные преобразования [5,16,21].

Пусть  $x(t)$  – входная последовательность, а  $y(t_1)$  – выходная последовательность, связанная со входной через некоторое функциональное преобразование  $T$

$$y(t_1) = T[x(t)] \quad (5.7)$$

Для линейных преобразований справедлив аддитивный закон :

$$T[ax_1(t) + bx_2(t)] = aT[x_1(t)] + bT[x_2(t)] = ay_1(t_1) + by_2(t_1), \quad (5.8)$$

где  $a$  и  $b$  – некоторые константы. Таким образом, линейное преобразование, применяемое к суперпозиции исходных сигналов эквивалентно по своему воздействию суперпозиции результатов преобразования каждого из сигналов. Свойство линейности является весьма важным для практических приложений, поскольку позволяет значительно упростить обработку различных сложных сигналов, являющихся суперпозицией некоторых элементарных сигналов. Так, в частности, за простейший элементарный сигнал может быть принят моногармонический сигнал  $x(t)$ , описываемый функцией:

$$x(t) = a \cos(2\pi vt - \varphi),$$

где  $a$  - амплитуда,  $\nu = \frac{1}{T}$  - частота,  $T$  - период,  $\varphi$  - начальная фаза.

Тогда более сложный полигармонический сигнал может быть записан как суперпозиция простейших моногармонических сигналов:

$$\hat{x}(t) = \sum_{k=0}^K a_k \cos(2\pi\nu_k t - \varphi_k)$$

Важное место в цифровой обработке сигналов имеет некоторый идеализированный простейший импульсный сигнал, называемый дельта-функцией или единичным импульсом:

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

Согласно теории цифровой обработки сигналов, любой сигнал может быть представлен как суперпозиция взвешенных единичных импульсов следующим образом:

$$x(t) = \sum_{k=-\infty}^{\infty} x(t_k) \delta(t - t_k) \quad (5.9)$$

где  $x(t)$  – отсчет сигнала в некоторый момент времени.

Если на вход системы ЦОС, выполняющей линейное преобразование, поступает единичный импульс, то сигнал  $h(t)$ , снимаемый с выхода системы и являющийся откликом системы на единичный импульс, носит название импульсной характеристики (импульсного отклика) системы. Импульсный отклик является важнейшей характеристикой системы и позволяет описать ее как “черный ящик”, задав реакцию системы на некоторый простейший эталонный сигнал.

Если  $h(t)$  конечна, то такие системы называются КИХ-системами, т.е. системами с конечной импульсной характеристикой. Если  $h(t)$  бесконечна, то это БИХ-системы, т.е. системы с бесконечной импульсной характеристикой. В цифровой обработке сигналов имеет смысл рассматривать только КИХ-системы, поскольку время обработки, т.е. реакции системы на входной сигнал должно быть конечно.

Подставив (5.9) в (5.8), получаем для линейных преобразований:

$$\mathcal{X}(t_1) = \mathcal{T}[x(t)] = \mathcal{T}\left[\sum_{k=-\infty}^{\infty} x(t_k) \delta(t - t_k)\right] = \sum_{k=-\infty}^{\infty} x(t_k) \mathcal{T}[\delta(t - t_k)] = \sum_{k=-\infty}^{\infty} x(t_k) h_k(t) \quad (5.10)$$

Таким образом, для линейной системы результат обработки любого поступившего на вход сложного сигнала может быть определен как суперпозиция импульсных откликов системы на поступившие на вход единичные импульсы с соответствующей начальной задержкой и весом, определяемым весом соответствующего отсчета исходного сигнала.

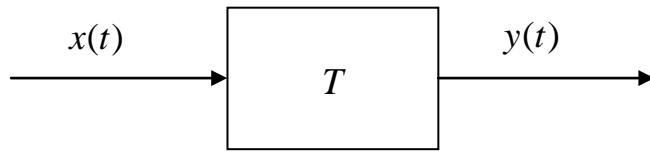
Примерами линейных преобразований могут служить преобразования Фурье, Хартли, свертка и корреляция. К нелинейным преобразованиям относятся, в частности, многие алгоритмы распознавания, гистограммные преобразования и ранговая фильтрация.

## 5.2. Характеристики линейных систем

## Импульсная характеристика

Пусть сигнал на выходе системы

$$y(t) = T[x(t)]$$



Если  $x(t) = \delta(t)$ , то

$$y(t) = T[\delta(t)] = h(t)$$

носит название импульсной характеристики или импульсного отклика системы.

$$y(t) = \int_{-\infty}^{\infty} x(t_1)h(t - t_1)dt_1.$$

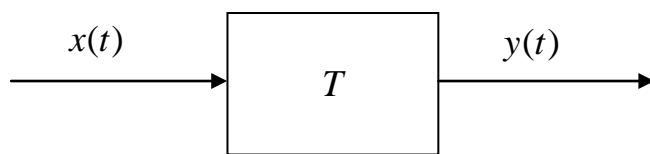
## Переходная характеристика

Реакция системы на поданную на вход функцию единичного скачка, обозначается как  $g(t)$ . Так как  $\delta$ -функция является производной по  $t$  от функции единичного скачка, то:

$$h(t) = \frac{dg(t)}{dt} \text{ или}$$

$$g(t) = \int_{-\infty}^t h(t_1)dt_1$$

**Комплексный коэффициент передачи (передаточная функция) системы:**



$$S_{\text{вых}}(\omega) = S_{\text{вх}}(\omega) \cdot K(\omega)$$

$$K(\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t} dt \text{ – комплексный коэффициент передачи или передаточная}$$

функция;

модуль  $K(\omega)$  – АЧХ системы;

фаза  $K(\omega)$  – ФЧХ системы;

**Коэффициент передачи по мощности:**

$$K_M(\omega) = |K(\omega)|^2 = K(\omega) \cdot K^*(\omega)$$

## Взаимный спектр входного и выходного сигналов

$$S_{\text{вых-вх}}(\omega) = S_{\text{вых}}(\omega) \cdot S_{\text{вх}}^*(\omega) = S_{\text{вх}}(\omega) \cdot K(\omega) \cdot S_{\text{вх}}^*(\omega) = |S_{\text{вх}}(\omega)| \cdot K(\omega)$$

## Взаимная корреляция между входом и выходом

$$B_{\text{вых-вх}}(\tau) = \int_{-\infty}^{\infty} B_{\text{вх}}(\tau_1) h(\tau - \tau_1) d\tau_1$$

## Преобразование случайных сигналов в линейной системе Спектральная плотность мощности

$$W_{\text{вых}}(\omega) = W_{\text{вх}}(\omega) \cdot K_M(\omega) = W_{\text{вх}}(\omega) \cdot |K(\omega)|^2$$

## Корреляционная функция

$$R_{\text{вых}}(\tau) = \int_{-\infty}^{\infty} R_{\text{вх}}(\tau_1) B_h(\tau - \tau_1) d\tau_1$$

$B_h(\tau)$  – есть результат обратного ПФ (Преобразования Фурье) от коэффициента передачи по мощности  $K_M(\omega)$ .  $B_h(\tau)$  определяет корреляционную функцию импульсного отклика системы, т.е.:

$$B_h(\tau) = \int_{-\infty}^{\infty} h(t) h(t - \tau) dt$$

## Дисперсия на выходе:

$$D_{\text{вых}} = R_{\text{вых}}(0) = \int_{-\infty}^{\infty} R_{\text{вх}}(\tau_1) B_h(\tau_1) d\tau_1$$

$$\text{или } D_{\text{вых}} = \frac{1}{2\pi} \int_{-\infty}^{\infty} W_{\text{вх}}(\omega) K_M(\omega) d\omega$$

## 5.3. Циклическая свертка и корреляция

В дискретном виде линейные преобразования могут быть описаны в общем виде как векторно-матричные операции [5, 16,21]:

$$Y = B_N X \quad (5.11)$$

где  $X$  - вектор отсчетов исходных данных, полученный в результате дискретизации непрерывного сигнала согласно теореме Котельникова,  $Y$  - вектор отсчетов результата,  $B_N$  - матрица размера  $N \times N$ , определяющая ядро выполняемого преобразования.

К числу подобных преобразований относится циклическая свертка последовательностей  $X = [x_0 \ x_1 \ x_2 \ \dots \ x_{N-1}]$  и  $G = [g_0 \ g_1 \ g_2 \ \dots \ g_{N-1}]$ , в этом случае строится матрица ядра свертки:

$$B_N^C = G_N = \begin{vmatrix} g_0 & g_{N-1} & g_{N-2} & \dots & g_1 \\ g_1 & g_0 & g_{N-1} & \dots & g_2 \\ g_2 & g_1 & g_0 & \dots & g_3 \\ \dots & \dots & \dots & \dots & \dots \\ g_{N-1} & g_{N-2} & g_{N-3} & \dots & g_0 \end{vmatrix}$$

Каждый элемент вектора  $Y$  может быть описан как:

$$y_m = \sum_{n=0}^{N-1} g_{m-n} x_n ; \quad m = \overline{0, N-1} \quad . \quad (5.12)$$

Матрица ядра циклической взаимокорреляции может быть построена как транспонированная матрица ядра свертки, т.е. следующим образом:

$$K_N = B_N^K = \begin{vmatrix} g_0 & g_1 & g_2 & \dots & g_{N-1} \\ g_{N-1} & g_0 & g_{N-1} & \dots & g_{N-2} \\ g_{N-2} & g_{N-1} & g_0 & \dots & g_{N-3} \\ \dots & \dots & \dots & \dots & \dots \\ g_1 & g_2 & g_3 & \dots & g_0 \end{vmatrix}$$

Поэтому каждый отсчет результата может быть записан как:

$$z_m = \sum_{n=0}^{N-1} g_{m+n} x_n ; \quad m = \overline{0, N-1} \quad (5.13)$$

причем  $g_{m+n} = g_n$  для  $m+n \geq N$ .

#### 5.4. Аперриодическая свертка и корреляция

Аперриодическая свертка и корреляция в отличие от циклической относятся к классу локальных преобразований. При этом как правило полагается, что размер вектора исходных данных значительно больше размера ядра свертки, что приводит к следующему выражению для вычисления любого отсчета результата:

$$y_i = \sum_m g_m x_{i-m} \quad (5.14)$$

Вычисление свертки и корреляции лежит в основе корреляционного метода подавления помех.

Сущность такого метода заключается в использовании различия между корреляционными функциями сигнала и помехи. Данный метод эффективен лишь в случае обработки периодических или квазипериодических сигналов.

Рассмотрим сущность метода на примере, когда полезный сигнал является гармоническим, а помеха - типа белого гауссова шума [21].

Автокорреляционная функция сигнала является тоже гармонической и имеет ту же частоту. Метод автокорреляционного приема основан на анализе автокорреляционной функции принятого сигнала  $y(t)=x(t)+p(t)$ .



Если сигнал и помеха взаимно независимы (типичный для практики случай), то

$$K_y(\tau) = K_x(\tau) + K_r(\tau),$$

т.е. автокорреляционная функция принятого сигнала равна сумме автокорреляционных функций сигнала и помехи.

Метод корреляционного приема позволяет обнаружить полезный сигнал, который имеет мощность значительно меньшую, чем мощность помехи.

### 5.5. Двумерная аperiodическая свертка и корреляция

Важное место среди операций линейной обработки сигналов занимает операция перемножения матриц одинаковой размерности. Такая операция имеет вид:

$$Z_N = B_N X_N, \quad (5.15)$$

где  $B_N$  и  $X_N$  исходные матрицы порядка  $N$ , а  $Z_N$  - результирующая матрица того же порядка. Каждый элемент матрицы  $Z_N$  формируется в соответствии с выражением:

$$z_{ij} = \sum_k x_{ik} b_{kj} \quad , \quad k, i, j = 1, N$$

где  $x_{ik}$ ,  $b_{kj}$  - элементы исходных матриц.

На основе операций (5.15) выполняется вычисление функций двумерной аperiodической свертки/корреляции исходного двумерного сигнала (изображения) с двумерным ядром. Подобное преобразование часто используется как для удаления шумов, так и для выделения мелких объектов.

Математически двумерная аperiodическая свертка может быть описана следующим образом:

$$y_{ij} = \sum_{n=1}^M \sum_{m=1}^M g_{mn} x_{i+m-1, j+n-1}$$

где  $y_{ij}$  - отсчеты результатов вычислений (отсчеты свертки),  $g_{mn}$  - отсчеты весовой функции окна (ядра свертки) размерностью  $M \times M$  отсчетов, причем  $N \gg M$ . Очевидно, что размерность матрицы, описывающей двумерную свертку, равна  $(N + M - 1) \times (N + M - 1)$  отсчетов. Поэтому матрица исходных данных также должна быть дополнена до указанного размера нулевыми элементами по краям кадра.

Отсчеты свертки формируются при перемещении окна вдоль строки исходного изображения. Для каждого положения окна формируется один отсчет свертки, после чего окно сдвигается на один элемент вдоль строки (т.е. на один столбец). Обработка начинается с элемента  $x$  исходного изображения.

После прохождения  $i$ -й строки изображения ( $i = 1, N$ ) окно смещается на одну строку вниз и возвращается к началу следующей ( $i + 1$ )-й строки изображения. По окончании обработки кадра изображения окно перемещается в исходное положение.

При вычислении свертки можно распараллелить вычисления по столбцам окна сканирования, выполняя параллельно вычисление произведений столбцов ядра свертки  $\hat{g}_n$  и столбцов  $X_{j+n-1}^\wedge$  текущей полосы сканирования изображения. Поэтому для окончательного вычисления отсчета свертки достаточно сформировать сумму:

$$Y_j = \sum_n \hat{g}_n X_{j+n-1}^\wedge \quad (5.16)$$

соответствующую отсчету свертки с номером  $j$ , расположенному в средней строке полосы шириной  $M$ .

Представляет интерес распараллеливание по разрядным срезам, поскольку в этом случае практически исключается операция умножения [16,21]. При разрядно-срезовой обработке данные должны быть представлены в формате с фиксированной точкой. Представим значение элемента изображения в следующем виде :

$$x_{mn} = \sum_q b_{mn}^q 2^{q-1}, \quad (5.17)$$

где  $b_{mn}^q$  -  $q$ -ый разряд  $x_{mn}$  ( $q = 1, \dots, Q$ ), ( $g = (1, Q)$ ) где  $Q$  - разрядность данных.

С учетом этого процедура вычисления свертки принимает вид:

$$y_{ij} = \sum_q 2^{q-1} \sum_m \sum_n g_{mn} b_{i+m-1, j+n-1}^q = \sum_q 2^{q-1} Y_q \quad (5.18)$$

Таким образом, процедура двумерной апериодической свертки для одного положения окна сводится к  $M \times N \times Q$  операциям сложения и  $(Q-1)$  операциям сдвига. Умножение под знаком суммы сводится к операции вида :

$$g_{mn} b_{mn}^q = g_{mn} \text{ при } b_{mn}^q = 1 \text{ и } g_{mn} b_{mn}^q = 0 \text{ при } b_{mn}^q = 0.$$

Поэтому разрядно-срезовой алгоритм вычисления свертки или корреляции для одного положения окна может быть представлен в виде [16]:

*начало;*

$S := 0;$

для  $q = 1, \dots, Q$  цикл:

$S1 := 0;$

для  $m = 1, \dots, M$  цикл:

для  $n = 1, \dots, M$  цикл:

если  $b_{i-m, j-n}^q = 1$ , то  $g_{mn} := g_{mn}$

иначе  $g_{mn} := 0;$

$S1 := S1 + g_{mn};$

конец цикла по  $n$ ;

конец цикла по  $m$ ;

$$S = S + sM_{q-1}(S1);$$

конец цикла по  $q$ ;

$$y_{ij} = S;$$

конец.

Физический смысл функций свертки и корреляции состоит в том, что они являются количественной мерой совпадения (сходства) двух последовательностей  $f(x)$  и  $g(x)$ . При этом наиболее полно мера сходства может быть определена по функции корреляции, в связи с чем функция взаимокорреляции (или кросскорреляции) может быть использована для распознавания сигналов. Если распознаваемый сигнал  $f(t)$  точно соответствует эталонному сигналу  $g(t)$ , то результирующий сигнал  $\omega^k(t)$  принимает значение:

$$\omega^k(t) = \max \text{ при } f(t) \equiv g(t),$$

что соответствует функции автокорреляции. Если сигналы отличаются, то  $\omega^k(t) < \max$ .

Кроме того, при обработке двумерных сигналов (изображений объектов) координаты максимума данной функции определяют центр тяжести исходного распознаваемого объекта, что позволяет определить и его местоположение (т.е. запеленговать объект). По этим причинам вычисление одномерной или двумерной корреляции лежит в основе целого ряда методов распознавания.

Таким образом, функции линейной апериодической свертки и корреляции полезны для распознавания сигналов заданной формы. На этом принципе работают корреляционные методы распознавания. Свертка определяется путем скольжения эталона по вектору исходного сигнала, и максимум функции будет тогда, когда исходный сигнал совпал с эталоном. Функция апериодической свертки, кроме того, оказывается полезной для удаления, например, низкочастотных помех.

## 5.6 Нерекурсивные и рекурсивные фильтры

Одной из областей ЦОС, в основе выполнения процедур которой лежит вычисление выражений типа свертки, является цифровая линейная фильтрация

Цифровые фильтры по принципу выполнения фильтрации могут быть разделены на два класса - рекурсивные и не рекурсивные фильтры [17,19, 21].

Математическое функционирование нерекурсивного фильтра описывается уравнением вида :

$$y_n = \sum_{k=0}^{N-1} h_k x_{n-k} \quad (5.19)$$

Такой фильтр может рассматриваться как устройство без обратной связи, имеющее структуру, показанную на рис.5.1 :

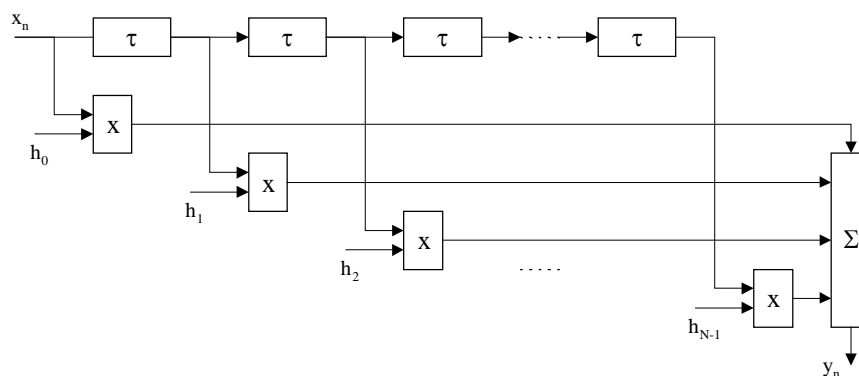


Рис. 5.1. Нерекурсивный линейный фильтр

Работа же рекурсивного фильтра описывается выражением :

$$y_n = - \sum_{j=1}^{N-1} b_j y_{n-j} - \sum_{k=0}^{N-1} h_k x_{n-k}, \quad (5.20)$$

где  $y_n$  -  $n$ -й отсчет выходного сигнала;  $x_n$  -  $n$ -й отсчет входного сигнала;  $h_k$  - коэффициенты фильтра.

Нетрудно видеть, что такой рекурсивный фильтр может рассматриваться как устройство с обратной связью.

Наиболее обобщенной структурой цифрового фильтра является структура рекурсивного фильтра (рис. 5.2). Последний включает как умножители с прямой связью (веса регулируются коэффициентами  $a$ ), так и умножители с обратной связью (веса регулируются коэффициентами  $b$ ). Характеристика такого  $n$ -звенного фильтра описывается дифференциальным уравнением  $n$ -го порядка, показывающим, что значение выборки на выходе фильтра в данный момент времени определяется линейной комбинацией взвешенных выборок в данный и предыдущий моменты времени (это справедливо и для предыдущих выборок). В результате построения такой структуры получается фильтр с характеристикой полюсно-нулевого типа, где размещение полюсов определяется коэффициентами  $b$ , а размещение нулей – коэффициентами  $a$ . Число полюсов и нулей, или порядок фильтра, задается количеством элементов задержки. В продаже имеются интегральные фильтры второго порядка для скоростей передачи входных импульсов (64 килобод), совместимых со скоростями передачи в цифровых телефонных системах.

Подобная структура рекурсивного фильтра имеет теоретически бесконечную память, и, следовательно, ее можно считать структурой фильтра с бесконечной импульсной характеристикой (БИХ). Такой фильтр не обладает неограниченной устойчивостью, если на значения коэффициентов  $b$  не наложены ограничения. Однако наличие в характеристике как полюсов, так и нулей позволяет реализовать фильтр с крутым срезом характеристики в сочетании с малой шириной полосы пропускания при небольшом числе элементов задержки (т.е. фильтр малой сложности). Один из недостатков фильтра БИХ-типа – отсутствие методов управления фазовой характеристикой (групповой задержкой) фильтра. Однако основной проблемой при проектировании адаптивных фильт-

ров БИХ-типа является возможная неустойчивость фильтра из-за наличия паразитных полюсов за пределами области устойчивости.

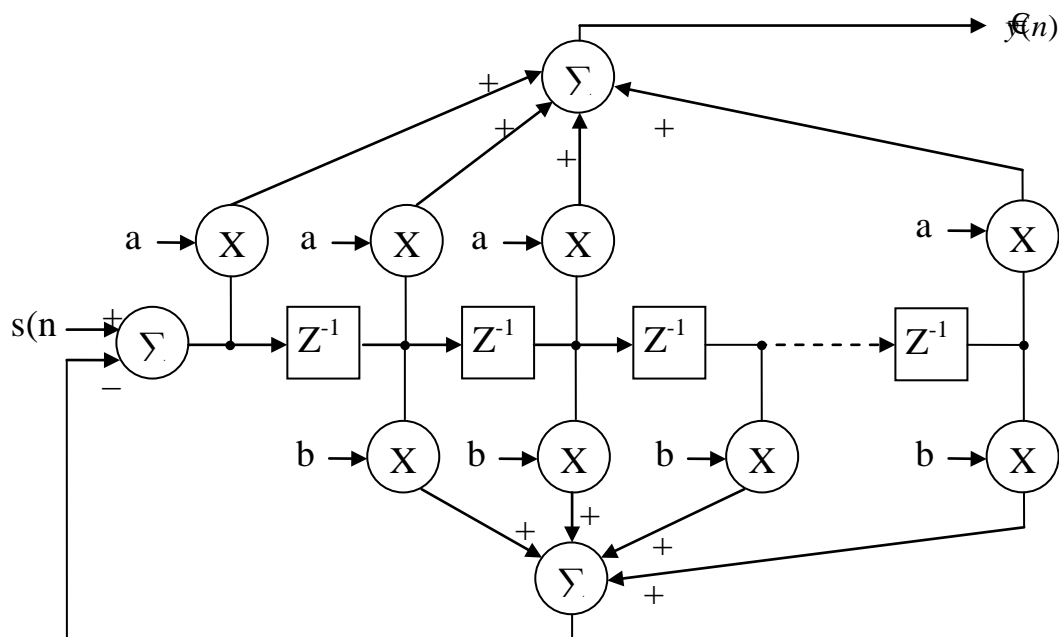


Рис. 5.2.. Структурная схема рекурсивного фильтра с бесконечной импульсной характеристикой.

Один из способов преодоления потенциальной неустойчивости фильтра является создание такого фильтра, который имеет в характеристике одни нули; в нем используются только умножители с прямой связью, и он безусловно устойчив (Рис. 5.3.). этот фильтр имеет лишь ограниченную память, регулируемую числом элементов задержки, что приводит к созданию фильтра с конечной импульсной характеристикой (КИХ-типа или трансверсального типа). Задержка входного сигнала производится с помощью некоторого числа элементов задержки. Выходы этих элементов задержки последовательно умножаются на ряд накопленных в памяти весов и полученные произведения суммируются для формирования выходного сигнала; тем самым предполагается, что выходной сигнал определяется сверткой входного сигнала с накопленными в памяти весами или значениями импульсной характеристики. Данный фильтр содержит в характеристике лишь нули (поскольку в нем нет рекурсивных элементов обратной связи), и, следовательно, для получения частотной характеристики с крутым срезом необходимо большое число элементов задержки. Однако фильтр всегда устойчив и может обеспечить линейную фазовую характеристику.

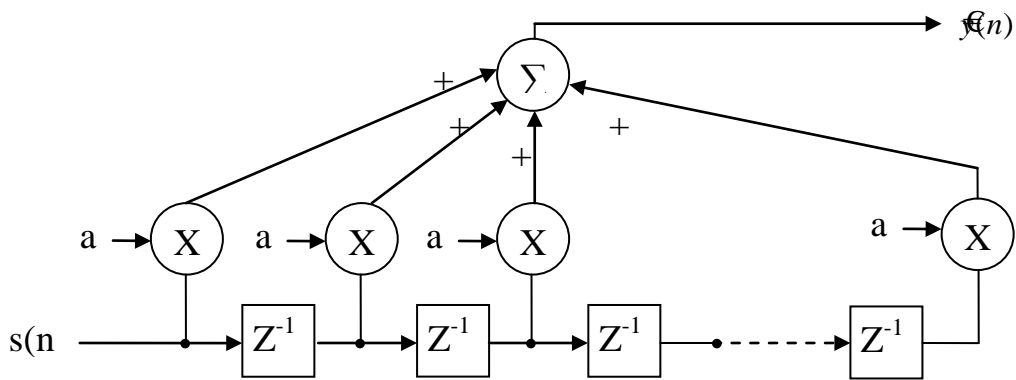


Рис. 5.3. Структурная схема нерекурсивного фильтра с конечной импульсной характеристикой

### 5.7. Метод синхронного или когерентного накопления

Метод накопления применим в том случае, если полезный сигнал в течении времени приема постоянен или является периодической функцией. Метод состоит в многократном повторении сигнала и суммировании отдельных его реализаций в устройстве обработки. Данный метод относится к группе точечных алгоритмов обработки сигналов.

Пусть полезный сигнал представлен двумя уровнями (рис.5.4).

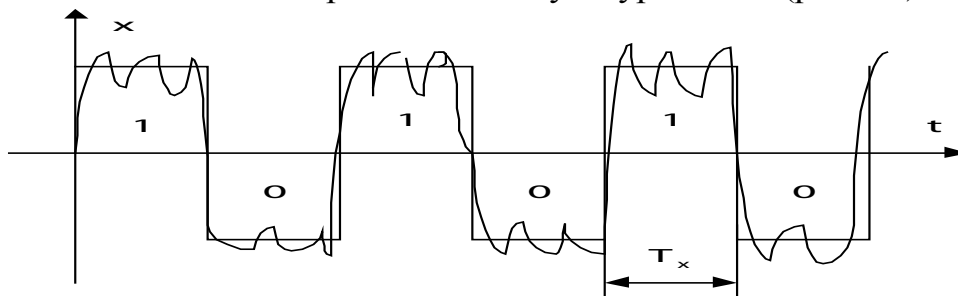


Рис.5.4. Входной двухуровневый периодический зашумленный сигнал

В интервале  $T_x$  сигнал постоянен. На интервале наблюдения  $T_x$  накапливается выборка значений принятого сигнала

$$\begin{aligned}
 y_1 &= x + n_1 \\
 y_2 &= x + n_2 \\
 &\dots\dots\dots \\
 y_m &= x + n_m
 \end{aligned}$$

и эти значения суммируются:

$$Y = \sum_{i=1}^n y_i = nx + \sum_{i=1}^n r_i.$$

Введем два допущения:

- 1) отсчеты помехи  $n_i$  не зависят друг от друга;
- 2) помеха стационарна (ее характеристики не зависят от времени)

и определим  $\left( \frac{P_x}{P_r} \right)_{\%hh}$  на выходе этого накопителя, т.е.

$$\begin{aligned} \left( \frac{P_x}{P_r} \right)_{\text{ВЫХ}} &= \frac{(nx)^2}{\left( \sum_{i=1}^n r_i \right)^2} = \\ &= \frac{(nx)^2}{nD_r} = \frac{n^2 x^2}{nP_r} = \frac{nP_x}{P_r}. \end{aligned} \quad (5.21)$$

Таким образом, при перечисленных выше условиях, в результате  $n$ -кратного отсчета, отношение мощностей сигнала и помехи увеличивается в  $n$  раз. Временной интервал между отдельными отсчетами должен быть больше интервала корреляции помехи  $\tau_0^r$ . В противном случае выигрыш за счет накопления будет меньше значения, даваемого выражением (5.21).

За счет увеличения числа отсчетов  $m$ , т.е. времени передачи  $T_x$ , можно сколь угодно увеличивать отношения сигнал/помеха.

Если сигнал представляет периодическую функцию времени, то отсчеты нужно производить через интервалы, равные или кратные периоду этой функции. В таких случаях метод носит название метода синхронного или когерентного накопления. Эффект накопления такой же, как в случае постоянного сигнала [21].

Эффект накопления можно осуществить также за счет интегрирования входного сигнала в течении времени  $T_x$ . Такой метод получил название интегрального приема.

Интегральный прием целесообразно применять в случае, когда полезный сигнал постоянен (или квазипостоянен).

### 5.8. Адаптивные фильтры.

Основным свойством адаптивной системы является изменяющееся во времени функционирование с саморегулированием. Необходимость такого функционирования очевидна из следующих рассуждений. Если разработчик проектирует «неизменяемую» систему, которую он считает оптимальной, то это означает, что разработчик предвидит все возможные условия на ее входе, по меньшей мере в статистическом смысле, и рассчитывает, что система будет работать при каждом из этих условий. Далее разработчик выбирает критерий, по которому должно оцениваться функционирование, например среднее число ошибок между выходным сигналом реальной системы и выходным сигналом некоторой выбранной модели или «идеальной» системы. Наконец, разработчик выбирает систему, которая оказывается лучшей в соответствии с установленным критерием функционирования, обычно из некоторого априорно ограниченного класса (например, из класса линейных систем).

Однако во многих случаях весь диапазон входных условий может быть не известен точно даже в статистическом смысле или условия могут время от времени изменяться. Тогда адаптивная система, которая, используя регулярный процесс поиска, постоянно ищет оптимум в пределах допустимого класса возможностей, имеет преимущества по сравнению с неизменяемой системой.

Адаптивные системы по своей природе должны быть изменяющимися во времени и нелинейными. Их свойства зависят, помимо всего прочего, от входных сигналов. Если на вход подается сигнал  $x_1$ , то адаптивная система настроится на него и сформирует выходной сигнал – назовем его  $y_1$ . Если на вход подается другой сигнал  $x_2$ , то система настроится на этот сигнал и сформирует выходной сигнал – назовем его  $y_2$ . В общем случае структура и процессы коррекции адаптивной системы будут различными для двух различных входных сигналов.

Для получения оптимального решения существует много методов подстройки значений весовых коэффициентов фильтра. Применялись методы случайных возмущений, которые изменяли весовые коэффициенты фильтра; далее анализировался входной сигнал для того, чтобы установить, приближается ли его случайное возмущение к искомому решению или отдаляет от него. В настоящее время для расчета весовых коэффициентов адаптивных фильтров широко применяется адаптивный алгоритм, основанный на методе наименьших квадратов (МНК), поскольку в нем используются градиентные методы, которые значительно эффективнее, чем другие, обеспечивают сходимость к оптимальному решению. Можно показать, что градиентный метод наименьших квадратов очень схож с методом максимизации отношения сигнал – шум, который разрабатывался с целью применения в тех случаях, когда необходимо получить оптимальные весовые коэффициенты адаптивных антенных решеток. Было также показано, что обнуляющий корректирующий фильтр Лаки является упрощением более общего градиентного метода наименьших квадратов.

Таким образом, адаптивный фильтр - это фильтр, передаточная функция (или частотная характеристика) которого адаптируется, т.е. изменяется таким образом, чтобы пропустить без искажений полезные составляющие сигнала и ослабить нежелательные сигналы или помехи [23]. Схема адаптивного фильтра представлена на рис.5.5.

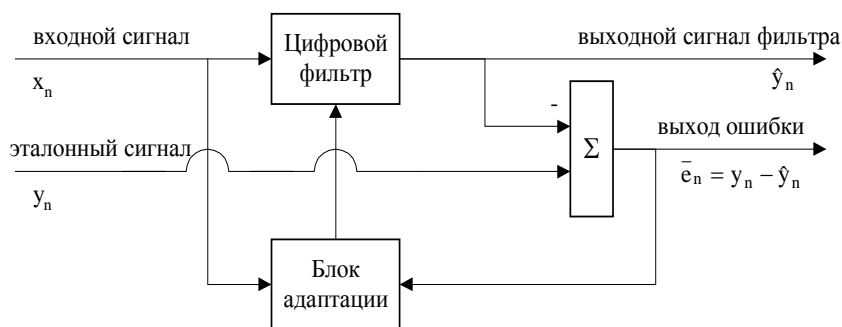


Рис.5.5. Адаптивный фильтр

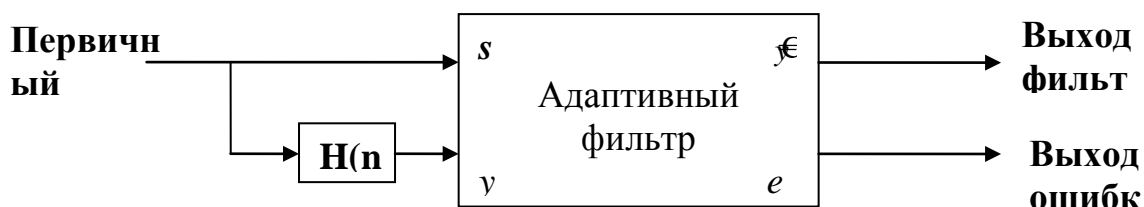
Подобный фильтр действует по принципу оценки статистических параметров сигнала и подстройки собственной передаточной функции таким образом, чтобы минимизировать некоторую целевую функцию. Эту функцию обычно формируют с помощью “эталонного” сигнала на задающем входе. Этот



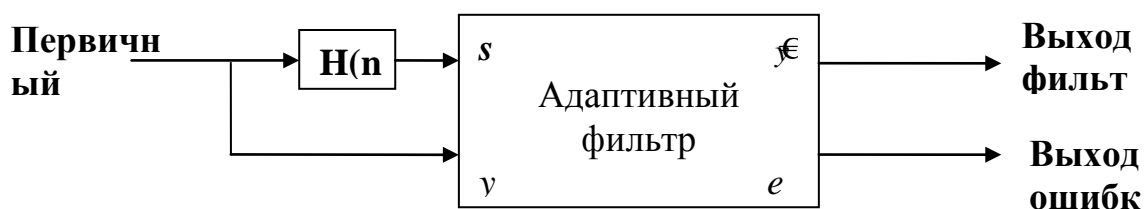
эталонный сигнал можно рассматривать как желаемый сигнал на выходе фильтра. Задача блока адаптации состоит в подстройке коэффициентов цифрового фильтра таким образом, чтобы свести к минимуму разность  $\hat{e}_n = \hat{y}_n - \hat{x}_n$ , определяющую ошибку в работе фильтра.

Важнейшей функцией, выполняемой адаптивным фильтром, является моделирование системы. Это иллюстрируется на рис. 5.6, где первичный сигнал с равномерной спектральной плотностью подается непосредственно либо на вход  $s$ , либо на вход  $y$  адаптивного фильтра. Первичный сигнал поступает на вход системы с импульсной характеристикой  $H(n)$ , выход системы соединен со вторым входом адаптивного фильтра. Для получения оптимальных весовых векторов  $H_{opt}$  адаптивного фильтра можно применить два разных подхода, которые приведут к различным результатам. Это имеет место в следующих случаях:

1. Неизвестная система  $H(n)$  подключена к входу  $y$  адаптивного фильтра (рис. 5.6, а). В этом случае оптимальная импульсная характеристика адаптивного фильтра является точной моделью соответствующей характеристики системы  $H(n)$ .
2. Неизвестная система  $H(n)$  подключена к входу  $s$  адаптивного фильтра (рис. 5.6, б). В этом случае оптимальная импульсная характеристика адаптивного фильтра является функцией, обратной соответствующей характеристике неизвестной системы.



*a*



*б*

Рис. 5.6. Применение адаптивного фильтра для прямого моделирования системы:  $H_{opt} = H(n)$  (а) и обратного моделирования системы:  $H_{opt} = H^{-1}(n)$  (б).

Практическим примером, иллюстрирующим работу адаптивного фильтра первого типа (т.е. прямое моделирование системы), является подавление отраженного сигнала в гибридной телефонной линии.

Примером, которым можно воспользоваться для иллюстрации принципа действия адаптивного фильтра, моделирующего обратную характеристику системы, является коррекция искажений при передаче данных по телефонным линиям. В этом случае вход телефонной линии возбуждается известным сигналом, а искаженный сигнал с выхода линии поступает на вход  $s(n)$  адаптивного фильтра. Затем фильтр перестраивается с помощью подачи на вход  $y(n)$  последовательной серии известных (неискаженных) первичных сигналов. Адаптивный фильтр моделирует импульсную характеристику, обратную характеристике линии, для получения на выходе отфильтрованных (свободных от искажений) данных.

Следующая область применения адаптивных фильтров – подавление шумов. В этой схеме первичный сигнал, содержащий искомую информацию наряду с мешающим сигналом, приложен к входу  $y(n)$ . Затем от другого источника, не содержащего никаких составляющих исходного сигнала, поступает независимый коррелированный сигнал – образец мешающего сигнала. Если этот коррелированный сигнал поступает непосредственно на вход  $s(n)$  адаптивного фильтра, фильтр формирует импульсную характеристику, обеспечивающую получение выходного сигнала  $y(n)$ , который когерентно вычитает из  $y(n)$  нежелательную составляющую, оставляя на выходе  $e(n)$  лишь искомый сигнал.

Одним из примеров использования этого метода является регистрация сердцебиения плода. Первичный сигнал поступает от преобразователь, расположенного на поверхности живота матери. Этот преобразователь вырабатывает сигнал, содержащий импульсы сердечных сокращений плода, которые, однако, существенно маскируются сердцебиением матери. Затем от второго преобразователя, расположенного на груди матери, получают вторичный сигнал, регистрирующий только сердцебиение матери. Далее адаптивный фильтр моделирует тракт искажений от преобразователя, расположенного на груди, до преобразователя расположенного на животе, для получения сигнала, который когерентно вычитается из сигнала с поверхности живота. Адаптивные фильтры применяются и в других случаях, например для устранения шума двигателя в микрофоне пилота в кабине самолета или для подавления акустических шумов окружающей среды, например на крупных электростанциях.

Еще одно применение адаптивных фильтров – это реализация самонастраивающегося фильтра, используемого для выделения синусоиды, маскируемой широкополосным шумом. Такое применение в адаптивном линейном усилителе (АЛУ) осуществляется путем подачи сигнала непосредственно на вход фильтра  $y(n)$  и подачи модификации сигнала с временной задержкой на вход фильтра  $s(n)$ . В том случае, если задержка превышает величину, обратную ширине полосы пропускания фильтра, шумовые составляющие на двух входах не будут коррелированы. Адаптивный

фильтр дает на выходе синусоиду с увеличенным отношением сигнал – шум, тогда как на выходе сигнала ошибки синусоидальные составляющие уменьшаются.

Адаптивные фильтры БИХ-типа в основном применялись для решения таких проблем, как ослабление влияния многолучевого распространения сигнала в системах радиолокации и радиосвязи. В этом случае принятый сигнал содержит исходный передаваемый сигнал, свернутый с импульсной характеристикой канала, которая при многолучевом распространении содержит только нули. Тогда для исключения интерференционных помех адаптивный приемник моделирует характеристику, обратную характеристике канала (рис. 5.6, б). Это наиболее эффективно осуществляется путем использования модели адаптивного фильтра с характеристикой, содержащей только полюсы, причем положения полюсов подбираются таким образом, чтобы они совпадали с положением нулей в характеристике канала.

При конструировании адаптивного фильтра КИХ-типа можно также учесть эту модель, но более экономично воспользоваться рекурсивной структурой, поскольку она реализует инверсную структуру фильтра при его более низком порядке и с меньшими весами. Отсюда с полным основанием можно сказать, что такая структура будет обеспечивать более быструю сходимость, чем ее трансверсальный аналог. Однако для обеспечения устойчивости адаптивного рекурсивного фильтра необходима высокая степень точности при расчете цифровой схемы. Метод адаптивной обработки сигналов на основе фильтров БИХ-типа применяется в электронных радиолокационных измерительных приемниках для выделения импульсов. Адаптивные же фильтры Калмана представляют интерес для идентификации типов радиолокационных колебаний, генерируемых определенными типами излучателей. Они также находят применение при фильтрации и ослаблении влияния многолучевого распространения в высокочастотных (от 3 до 30 МГц) каналах цифровой связи, где первоочередное значение имеет присущая этим фильтрам высокая скорость сходимости.

Большинство КИХ-фильтров строится с учетом довольно простых общепринятых допущений. Эти допущения приводят к хорошо известным несложным алгоритмам адаптации (например, МНК), реализация которых подробно разработана в отношении скорости сходимости, остаточной ошибки и т.д. Таким подходом шире всего пользуются при применении адаптивных фильтров в системах дальней связи, например для выравнивания и гашения отраженного сигнала.

В 1971 г. Чанг внес значительный вклад в классификацию типов фильтров: он предпринял попытку объединить все подходы и создать одну обобщенную структуру выравнивателя, или корректирующего фильтра (рис. 5.7.) [23]. Эта структура содержит набор произвольных фильтров, подключенных к линейной взвешивающей и комбинирующей цепи. Фильтр КИХ-типа можно получить из этой обобщенной структуры путем замены произвольного фильтра линией задержки с отводами, дающей на выходах серию выборок сигнала с временной задержкой. Фильтр БИХ-типа благодаря наличию элементов рекурсивной об-

ратной связи осуществляет дальнейшую обработку сигнала до получения выборок сигнала с временной задержкой, которые последовательно поступают на взвешивающую и комбинирующую цепь.

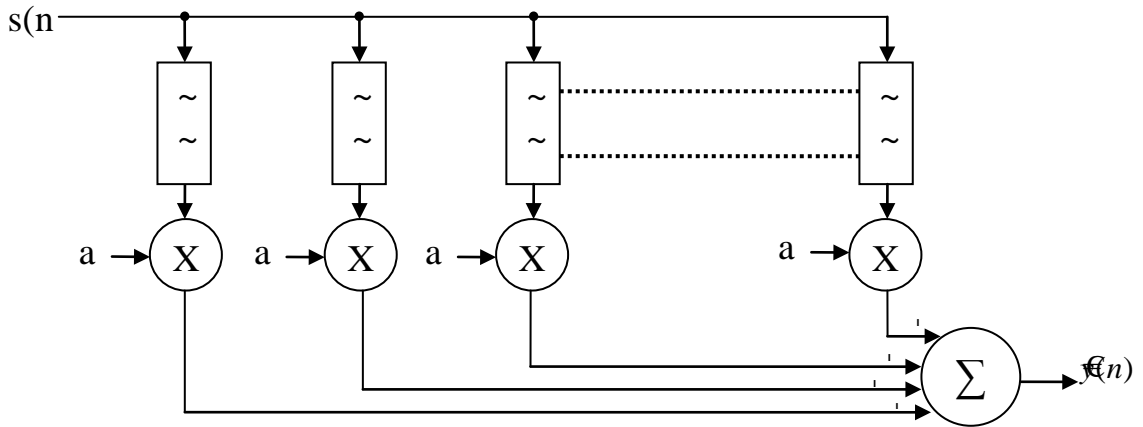


Рис. 5.7. Обобщенная структурная схема корректирующего фильтра.

Альтернативным способом реализации фильтра КИХ-типа является решетчатая структура, которую можно рассматривать в качестве каскадного соединения неразветвленных предсказывающих фильтров ошибки (рис. 5.8). Эта структура, широко применяемая в линейных предсказывающих устройствах для обработки речи, расщепляет сигнал на набор прямых ( $f$ ) и обратных ( $b$ ) выборок разносных сигналов с задержками, добавляемыми в обратном канале. Сигналы умножаются на коэффициенты частичной корреляции PARCOR (partial correlation)  $k(n)$ , названные так из-за их аналогии с коэффициентами отражения дискретной решетки. Прямой разностный PARCOR-коэффициент для любого звена обычно равен комплексно сопряженной величине обратного коэффициента; исключение составляют процессоры, которые производят обработку дискретной информации по основному каналу и в которых эти коэффициенты равны.

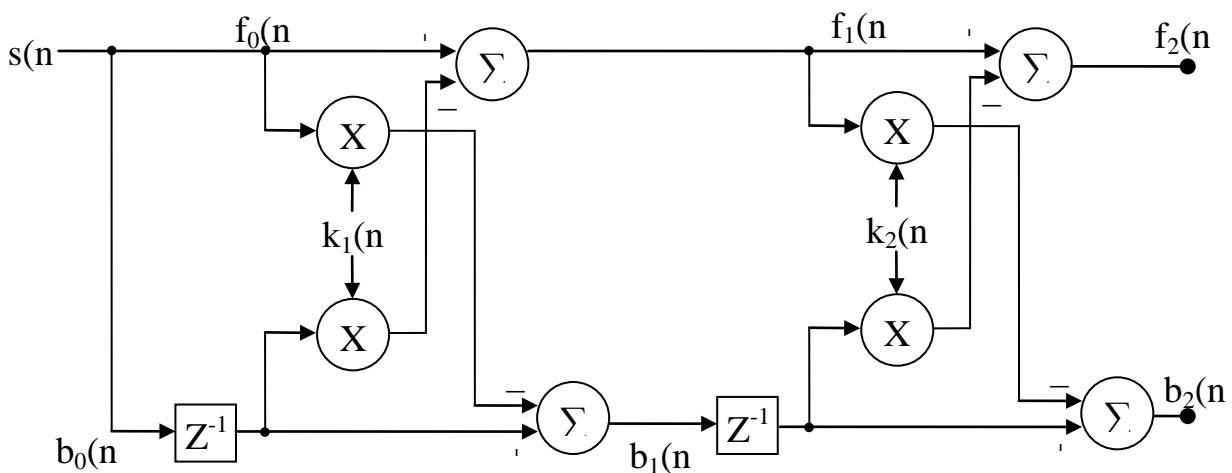


Рис. 5.8. Структурная схема решетчатого фильтра с конечной импульсной характеристикой.

Основным достоинством решетчатой структуры является то, что она измеряет на обратных разностных выходах автокорреляцию сигнала при последовательно возрастающих задержках для получения на выходе серии информационно зависимых ортогональных выборок сигналов, которые затем можно использовать для подачи на взвешивающий комбинатор обобщенной структуры Чанга.

### 5.8.1. Фильтр Винера-Хопфа.

Концепции оптимального линейного оценивания являются фундаментальными при любом рассмотрении адаптивных фильтров. Процесс адаптивной фильтрации включает два этапа проведения оценивания: 1) оценивание искомого выхода фильтра и 2) оценивание весов фильтра, необходимых для достижения вышеупомянутой цели. Второй из этих двух этапов необходим вследствие того, что в случае адаптивной фильтрации характеристики входного сигнала априорно не известны.

Наиболее широко распространенным типом структуры адаптивного фильтра является структура, в которой используется архитектура с конечной импульсной характеристикой (КИХ). Эти фильтры должны сходиться к решению с помощью оптимального нерекурсивного устройства оценки, причем решение задается уравнением Винера – Хопфа [23].

Синтез КИХ и БИХ устройств оценки существенно зависит от определения стоимостной функции, в соответствии с которым качество оценивания характеризуется разностью между выходным сигналом устройства оценки и истинным параметром, подлежащим оцениванию:

$$e(n) = x(n) - \hat{x}(n). \quad (5.22)$$

Здесь  $e(n)$  – ошибка оценивания;  $x(n)$  – случайная величина, которую необходимо оценить и которая может быть детерминированной, а  $\hat{x}(n)$  – оценка  $x(n)$ , выполненная с помощью нашей системы оценивания, причем

$$\hat{x}(n) = f\{y(n), h(n)\}, \quad (5.23)$$

т.е.  $x(n)$  – линейная функция последовательности входных сигналов  $y(n)$  и набора весов фильтра  $h(n)$ . Наблюдаемую последовательность сигналов  $y(n)$  в общем виде можно представить как исходную последовательность  $x(n)$ , искаженную адаптивным белым шумом  $v(n)$  с дисперсией  $\sigma_v^2$ :

$$y(n) = x(n) + v(n). \quad (5.24)$$

Наиболее употребительным при проведении оптимального оценивания  $\hat{x}(n)$  является метод наименьших квадратов (МНК). Среднеквадратическая ошибка определяется как

$$E\{e^2(n)\} = \overline{[x(n) - \hat{x}(n)]^2}. \quad (5.25)$$

Она минимизируется относительно весовых коэффициентов устройства оценки для получения оптимального оценивания по критерию МНК. Следует отметить, что можно применять не только описанную функцию стоимости. Альтернативными будут такие функции, как абсолютная величина ошибки и нелинейная пороговая функция. Такая функция ошибки используется в том случае, когда имеется приемлемый интервал ошибок (т.е. существует заданная

допустимая ошибка). При использовании критерия наименьшего среднеквадратичного малые ошибки вносят меньший вклад, чем большие ошибки (в противоположность критерию абсолютной величины ошибки, который дает одинаковый вес для всех ошибок).

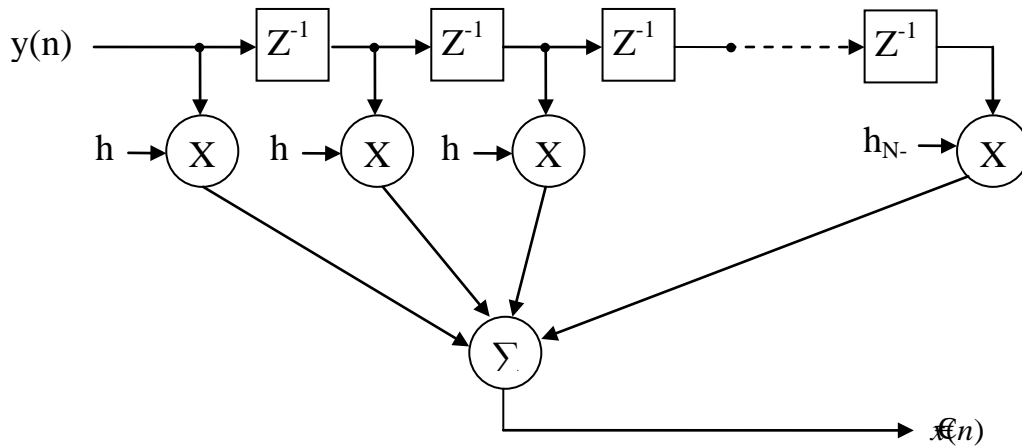


Рис. 5.9. Обобщенный нерекурсивный фильтр или устройство оценки.

В нерекурсивном устройстве оценки оценка  $x(n)$  определяется в виде конечного линейного полинома  $y(n)$ :

$$\hat{x}(n) = \sum_{k=0}^{N-1} y(n-k)h_k, \quad (5.26)$$

где  $h_k$  – отдельные веса в структуре нерекурсивного фильтра КИХ-типа, показанного на рис. 5.9. Выражение (5.28) можно переписать в матрично-векторной системе обозначений:

$$\hat{x}(n) = Y^T(n)H = H^T Y(n), \quad (5.27)$$

где

$$Y^T(n) = [y(n) y(n-1) \dots y(n-N+1)] \quad \text{и} \quad H^T = [h_0 h_1 h_2 \dots h_{N-1}],$$

а верхний индекс T обозначает транспонирование матрицы. Тогда функция среднеквадратичной ошибки принимает вид

$$E\{e^2(n)\} = E\{x(n) - H^T Y(n)\}^2. \quad (5.28)$$

Это выражение описывает стандартную поверхность квадратичной ошибки с одним единственным минимумом. Дифференцирование (5.30) по  $H^T$  дает

$$\frac{\partial e^2(n)}{\partial H^T} = -2E\{[x(n) - H^T Y(n)]Y^T(n)\}. \quad (5.29)$$

а допуская, что (5.31) равно нулю, имеем

$$\begin{aligned} E\{[x(n) - H^T Y(n)]Y^T(n)\} &= 0 \\ E\{x(n)Y^T(n)\} &= E\{H^T Y(n)Y^T(n)\}. \end{aligned} \quad (5.30)$$

Полагая, что весовой вектор  $H^T$  и вектор сигнала  $Y(n)$  не коррелированы, получаем

$$E\{x(n)Y^T(n)\} = H^{T_{opt}} E\{Y(n)Y^T(n)\}. \quad (5.31)$$

Члены математического ожидания, входящие в (5.33), можно определить следующим образом:

$P = E\{x(n)Y(n)\}$  – взаимная корреляция между входным сигналом и оцениваемым параметром;

$R = E\{Y(n)Y^T(n)\}$  – автокорреляционная матрица входной сигнальной последовательности.

Тогда (5.33) можно переписать в виде

$$P^T = H_{opt}^T R. \quad (5.32)$$

Уравнение (5.34) является общеизвестным уравнением Винера – Хопфа, которое дает оптимальное (по методу наименьших квадратов) винеровское решение для  $H$ :

$$H_{opt} = R^{-1}P. \quad (5.33)$$

### 5.10. Фильтр Калмана.

Винеровская оценка, рассмотренная в предыдущем разделе, по существу является блочным процессом оценки, который лучше всего подходит для случая, когда в распоряжении имеется лишь конечная выборка (блок) данных, и оценку можно произвести на «автономном» компьютере. Однако если иметь дело с бесконечным временным рядом, винеровская оценка для каждой новой выборки потребовала бы полного пересчета всех членов авто- и взаимно-корреляционных функций. В оптимальном рекурсивном (или калмановском) устройстве оценки поступающая информация используется для корректировки рекурсивной оценки [23].

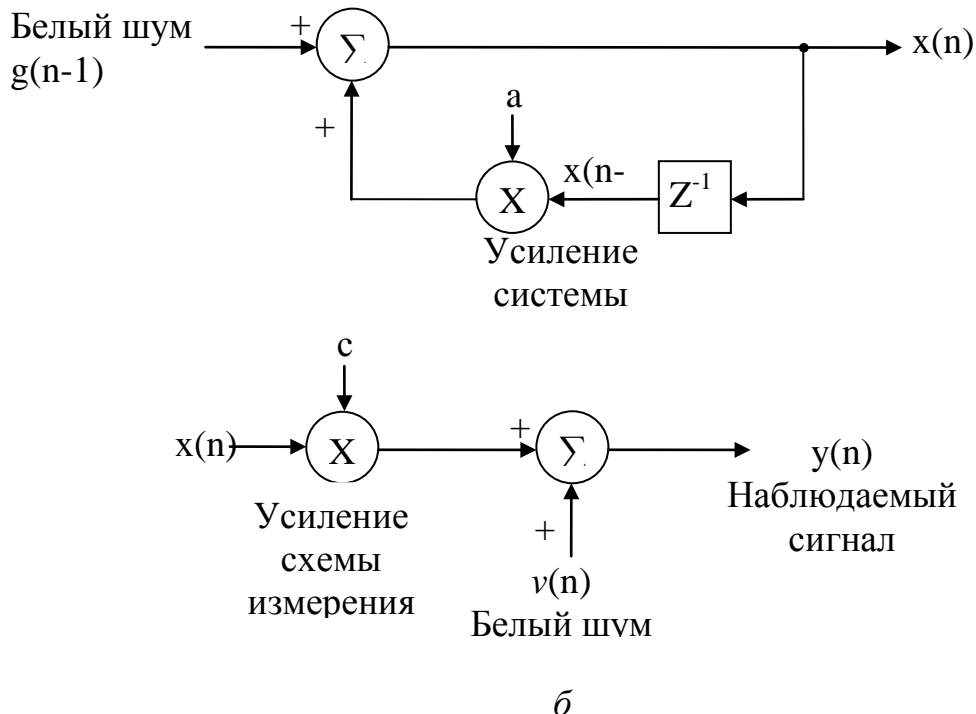


Рис. 5.10. а – рекурсивная модель генерации сигнала; б – модель схемы измерения данных.

По существу, устройство калмановской оценки реализует процесс параметрического оценивания, основанный на авторегрессивной (АР) модели процесса генерации сигнала. АР-модель процесса первого порядка данного типа показана на рис. 5.10, а, а соответствующая модель измерений – на рис. 5.10, б. Модель измерений представляет просто усилительное зерно  $c$  и источник адаптивного белого шума  $v(n)$ . Приняв эту модель генерации сигнала, поступающую выборку сигнала с номером  $n$  можно определить как

$$y(n) = cx(n) + v(n)$$

Рекурсивная формула оценки первого порядка имеет вид

$$\hat{x}(n) = b(n)\hat{x}(n-1) + k(n)y(n).$$

Отметим, что коэффициенты передачи обоих усилительных звеньев зависят от времени (структурная схема этого устройства оценки в общем виде показана на рис. 5.11).

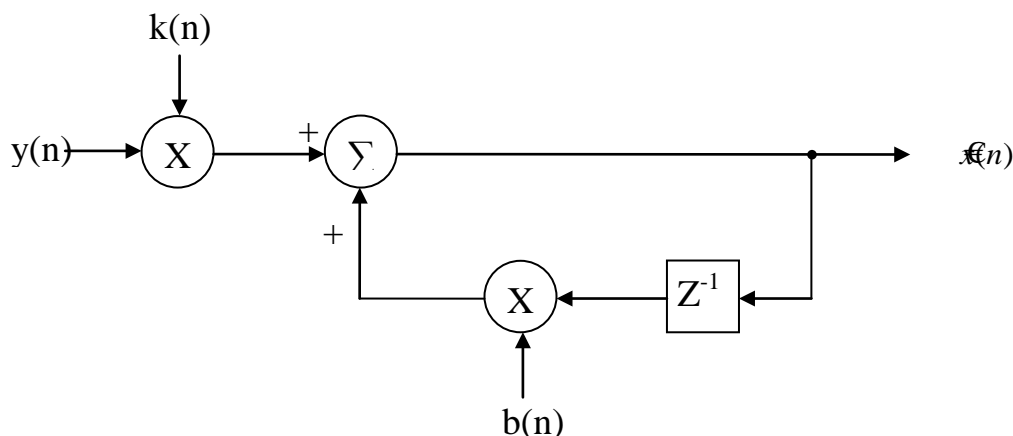


Рис. 5.11. Обобщенная структурная схема рекурсивного устройства оценки первого порядка.

Для получения оптимального (с точки зрения метода наименьших квадратов) устройства оценки среднеквадратическая ошибка  $p(n)$  дифференцируется по  $b(n)$  и  $k(n)$ , а результаты приравниваются нулю:

$$p(n) = E[\hat{x}(n) - x(n)]^2 = \\ = E[b(n)\hat{x}(n-1) + k(n)y(n) - x(n)]^2,$$

$$\frac{\partial p(n)}{\partial b(n)} = 2E\{[b(n)\hat{x}(n-1) + k(n)y(n) - x(n)]\hat{x}(n-1)\} = 0,$$

$$\frac{\partial p(n)}{\partial k(n)} = 2E\{[b(n)\hat{x}(n-1) + k(n)y(n) - x(n)]y(n)\} = 0.$$

Соотношение между  $b(n)$  и  $k(n)$  можно вывести следующим образом:

$$2E\{[b(n)\hat{x}(n-1) + k(n)y(n) - x(n)]\hat{x}(n-1)\} = 0 \Rightarrow \\ \Rightarrow E\{[b(n)\hat{x}(n-1)]\hat{x}(n-1)\} = E\{[k(n)y(n) - x(n)]\hat{x}(n-1)\} \Rightarrow \\ \Rightarrow E\{[b(n)[\hat{x}(n-1) - x(n-1)] + b(n)x(n-1)]\hat{x}(n-1)\} = \\ = E\{[x(n) - k(n)y(n)]\hat{x}(n-1)\}.$$

Подставив значение для  $y(n)$ , находим

$$b(n)E\{e(n-1)\hat{x}(n-1) + x(n-1)\hat{x}(n-1)\} =$$



$$E\{[x(n)[1 - ck(n)] - k(n)v(n)]\epsilon(n-1)\}.$$

Для оптимального устройства оценки должен выполняться принцип ортогональности, который приводит к следующим соотношениям:

$$E[e(n)\epsilon(n-1)] = 0 \text{ и } E[v(n)\epsilon(n-1)] = 0.$$

Тогда уравнение примет вид

$$b(n)E[x(n-1)\epsilon(n-1)] = [1 - ck(n)]E[x(n)\epsilon(n-1)].$$

Из нашей модели генерации сигнала имеем

$$x(n) = ax(n-1) + g(n-1).$$

С учетом этого получаем

$$b(n)E[x(n-1)\epsilon(n-1)] = [1 - ck(n)]E[ax(n-1)\epsilon(n-1) + g(n-1)\epsilon(n-1)].$$

Откуда находим

$$\epsilon(n) = b(n)\epsilon(n-1) + k(n)cx(n) + k(n)v(n),$$

а подстановка  $x(n)$  дает:

$$\epsilon(n) = b(n)\epsilon(n-1) + k(n)acx(n-1) + k(n)cg(n-1) + k(n)v(n),$$

$$\epsilon(n-1) = b(n-1)\epsilon(n-2) + k(n-1)acx(n-2) + k(n-1)cg(n-2) + k(n-1)v(n-1),$$

и поскольку среднее всех произведений членов на  $g(n-1)$  равно нулю, можно записать

$$E[\epsilon(n-1)g(n-1)] = 0.$$

Воспользовавшись этим соотношением, преобразуем:

$$b(n)E[x(n-1)\epsilon(n-1)] = a[1 - ck(n)]E[x(n-1)\epsilon(n-1)].$$

Это приводит к окончательному соотношению между  $b(n)$  и  $k(n)$ :

$$b(n) = a[1 - ck(n)].$$

В конечном виде получаем

$$\epsilon(n) = a\epsilon(n-1) + k(n)[y(n) + ac\epsilon(n-1)]. \quad (5.34)$$

Уравнение (5.34) является определением рекурсивного устройства оценки первого порядка, или скалярного фильтра Калмана. Первый член  $a\epsilon(n-1)$  предсказывает текущую выборку, а второй член корректирует на основании оценки ошибки с учетом калмановского коэффициента  $k(n)$ . Структура такого фильтра иллюстрируется на рис. 5.12.

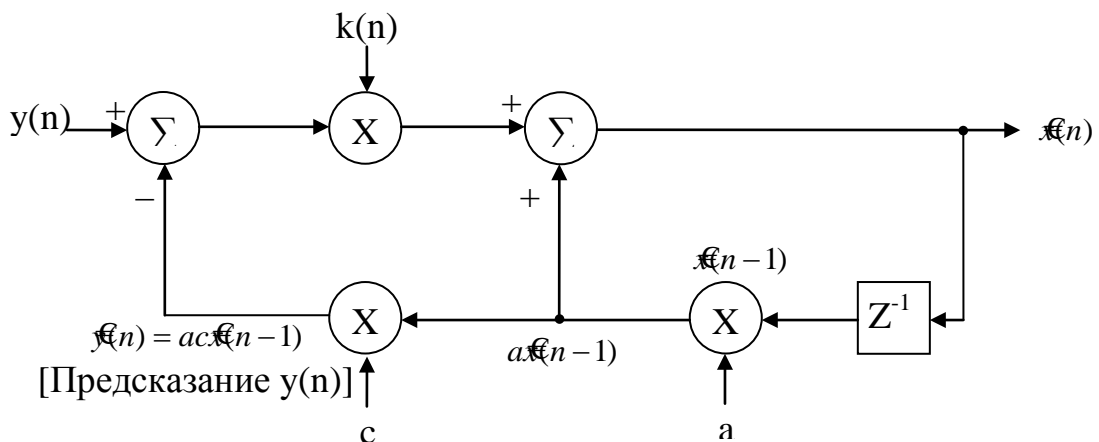


Рис. 5.12. Блок-схема скалярного фильтра Калмана первого порядка.

Определив структуру фильтра Калмана, необходимо получить выражения для изменяющегося во времени коэффициента усиления Калмана  $k(n)$ . Сначала, определим среднеквадратическую ошибку в виде

$$p(n) = E\{e(n)[b(n)x(n-1) + k(n)y(n) - x(n)]\} = -E\{e(n)x(n)\} \quad (5.35)$$

Используем (5.34) вместо  $x(n)$  для подстановки в (5.35):

$$cE\{e(n)x(n)\} = -E\{e(n)v(n)\}. \quad (5.36)$$

Подставляя (5.36) в (5.35), находим

$$p(n) = \frac{1}{c} E\{e(n)v(n)\};$$

раскрывая  $e(n)$  и используя выражение для  $x(n)$ , получаем

$$p(n) = \frac{1}{c} k(n) E\{e(n)v(n)\} = \frac{1}{c} k(n) \sigma^2,$$

$$k(n) = \frac{cp(n)}{\sigma_v^2},$$

где  $\sigma_v^2 = E[v(n)]^2$ . Теперь, подставляя (5.34) в выражение для среднеквадратической ошибки, имеем

$$p(n) = a^2[1 - ck(n)]^2 p(n-1) + [1 - ck(n)]^2 \sigma_g^2 + k^2(n) \sigma_v^2,$$

откуда получаем

$$k(n) = \frac{c[a^2 p(n-1) + \sigma_g^2]}{\sigma_v^2 + c^2 \sigma_g^2 + c^2 a^2 p(n-1)}. \quad (5.37)$$

Отметим, что сначала, зная  $p(n-1)$ , надо рассчитать  $k(n)$ , а затем уже  $p(n)$  по формуле

$$p(n) = \frac{1}{c} \sigma_v^2 k(n)., \quad (5.38)$$

Три уравнения: (5.34), (5.37) и (5.38) – являются рекурсивными уравнениями, необходимыми для реализации фильтра Калмана первого порядка. В отличие от фильтра Винера усиление фильтра Калмана должно быть выражено итерационным соотношением, а следовательно, его нельзя представить в виде универсального стационарного решения.

**Методы обращения матриц.** Для фильтра Винера-Хопфа при вычислении оптимального импульсного отклика фильтра необходимо выполнять обращение автокорреляционных или ковариационных матриц. Поэтому остановимся на методах обращения матриц, используемых при решении задач ЦОС.

Вычислительная сложность процедуры обращения матриц достаточно велика, поэтому предназначенные для решения задач ЦОС в реальном масштабе времени алгоритмы обращения должны допускать распараллеливание и конвейеризацию вычислений. По этим причинам для обращения матриц в задачах ЦОС часто используют методы обращения, основанные на процедуре исключения (алгоритмы Гаусса, Гаусса-Жордана) [17, 21]. Рассмотрим процедуру обращения матриц по методу Гаусса-Жордана.

Пусть для матрицы  $A_N$  порядка  $N$  требуется найти обратную матрицу  $Z_N$ :

$$Z_N = A_N^{-1}$$

Заметим, что для матрицы  $A_N$  может и не существовать обратная матрица. Для того, чтобы существовала матрица  $Z_N$ , необходимо и достаточно, чтобы матрица  $A_N$  была не вырожденной, т.е. чтобы определитель матрицы  $\det[A_N] \neq 0$ .

Запишем матричное уравнение:

$$A_N Z_N = I_N$$

где  $I_N$  - единичная матрица. Сформируем матрицу вида:

$$\hat{A}_0 = [A_N^1 | I_N]$$

размером  $N \times 2N$  и за  $N$  итераций приведем ее к виду:

$$\hat{A}_N = [I_N | Z_N]$$

при помощи соотношений:

$$\begin{cases} b^k = a_{kk}^{k-1}, k = \overline{1, N} \\ a_{kj}^k = \frac{a_{kj}^{k-1}}{b^k}, j = \overline{k, 2N} \end{cases} \quad (5.39)$$

$$\begin{cases} U_l^k = a_{lk}^{k-1}, l = \overline{1, N} \\ a_{ij}^k = a_{ij}^{k-1} - a_{kj}^k U_l^k, l \neq k \end{cases} \quad (5.40)$$

причем  $a_{ij}^{k-1} = \hat{a}_{ij}^{k-1}$ . Здесь  $b^k$  - ведущий элемент, который может определяться по номеру итерации как  $k$ -тый элемент  $k$ -й строки, либо как максимальный элемент  $k$ -го столбца [17]. Выражение (5.65) определяет главный или ведущий элемент на каждой итерации и приведение (масштабирование) коэффициентов строки, содержащей главный элемент. Назовем такую строку ведущей. Выражение (5.29) определяет порядок исключения, т.е. вычитания ведущей строки из остальных строк матрицы.

Подобный алгоритм почти в  $N$  раз снижает вычислительные затраты по сравнению с обращением матрицы по формулам Крамера.

Тем не менее, вычислительная сложность процедуры обращения матрицы по выражениям (5.39) и (5.40) все же достаточно велика и составляет порядка  $N^3$  базовых операций.

С целью дальнейшего снижения вычислительной сложности в ряде случаев используют приближенные итерационные процедуры обращения, например, по алгоритму Ньютона. Согласно такому алгоритму, для матрицы  $A_N$  выполняются последовательные приближения:

$$X_N^{j+1} = X_N^j (2X_N^{(0)} - A_N X_N^j), \quad j = 0, 1, 2, \dots \quad (5.41)$$

где  $X_N^{(0)}$  - произвольное начальное значение исходной матрицы. Если данная последовательность сходится, то ее пределом является  $Z_N = A_N^{-1}$ .

Для ряда практических применений, например, расчета оптимального импульсного отклика адаптивного фильтра, используют упрощенные алгоритмы, к которым относится алгоритм Гриффитса [23]:

$$Y^{(k)} = N_N^{(k)} X^{(k)}$$

$$N_N^{(k)} = R_N^{-1(k)} \quad (5.42)$$

$$r_{mn} = x_m^{(k)} y_n^{(k)}$$

Вместо нахождения обратной матрицы при нахождении импульсного отклика фильтра Винера-Хопфа:

$$\hat{H} = R^{-1}$$

вычисляют матрицу:

$$\tilde{H}^{(k+1)} = \tilde{H}^{(k)} - \mu[I_N - R_N] \quad (5.43)$$

где  $\mu$  - некоторая действительная переменная:

$$\mu = \frac{1}{\sum_m (\overline{x_m})^2}$$

Однако при этом остается открытым вопрос о сходимости последовательности  $\tilde{H}^{(k+1)}$ , что требует в конкретном случае предварительного исследования такой сходимости.

## 6. ДИСКРЕТНЫЕ ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ

Линейное преобразование может быть записано как векторно-матричная процедура:

$$F = kB_N X$$

где  $X = [x_0, x_1, \dots, x_{N-1}]^T$  - вектор исходных данных;  $B_N$  - квадратная матрица размера  $N \times N$  ядра преобразования;  $F = [f_0, f_1, \dots, f_{N-1}]^T$  - вектор результатов.

Если для матрицы  $B_N$  существует обратная матрица  $B_N^{-1}$ , то существует и обратное преобразование:

$$X = kB_N^{-1} F.$$

Однако обратная матрица может быть определена лишь в том случае, если  $\det[B]$  отличен от нуля.

Методы дискретных ортогональных преобразований в настоящее время широко используются при обработке сигналов. В таблице 6.1 приведены основные области применения спектральных методов и задачи, решаемые с их помощью [11].

Таблица 6.1

**Задачи ЦОС, решаемые методами дискретных ортогональных преобразований**

<b>Решаемые задачи</b>	<b>Область применения</b>	<b>Выполняемые преобразования</b>
Формирование характеристик направленности антенных устройств	Радиофизика, радиолокация, гидроакустика	Двумерное ДПФ матрицы данных
Выделение сигнала на фоне шумов	Радиофизика, радиолокация, обработка изображений	Прямое ДПФ + сглаживание спектра + обратное ДПФ
Обнаружение и определение координат объекта на изображении	Радиофизика, гидроакустика, обработка изображений	Прямое ДПФ + согласованная фильтрация спектра + обратное ДПФ
Определение скорости движущегося объекта	Радиолокация, гидроакустика	Одномерное ДПФ временной последовательности сигнала
Сжатие и восстановление изображений	Системы передачи, обработки и архивации изображений	Двумерное косинусное прямое или обратное преобразование по окну
Восстановление изображения по проекциям	Томография	Преобразование Радона для векторов данных через ДПФ

## 6.1. Дискретное преобразование Фурье

Перейдём от интегрального преобразования Фурье (2.3) к дискретному преобразованию Фурье (ДПФ), при условии что точки дискретизации выбраны согласно теореме отсчётов (теоремы Котельникова) [5,21]:

$$\begin{aligned}
 F(\Delta\xi) &= \sum_{k=0}^{N-1} x(k\Delta x) \delta(x - k\Delta x) * e^{-j2\pi k\Delta\xi\Delta x} = \sum_{k=0}^{N-1} x_k e^{-j2\pi \frac{2x_{\max} * k}{N} \frac{1}{2x_{\max}} n} = \\
 &= k * \sum_{k=0}^{N-1} x_k * e^{-j\frac{2\pi}{N} kn}
 \end{aligned}$$

$$\text{где } \begin{cases} \frac{2x_{\max}}{N} = \Delta x \\ \frac{1}{2x_{\max}} = \Delta\xi \end{cases} \quad (6.1)$$

Тогда нетрудно получить, что

$$\Delta x * \Delta\xi = \frac{1}{N}$$

Подобным же образом можно получить и для обратного преобразования

$$x_m = k \sum F_n * e^{j\frac{2\pi}{N} nm} \quad (6.2)$$

$$k = \frac{1}{\sqrt{N}}$$

Заметим, что происхождение множителя  $k = \frac{1}{\sqrt{N}}$  связано с заменой при дискретизации согласно теореме Котельникова восстанавливающую функцию в (6.2) на “гребёнку” отсчётов.

Таким образом, в матричной форме:

$$F = \frac{1}{\sqrt{N}} E_N X \quad (6.3)$$

$$E_N = \begin{pmatrix} W^0 & W^0 & W^0 & \dots & W^0 \\ W^0 & W^1 & W^2 & \dots & W^{N-1} \\ W^0 & W^2 & W^4 & \dots & W^{2(N-1)} \\ W^0 & W^3 & W^6 & \dots & W^{3(N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ W^0 & W^{N-1} & W^{2(N-1)} & \dots & W^{(N-1)(N-1)} \end{pmatrix}$$

где  $W^k = e^{-j2\pi k/N}$ , а сама матрица ядра ДПФ носит название матрицы дискретных экспоненциальных функций (ДЭФ). При этом строки матрицы определяют набор ортогональных функций или базис разложения.

При выполнении преобразования Фурье строки матрицы ядра задают набор ортогональных функций, по которым выполняется разложение исходного

сигнала. Каждый элемент вектора результата определяет вклад соответствующей ортогональной функции в формирование исходного сигнала.

Для преобразования Фурье, как и для любого ортогонального преобразования, матрица ядра преобразования  $E_N$  обратима (т.е. определитель отличен от "0"), что позволяет выполнить как прямое, так и обратное преобразования:

$$\begin{cases} F = \frac{1}{\sqrt{N}} E_N X \\ X = \frac{1}{\sqrt{N}} E_N^{-1} F \end{cases} \quad (6.4),$$

поскольку  $\frac{1}{N} E_N * E_N^{-1} = \frac{1}{N} I_N =$

$$\begin{vmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{vmatrix}$$

При этом матрица ядра обратного преобразования обладает свойством  $E_N^{-1} \equiv E_N^*$ , где  $E_N^*$  - эрмитово-сопряжённая матрица. Понятие эрмитово-сопряжённой матрицы предусматривает, что матрица обратного преобразования является транспонированной по отношению к  $E_N$  и элементы её есть комплексно сопряжённые к  $W_{ij}$ .

Рассмотрим основные свойства матрицы ядра преобразования  $E_N$ . Коэффициенты такой матрицы обладают следующими свойствами:

1) цикличность:  $W^{k+N} \equiv W^k$  или  $W^{-k} \equiv W^{N-k}$

2) мультипликативностью  $W^{k+m} \equiv W^k * W^m$

Из указанных свойств следует, матрица  $E_N$  из  $N^2$  элементов содержит только  $N$  попарно различных элементов.

3) симметричностью  $W_{ij} \equiv W_{ji}$

Рассмотрим примеры матрицы  $E_N$  для некоторых  $N$ :

$N=2$

$$E_2 = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$$

$N=3$

$$E_3 = \begin{vmatrix} W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 \\ W^0 & W^2 & W^1 \end{vmatrix}$$

$$W^4 \equiv W^{3+1} \equiv W^1$$

$N=4$

$$E_4 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{vmatrix}$$

и, наконец, для  $N=8$

$$E_8 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{\sqrt{2}}{2}(1-j) & -j & -\frac{\sqrt{2}}{2}(1+j) & -1 & \frac{\sqrt{2}}{2}(-1+j) & j & \frac{\sqrt{2}}{2}(1+j) \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & -\frac{\sqrt{2}}{2}(1+j) & j & \frac{\sqrt{2}}{2}(1-j) & -1 & \frac{\sqrt{2}}{2}(1+j) & -j & \frac{\sqrt{2}}{2}(-1+j) \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \frac{\sqrt{2}}{2}(1-j) & j & \frac{\sqrt{2}}{2}(1+j) & -1 & \frac{\sqrt{2}}{2}(1-j) & -j & \frac{\sqrt{2}}{2}(-1-j) \\ 1 & j & -1 & -j & 1 & -j & -1 & -j \\ 1 & \frac{\sqrt{2}}{2}(1+j) & j & \frac{\sqrt{2}}{2}(-1+j) & -1 & -\frac{\sqrt{2}}{2}(1+j) & -j & \frac{\sqrt{2}}{2}(1-j) \end{vmatrix}$$

## 6.2. Дискретное преобразование Хартли

Дискретное преобразование Хартли имеет вид [3]:

- прямое преобразование

$$h_n = \sum_{k=0}^{N-1} x_k \cos \frac{2\pi kn}{N} + \sin \frac{2\pi kn}{N} = \sum_{k=0}^{N-1} x_k \text{cas} \frac{2\pi kn}{N} \quad (6.5)$$

$$\text{cas}[\dots] = \cos[\dots] + \sin[\dots]$$

- обратное преобразование

$$x_m = \sum_{n=0}^{N-1} h_n \text{cas} \frac{2\pi mn}{N}$$

Матрица ядра преобразования Хартли может быть записана как:

$$C_N = \begin{vmatrix} C_0 & C_0 & C_0 & \dots & C_0 \\ C_0 & C_1 & C_2 & \dots & C_{N-1} \\ C_0 & C_2 & C_4 & \dots & C_{2(N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ C_0 & C_{N-1} & C_{2(N-1)} & \dots & C_{(N-1)^2} \end{vmatrix}$$

где  $C_k = \text{cas} \frac{2\pi k}{N}$ , причем  $C_0 \equiv 1$

Матрица ядра преобразования обладает следующими свойствами:

- 1) цикличностью  $C_{k+N} \equiv C_k$ ;  $C_{N-k} \equiv C_{-k}$ ;
- 2) отсутствием мультипликативности, т.е.:  $C_{k+l} \neq C_k * C_l$
- 3) симметричностью  $C_{ij} \equiv C_{ji}$

Отсюда следует, что обратная матрица  $C_N^{-1} \equiv C_N$ , поскольку  $C_N \equiv C_N^T$ .

4) из свойства цикличности следует, что в матрице  $C_N$  имеется лишь  $N$  различных между собой коэффициентов  $C_{ij}$  из  $N^2$



Для  $N=2$ ,  $N=4$  и  $N=8$  матрицы ядра преобразования будут иметь вид:

$$\begin{aligned}
 N=2 \quad C_2 &= \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} = E_2 \\
 N=4 \quad C_4 &= \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix} \neq E_4 \\
 N=8 \quad C_8 &= \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \sqrt{2} & 1 & 0 & -1 & -\sqrt{2} & -1 & 0 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & \sqrt{2} & -1 & 0 & 1 & -\sqrt{2} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\sqrt{2} & 1 & 0 & -1 & \sqrt{2} & -1 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & -1 & -\sqrt{2} & -1 & 0 & 1 & \sqrt{2} \end{vmatrix}
 \end{aligned}$$

Таким образом, в отличие от матрицы ДЭФ, матрица ядра преобразования Хартли содержит коэффициенты, большие единицы.

### 6.3. Двумерные дискретные преобразования Фурье и Хартли

Для обработки двумерных сигналов, в частности, изображений, широко используются двумерные спектральные преобразования. Двумерное дискретное преобразование Фурье имеет вид [7, 21]:

$$f_{k,l} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} e^{-j\frac{2\pi}{N}[km+ln]} \quad (6.6)$$

Поскольку ядро преобразования Фурье является разделимым по переменным интегрирования, то для выполнения двумерного ДПФ на практике используется строчно-столбцовый метод.

Действительно,

$$e^{-j\frac{2\pi}{N}[km+ln]} = e^{-j\frac{2\pi}{N}km} e^{-j\frac{2\pi}{N}ln}$$

откуда получаем с учетом (2.16):

$$f_{k,l} = \sum_{n=0}^{N-1} \left[ \sum_{m=0}^{N-1} x_{m,n} e^{-j\frac{2\pi}{N}km} \right] e^{-j\frac{2\pi}{N}ln}$$

Введя обозначение

$$\hat{f}_{k,l} = \sum_{m=0}^{N-1} x_{m,n} e^{-j\frac{2\pi}{N}km} \quad (6.7)$$

получим:

$$f_{k,l} = \sum_{n=0}^{N-1} \hat{f}_{k,l} e^{-j \frac{2\pi}{N} ln} . \quad (6.8)$$

Согласно такому подходу, вначале выполняются одномерные ДПФ по строкам матрицы с отсчетами изображения, а затем выполняются одномерные ДПФ по столбцам матрицы промежуточных результатов, представляющих собой одномерные спектры по строкам изображения. В матричном виде это может быть записано следующим образом [7]:

$$F_N = [E_N X_N] E_N \quad (6.9).$$

Из (2.19) следует, что двумерное ДПФ сводится вначале к одномерному ДПФ матрицы  $X$  по столбцам, а затем к одномерному ДПФ по строкам матрицы промежуточных результатов  $\hat{F}_N$ .

Сложность вычислений на первом этапе составит  $N$  раз по  $N^2$  базовых операций, под которыми понимаются операции вычисления выражения под знаком суммы в (6.7) и (6.8), и столько же на втором, откуда:  $Q_{ДПФ} = 2N^3$  (Б.О.)

В то же время непосредственные вычисления двумерного ДПФ по формуле (2.16) требуют вычислительных затрат:  $Q_{ДПФ_2} = N^4$  (Б.О.)

Таким образом, строчно-столбцовый метод позволят существенно снизить вычислительную сложность алгоритма двумерного ДПФ с  $N^4$  до  $2N^3$  операций умножения. Вычисление одномерного ДПФ по каждой из координат преобразование выполняется на основе процедуры быстрого преобразования Фурье. Такой подход, однако, требует выполнения дополнительной процедуры транспонирования матрицы промежуточных результатов после выполнения обработки по одной из координат матрицы. При этом преобразование по следующей координате может выполняться только после того, как сформирована вся матрица промежуточных результатов и выполнено ее транспонирование.

Поэтому использование свойства разделимости ядра двумерного ДПФ по переменным суммирования приводит к снижению сложности вычислений в  $N/2$  раз.

Однако далеко не все ядра двумерных спектральных ортогональных преобразований в явном виде обладают свойством разделимости. К таким преобразованиям относится, например, двумерное дискретное преобразование Хартли [3]:

$$H_{kl} = \sum_m \sum_n x_{mn} [ \text{cas}(2\pi(km + ln) / N) ]$$

где  $\text{cas}[\dots] = \cos[\dots] + \sin[\dots]$

Рассмотрим применение специального или модифицированного преобразования Хартли, в котором искусственно выполнено разделение ядра по координатам [3]:

$$H_{kl}^{\wedge} = \sum_n \left[ \sum_m x_{mn} \text{cas}(2\pi km / N) \right] \text{cas}(2\pi ln / N) \quad (6.10)$$

Очевидно, что выражение (6.11) может быть представлено в виде:

$$\hat{H}_{kl} = \sum_m [\cos(2\pi km / N) + \sin(2\pi km / N)] * \left\{ \sum_n x_{mn} [\cos(2\pi ln / N) + \sin(2\pi ln / N)] \right\}$$

Представляет интерес установить взаимосвязь между традиционным двумерным преобразованием Хартли (выражение (6.20)) и модифицированным преобразованием Хартли, определяемым согласно выражению (6.21). После ряда тригонометрических и алгебраических преобразований ядра в выражении (2.21) нетрудно получить [24]:

$$H_{kl} = [H_{kl}^{\wedge} + H_{k,-l}^{\wedge} + H_{-k,l}^{\wedge} - H_{-k,-l}^{\wedge}] / 2 \quad (6.11)$$

где в силу цикличности ядра  $(N-k) \bmod N = -k$ ;  $(N-l) \bmod N = -l$ .

От компонент спектра Хартли  $H_{kl}$  можно перейти к компонентам спектра Фурье, используя известное соотношение:

$$F_{kl} = \left\{ [H_{kl} + H_{-k,-l}] - j[H_{kl} - H_{-k,-l}] \right\} / 2$$

Если записать  $H_{-k,-l}$  в виде, подобном выражению (6.12), то получим выражение, определяющее связь между компонентами спектра Фурье и компонентами модифицированного преобразования Хартли:

$$F_{kl} = \left\{ [H_{k,-l}^{\wedge} + H_{-k,l}^{\wedge}] - j[H_{k,l}^{\wedge} - H_{-k,-l}^{\wedge}] \right\} / 2 \quad (6.12)$$

Следует подчеркнуть, что определение  $F_{kl}$  как через традиционное, так и через модифицированное преобразование Хартли требует вычисления только лишь  $H_{kl}$  или  $H_{kl}^{\wedge}$  соответственно. Остальные требуемые слагаемые легко могут быть найдены путем перекомпоновки матрицы  $H_{kl}$  или  $H_{kl}^{\wedge}$

Выражения (6.12) и (6.13) справедливы для любых действительных функций  $x_{mn}$ . В частных случаях, когда  $x_{mn}$  обладает симметрией относительно одной или двух координат, эти выражения существенно упрощаются.

Пусть  $x_{mn} = x_{m,-n}$ . Тогда  $H_{kl}^{\wedge} = H_{k,-l}^{\wedge}$ ,  $H_{-k,-l}^{\wedge} = H_{-k,l}^{\wedge}$ , а формулы (6.12) и (6.13) приобретают вид:

$$H_{kl} = H_{kl}^{\wedge}$$

$$F_{kl} = \left\{ [H_{kl}^{\wedge} + H_{-k,l}^{\wedge}] - j[H_{k,l}^{\wedge} - H_{-k,-l}^{\wedge}] \right\} / 2$$

Если  $x_{mn} = x_{-m,n} = x_{m,-n}$ , то как преобразование Хартли, так и преобразование Фурье эквивалентны модифицированному преобразованию Хартли:

$$H_{kl} = H_{kl}^{\wedge}; F_{kl} = H_{kl}^{\wedge}$$

## 6.4. Ортогональные преобразования в диадных базисах

Ортогональные преобразования в диадных (или иначе говоря, двузначных знакопеременных) базисах определены для данных, представленных векторами длиной  $N = 2^M$ . К таким преобразованиям относятся преобразования Адамара, Пэли, Уолша, Трахтмана, Качмарджа и ряд других [7,17,21]. Матрица ядра любого из подобных преобразований содержит целочисленные коэффициенты из множества  $\{-1; +1\}$ . Очевидно, что при выполнении подобных преобразований существенно сокращается объем вычислений за счет исключения умножения в каждой базовой операции.

Матрица ядра преобразования Уолша - Адамара для  $N = 2^m$  может быть описана как результат кронекеровского произведения  $m$  матриц ДЭФ  $E_2$  размера  $2 \times 2$  [7,25]:

$$A_N = A_{2^m} = E_2 \otimes E_2 \otimes E_2 \otimes \dots \otimes E_2 = [E_2]^m = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^m, \quad (6.13)$$

где символ  $\otimes$  - операция кронекеровского умножения векторов, в результате чего порождается матрица блочной структуры. Заметим, что операция кронекеровского умножения двух матриц и состоит в получении блочной матрицы, блоками которой является умноженная на соответствующий элемент правой матрицы левая матрица, т.е.:

$$A_4 = E_2 \otimes E_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}$$

аналогично можно получить и для  $N = 8$ :

$$A_8 = E_2 \otimes E_2 \otimes E_2 = A_4 \otimes E_2 = \begin{array}{cc|cc|cc|cc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ \hline 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}$$

Матрица ядра Адамара обладает следующими свойствами:

- 1) цикличность  $a_{N+k} = a_k$ ;  $a_{N-k} = a_{-k}$
- 2) мультипликативность  $a_{k+l} = a_k * a_l$
- 3) симметричность  $A_N = A_N^T$

В задачах ЦОС используются и другие, подобные Адамару, преобразования - Пэли, Уолша, Трахтмана и других. Ядра (матрицы) этих преобразований могут быть получены на основе матрицы ядра преобразования Адамара при определенном переупорядочении их строк. Поэтому перед рассмотрением указанных преобразований рассмотрим правила

переупорядочения матриц, применяемые для формирования ядер таких преобразований.

Переупорядочим элементы вектора  $X = [x_0, x_1, x_2, \dots, x_{N-1}]$  таким образом, чтобы первые элементы вектора были бы чётными, а вторые - нечётными. Повторим эту процедуру до тех пор, умножая каждый раз длину соответствующего вектора вдвое, пока длина подвектора не станет равной 2. В результате получится переупорядоченный вектор, элементы которого упорядочены по закону так называемой двоичной инверсии.

В частности, для  $N=8$  такое переупорядочение будет выполняться следующим образом:

$$X = \begin{array}{|c|} \hline x_0 \\ \hline x_1 \\ \hline x_2 \\ \hline x_3 \\ \hline x_4 \\ \hline x_5 \\ \hline x_6 \\ \hline x_7 \\ \hline \end{array} \Rightarrow \begin{array}{|c|} \hline x_0 \\ \hline x_2 \\ \hline x_4 \\ \hline x_6 \\ \hline x_1 \\ \hline x_3 \\ \hline x_5 \\ \hline x_7 \\ \hline \end{array} \Rightarrow \begin{array}{|c|} \hline x_0 \\ \hline x_4 \\ \hline x_2 \\ \hline x_6 \\ \hline x_1 \\ \hline x_5 \\ \hline x_3 \\ \hline x_7 \\ \hline \end{array} = \hat{X}$$

Вектор  $\hat{X}$  можно получить из  $X$ , если задать следующее правило формирования индекса текущего элемента вектора  $\hat{X}$  из индекса элемента  $x_i$  исходного вектора  $X$ :

- 1) записывается двоичный код текущего индекса  $i$ ;
- 2) двоичный код записывается в обратном порядке как рефлексивный или отраженный код (т.е. начиная с младших разрядов);
- 3) полученный код определяет индекс текущего элемента вектора  $\hat{X}$ ;
- 4) переводим полученный индекс в десятичную систему.

Для рассмотренного выше случая ( $N=8$ ) получим:

$$\begin{array}{|c|} \hline 000 \\ \hline 001 \\ \hline 010 \\ \hline 011 \\ \hline 100 \\ \hline 101 \\ \hline 110 \\ \hline 111 \\ \hline \end{array} \Rightarrow \begin{array}{|c|} \hline 000 \\ \hline 100 \\ \hline 010 \\ \hline 110 \\ \hline 001 \\ \hline 101 \\ \hline 011 \\ \hline 111 \\ \hline \end{array} \Rightarrow \begin{array}{|c|} \hline x_0 \\ \hline x_4 \\ \hline x_2 \\ \hline x_6 \\ \hline x_1 \\ \hline x_5 \\ \hline x_3 \\ \hline x_7 \\ \hline \end{array}$$

Процедура перестановки в терминах векторно-матричных операций может быть записана как:

$$\hat{X} = S_N X, \quad (6.14)$$

где  $S_N$  - перестановочная матрица. Для случая двоично-инверсной перестановки перестановочная матрица будет иметь вид:

$$S_8 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Элементы вектора могут быть также переупорядочены по коду Грея:

$$\begin{vmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{vmatrix} \Rightarrow \begin{vmatrix} 000 \\ 001 \\ 011 \\ 010 \\ 110 \\ 111 \\ 101 \\ 100 \end{vmatrix}$$

В терминах векторно-матричных операций подобная перестановка элементов вектора может быть описана так:

$$\hat{X} = \Gamma_N X,$$

где  $\Gamma_N$  - перестановочная матрица, в частности, для  $N=8$ :

$$\Gamma_8 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{vmatrix}$$

Ядро преобразования Пэли можно получить из ядра преобразования Адамара, переупорядочив строки матрицы  $A_N$  по закону двоичной инверсии, в частности, для  $N=8$  получаем:

$$P_8 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 000 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 001 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 010 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 011 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 100 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 101 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 110 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 111 \end{vmatrix}$$

В общем случае правило перехода от матрицы Адамара к матрице Пэли может быть представлено как [6]:

$$A_N \xrightarrow{\text{Дв.инверсия}} P_N \quad (6.15)$$

Если теперь переупорядочить строки матрицы ядра преобразования Пэли по коду Грея, то получим матрицу ядра преобразования Уолша[8]

$$W_8 = \begin{array}{cccccccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 000 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 001 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 011 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 010 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 110 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 111 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 101 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 100 \end{array}$$

Или в общем виде:

$$\begin{array}{l} A_N \xrightarrow{\text{Дв.инв.}} P_N \xrightarrow{\text{Грей}} W_N \\ P_N \xrightarrow{\text{Грей}} W_N \end{array} \quad (6.16)$$

Очевидно, что свойства матриц  $P_N$  и  $W_N$  аналогичны свойствам матрицы  $A_N$ . Матрицы ядер преобразования Трахтмана и Качмарджа также могут быть получены на основе матрицы ядра Адамара, но при использовании обратной перестановки по коду Грея.

Заметим, что матрицы преобразований в указанных базисах отличаются порядком строк. Очевидно, что при обработке одного и того же вектора исходных данных  $X$  в различных базисах вектор результата будет содержать одинаковые по своей величине элементы, отличаясь для каждого преобразования лишь порядком их следования. Поэтому для всех подобных преобразований можно использовать одну и ту же матрицу ядра (например,  $A_N$ ), а для получения вектора  $F$  с нужным для каждого преобразования порядком следования элементов, переупорядочить элементы исходного вектора  $X$ . Так, для преобразования Пэли необходима двоично-инверсная перестановка, а для преобразования Уолша - двоично-инверсная перестановка и затем перестановка по коду Грея.

Если проанализировать все рассмотренные во второй главе преобразования, то можно прийти к выводу, что их сущность состоит в разложении исходной функции на ряд чётных и нечётных составляющих, которые задаются строками матриц  $E_N$ ,  $H_N$ ,  $A_N$ ,  $W_N$  и  $P_N$ . Так, для матриц  $E_N$  и  $H_N$  это набор гармонических функций  $\cos[...]$  и  $\sin[...]$ , а для  $A_N$ ,  $W_N$  и  $P_N$  - наборы прямоугольных знакопеременных функций.

Тем самым, в зависимости от вида исходной функции, в её составе будут определены отдельные компоненты (и их частотное значение), которые задаются строками матрицы ядра преобразования. Очевидно, что разложение по ортогональным функциям, задаваемым строками матрицы  $E_N$ , позволяет

определить частотный состав исходной функции (сигнала), что имеет вполне понятный физический смысл.

Точно также при разложении по функциям, задаваемым строками матриц  $A_N$ ,  $W_N$  и  $P_N$ , можно определить, какой вклад вносит та или иная знакопеременная функция в состав исходного сигнала.

Очевидно, что исходный сигнал может быть разложен по различным ортогональным составляющим. При этом вклад таких составляющих в исходный сигнал будет различен для разных матриц ортогональных преобразований.

Согласно теореме Коруэна-Лоева [17], может быть определено оптимальное разложение исходной функции по набору ортогональных функций в смысле наименьшего числа отличных от нуля компонент такого разложения.

На практике для сигналов гармонической природы удобно использовать различные гармонические функции, т.е. для определения частотного состава сигнала выполнить преобразование Фурье или Хартли (ДПФ или БПФ). Действительно, для моногармонического сигнала лишь одна компонента разложения по строкам матрицы  $E_N$  будет отлична от "0" (при условии, что  $f_0$  кратно  $k\pi$ ).

Для сигналов, описываемых знакопеременными функциями, близким к оптимальному является разложение по знакопеременным функциям типа Уолша, Адамара. Поэтому в задачах кодирования и распознавания речи, где для представления сигналов широко используется метод широтно-импульсной модуляции, удобно выполнять разложение по ортогональным функциям, задаваемым строками матриц  $A_N$ ,  $W_N$  или  $P_N$ .

## 6.5. Дискретное косинусное преобразование

Одним из дискретных ортогональных преобразований, которое широко используется при сжатии цифровых изображений с потерями, является дискретное косинусное преобразование (ДКП). В частности, на использовании такого преобразования для сжатия изображений ориентирован широко распространенный стандарт JPEG, предложенный в 1985 году [4,22].

Основной этап процедуры сжатия цифровых изображений заключается в преобразовании небольших блоков изображения при помощи двумерного ДКП. Обработка ведется блоками 8x8 пикселей. Выбор ДКП в качестве стандартного решения диктуется следующими причинами:

- Для изображений с сильно коррелированными отсчетами (коэффициент корреляции  $>0,7$ ) эффективность ДКП в смысле компактности представления данных близка к преобразованию Карунена-Лоэва (это преобразование является оптимальным в том смысле, что оно ортонормированно и гарантирует некоррелированность коэффициентов преобразования – элементов яркости изображения  $Y$ ).
- ДКП представляет собой ортогональное сепарабельное преобразование, независящее от изображения, поэтому его вычислительная сложность невелика.



Обработка каждой клетки выполняется независимо и заключается в выполнении ДКП по строкам и столбцам клетки, которое имеет вид:

$$F(u, v) = \frac{c(u, v)}{4} * \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) * \cos\left(\frac{2x+1}{16} * u * \pi\right) * \cos\left(\frac{2y+1}{16} * v * \pi\right) \quad (6.17)$$

В выражении (6.19) множитель  $C(u, v)$  является нормирующим и равен  $1/\sqrt{2}$  при  $u=0$  или  $v=0$  и равен единице для остальных значений индексов.

Общим недостатком дискретных ортогональных преобразований является их высокая вычислительная сложность. В связи с этим используются так называемые быстрые алгоритмы выполнения косинусного преобразования. Известные из литературы алгоритмы быстрых преобразований в базисах косинусных функций, хотя и отличаются меньшим числом операций умножения, но требуют дополнительных перекомпоновок после каждой итерации алгоритма. Одномерное косинусное преобразование может быть вычислено через одномерное преобразование Хартли [2]. При этом вначале производится перестановка элементов вектора исходных данных таким образом, что первую половину последовательности составляют нечетные элементы, а вторую половину последовательности - четные элементы в порядке возрастания номеров, а затем выполняется одномерное преобразование Хартли над модифицированным вектором [3]:

$$H_k = \sum_m X^m \text{cas}[\pi(2\pi km / N)]$$

(где  $\text{cas}[...] = \cos[...]+ \sin[...]$ ), которое может быть вычислено по быстрому алгоритму; после чего выполняются дополнительные вычисления с элементами вектора результатов преобразования Хартли:

$$c^k = \{H_k [\cos(\pi k / 2N) - \sin(\pi k / 2N)] + H_{N-k} [\cos(\pi k / 2N) + \sin(\pi k / 2N)]\} / 2 \quad (6.18)$$

где  $k = [1, \dots, N-1]$ , причем  $c^0 = H_0$ .

Процедура обратного косинусного преобразования отличается от прямого алгоритма другой последовательностью вычислений - вначале выполняются преобразования, подобные выражению (6.20), после чего выполняется преобразование Хартли и лишь затем производится перестановка элементов вектора результатов.

К неудобствам косинусного преобразования следует отнести:

- неразделимость ядра преобразования для двумерного варианта;
- негибкие алгоритмы для различного размера ядра;
- несимметричный алгоритм обратного косинусного преобразования.

Заметим, что строго определяемое двумерное косинусное преобразование не обладает разделимым по координатам ядром, поэтому выполнение двумерного косинусного преобразования в целях сокращения объема вычислений может быть выполнено как модифицированное преобразование строчно-столбцовым методом, подобно тому, как поступают при выполнении двумерного преобразования Хартли [3].

В результате выполнения ДКП формируется 64 частотных компоненты фрагмента (или, что тоже, коэффициентов ДКП). В результате исходный фрагмент представлен в области пространственных частот. Этот шаг еще не приводит к сжатию изображения. Однако, при его выполнении полагается, что в подавляющем большинстве изображений близкие по своим координатам пиксели имеют и близкие значения. Поэтому, при переходе от фрагмента к его частотному представлению большая часть энергии сигнала сосредотачивается в области низких частот, т.е. компоненты с меньшим значением индекса  $k$  в выражении (6.18) имеют большие значения (см. рис. 6.1.).

При выполнении этой операции 64 исходных пикселей преобразуются в матрицу из 64 коэффициентов, которые характеризуют "энергию" исходных пикселей. Важнейшей особенностью этой матрицы коэффициентов является то, что первый коэффициент передает подавляющую часть "энергии", а количество "энергии", передаваемой остальными коэффициентами, очень быстро убывает. То есть, большая часть информации исходной матрицы  $8 \times 8$  пикселей представляется первым элементом матрицы, преобразованной по способу ДКП.

На этом этапе происходит некоторая потеря информации, связанная с принципиальной невозможностью точного обратного преобразования (на этапе восстановления изображения). Однако эта потеря информации весьма незначительна по сравнению с потерями на следующем этапе.

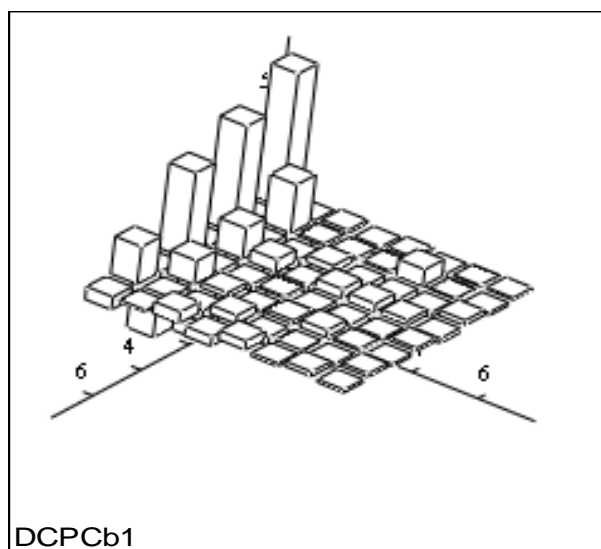


Рис.6.1. Спектр ДКП отдельного фрагмента изображения

Преобразованная матрица из 64 пикселей затем проходит операцию квантования, которая применяется для сокращения разрядности коэффициентов. Процесс квантования, который ведет к сжатию коэффициентов ДКП, выражается следующим образом [4]:

$$z_{kl} = \text{round}(y_{kl}/q_{kl}) = \lfloor (y_{kl} + \lfloor q_{kl}/2 \rfloor) / q_{kl} \rfloor, \quad k, l = 0, 1, \dots, 7,$$

где  $q_{kl}$  - весовой множитель матрицы квантования  $Q$  размера  $8 \times 8$  с номером  $kl$  ( $\lfloor x \rfloor$  обозначает наибольшее целое меньшее или равное  $x$ ).

Иногда для формирования матрицы квантования может использоваться специальная весовая функция, позволяющая сформировать коэффициенты квантования, обращающие в 0 наибольшее число высоко и средне частотных коэффициентов. Согласно литературе [4], элементы матрицы  $Q$  линейно возрастают пропорционально сумме индексов элемента матрицы, например:

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

В результате квантования произошло обнуление многих коэффициентов  $u_{kl}$ . Выбор матрицы  $Q$  определяется требуемым коэффициентом сжатия.

Именно здесь происходит самая значительная потеря информации - отбрасываются малые изменения коэффициентов. Поэтому в процессе восстановления изображения после операции обратного ДКП получаются уже другие параметры пикселей. Квантование также обеспечивает возможность последующего эффективного сжатия данных при помощи любого способа сжатия без потерь.

После квантования компоненты спектра всех обработанных фрагментов «вытягиваются» в последовательность чисел с помощью алгоритма диагонального сканирования. Схема обхода матрицы спектральных коэффициентов фрагмента показана на рис. 6.2. В основе такого сканирования лежит прием, позволяющий достичь большего уплотнения и основывающийся также на характерном виде спектра изображений реальных сцен.

Статистически доказано, что для реальных многоуровневых изображений двумерный квантованный спектр представляет собой матрицу треугольного вида. Большинство значений снизу и справа – нули. Построение элементов матрицы в цепочку производится так, как это показано на рис. 1.6. При этом в последовательность включаются только элементы от первого до последнего ненулевого. После него в последовательность включается специальный стоп – код. Это позволяет исключить из последовательности встречающиеся нули.

Далее обычно применяется метод однопроходного кодирования Хаффмана. Сначала анализируется вся последовательность символов. Часто повторяющимся сериям бит присваиваются короткие обозначения (маркеры). Различие размеров маркеров и представляемых ими битовых серий определяет достигаемую степень сжатия.

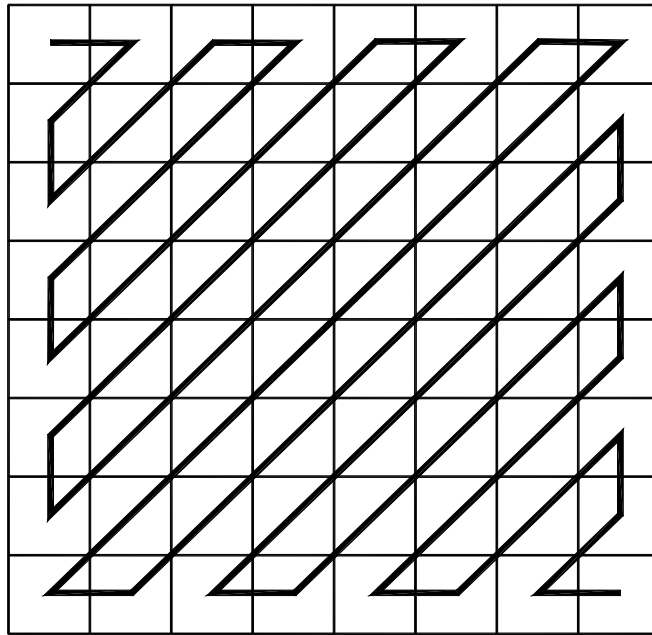


Рис. 6.2. Диагональное «зиг-заг» сканирование спектральных компонент

**Восстановление.** При восстановлении изображений перечисленные выше шаги выполняются в обратном порядке. Декодирование начинается с восстановления из полученного битового потока закодированных неравномерным кодом длин серий нулей и значащих элементов матрицы  $Z$ . Восстановление коэффициентов разложения  $Y^{\wedge}$  по квантованным значениям  $Z$  выполняется по формуле

$$y^{\wedge}_{kl} = z_{kl} q_{kl} \quad (6.19)$$

Далее выполняется обратное ДКП:

$$f(x, y) = \frac{1}{4} * \sum_{u=0}^7 \sum_{v=0}^7 c(u, v) * F(u, v) * \cos\left(\frac{2x+1}{16} * u * \pi\right) * \cos\left(\frac{2y+1}{16} * v * \pi\right)$$

После этого цветовое пространство данных изображения можно преобразовать в исходный вид. Потери при обратном ДКП также не велики по сравнению с потерями квантования.

Коэффициент архивации в JPEG может изменяться в пределах от 2 до 200 раз (на практике коэффициент сжатия не превосходит 20-25) [4].

### 6.6. Оконное преобразование Фурье

Главный недостаток ДПФ и ДКП заключается в интегральной оценке всех частотных составляющих спектра вне зависимости от времени их существования. Поэтому ДПФ и ДКП не годятся для нестационарных сигналов, у которых определённые частотные компоненты существуют только в определённые промежутки времени [11,17,19].

Проблемы спектрального анализа ограниченных во времени сигналов частично решаются с помощью оконного преобразования Фурье [11].

Идея данного преобразования заключается в разбиении временного интервала на ряд промежутков – окон. Для каждого из окон вычисляется своё Фурье преобразование. Таким образом, можно перейти к частотно-временному представлению сигнала [17].

При выполнении оконного Фурье-преобразования по методу Бартлетта весь интервал наблюдения сигнала из  $N$  отсчетов сигнала разбивается на  $K$  неперекрывающихся выборок по  $M$ -отсчётов в каждой и последующего их усреднения (рис. 6.3).

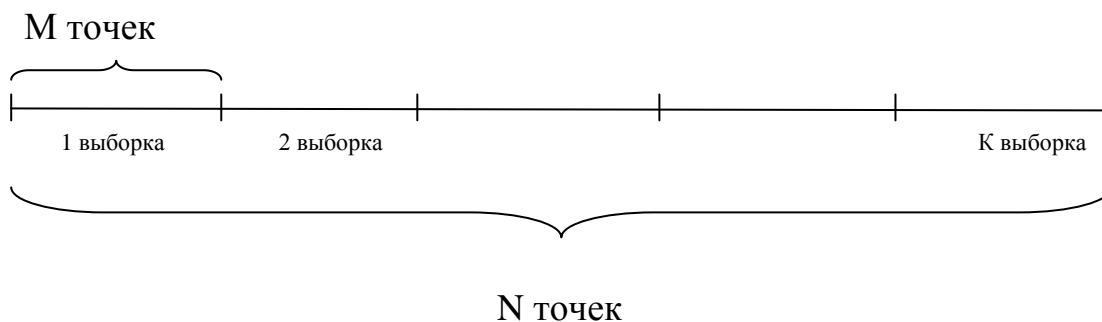


Рис.6.3. Разбиения эргодического сигнала на окна по методу Бартлетта

Известен метод Уэлша, который основан на несколько модифицированном методе сегментирования за счёт применения окна данных и использования перекрывающихся сегментов. Вычисление спектральной плотности каждого сегмента (периодограммы или сонограммы) предваряется взвешиванием этих данных на оконную функцию. Это позволяет за счёт небольшого ухудшения разрешения ослабить эффекты, обусловленные боковыми лепестками и уменьшить смещение оценок. С помощью перекрытия сегментов получается увеличить число усредняемых периодограмм при заданной длине записи данных и тем самым уменьшить дисперсию оценки спектральной плотности мощности (рис. 6.4.).

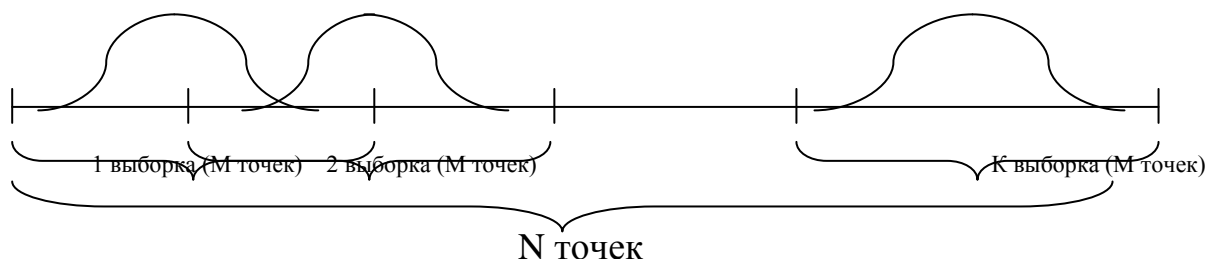
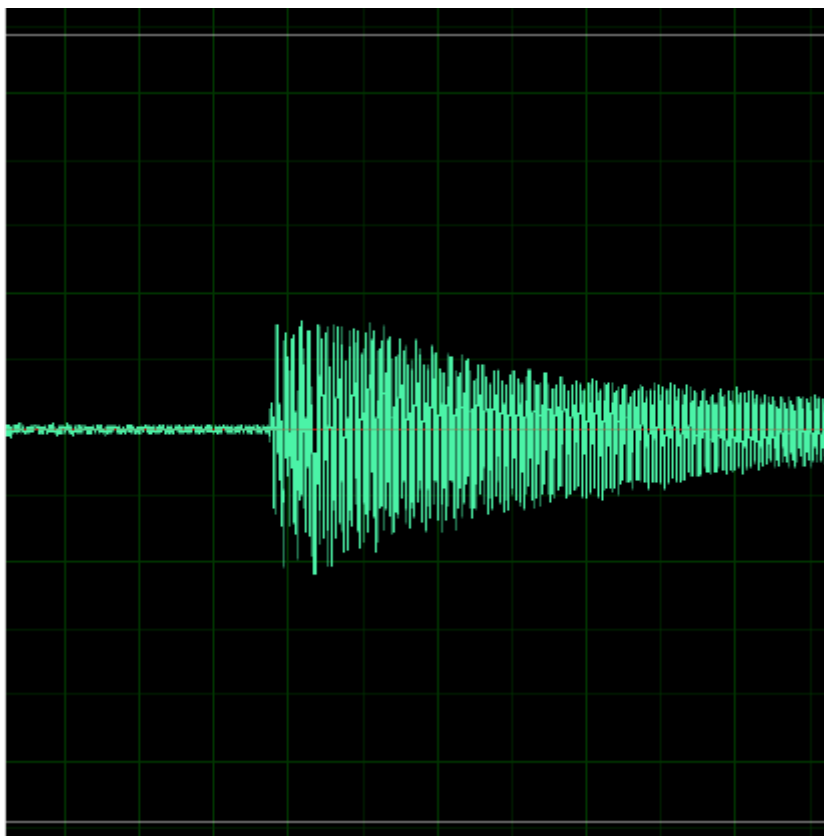


Рис. 6.4. Усреднение спектральных оценок по методу Уэлша

На рис. 6.5, а) предствалена в качестве иллюстрация временная диаграмма колебания струны, а на рис. 6.5, б) – сонограмма, т.е зависимость амплитуды от частоты и от времени, построенная как результат оконного преобразования Фурье по методу Уэлша.

а)



б)

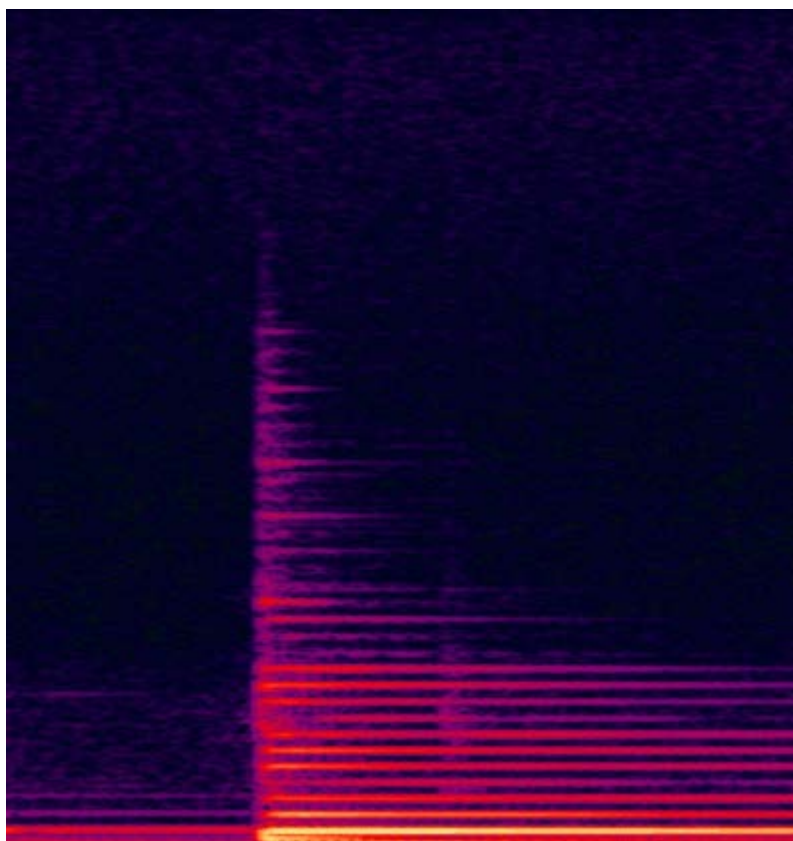


Рис. 6.5. Временная диаграмма колебания струны (а) и ее сонограмма (б).

При вычислении оконного ДПФ по методу Уэлча имеется возможность существенного сокращения вычислительных затрат за счет оптимизации вычислений.

Основная идея оптимизации вычислений заключается в выявлении зависимостей между результирующими отсчетами с одинаковыми порядковыми номерами с целью использования предыдущих значений отсчетов для расчета текущих. Очевидно, что полученные зависимости позволят выполнять преобразования тем быстрее по сравнению с традиционными методами, чем больше размер окна.

ДПФ в матричной форме выглядит следующим образом:

$$e^{-j\frac{2\pi}{N}kn} = W^{kn}$$

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W^{1 \times 1} & W^{1 \times 2} & \dots & W^{1 \times (N-1)} \\ 1 & W^{2 \times 1} & W^{2 \times 2} & \dots & W^{2 \times (N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & W^{(N-1) \times 1} & W^{(N-1) \times 2} & \dots & W^{(N-1) \times (N-1)} \end{pmatrix} \times \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_{N-1} \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \dots \\ f_{N-1} \end{pmatrix}$$

В начале выполняется ДПФ для отсчетов  $x_0 \dots x_{N-1}$ :

$$\begin{aligned} f_0^0 &= x_0 + x_1 + x_2 + \dots + x_{N-1} \\ f_1^0 &= x_0 + x_1 W^{1 \times 1} + x_2 W^{1 \times 2} + \dots + x_{N-1} W^{1 \times (N-1)} \\ f_2^0 &= x_0 + x_1 W^{2 \times 1} + x_2 W^{2 \times 2} + \dots + x_{N-1} W^{2 \times (N-1)} \\ f_{N-1}^0 &= x_0 + x_1 W^{(N-1) \times 1} + x_2 W^{(N-1) \times 2} + \dots + x_{N-1} W^{(N-1) \times (N-1)} \end{aligned} \quad (6.20)$$

Далее, при сдвиге окна на один отсчет ДПФ досчитывается на основе предыдущего результата:

$$\begin{aligned} f_0^1 &= x_1 + x_2 + x_3 + \dots + x_N = (f_0^0 - x_0) W^{-0} + x_N W^{0 \times (N-1)} \\ f_1^1 &= x_1 + x_2 W^{1 \times 1} + x_3 W^{1 \times 2} + \dots + x_N W^{1 \times (N-1)} = (f_1^0 - x_0) W^{-1} + x_N W^{1 \times (N-1)} \\ f_2^1 &= x_1 + x_2 W^{2 \times 1} + x_3 W^{2 \times 2} + \dots + x_N W^{2 \times (N-1)} = (f_2^0 - x_0) W^{-2} + x_N W^{2 \times (N-1)} \\ f_{N-1}^1 &= x_1 + x_2 W^{(N-1) \times 1} + x_3 W^{(N-1) \times 2} + \dots + x_N W^{(N-1) \times (N-1)} = (f_{N-1}^0 - x_0) W^{-(N-1)} + x_N W^{(N-1) \times (N-1)} \end{aligned} \quad (6.21)$$

Так как  $W^{kn}$  является периодической последовательностью<sup>1</sup> с периодом  $N$ , то можно записать:

$$f_k^1 = (f_k^0 - x_0) W^{-k} + x_N W^{k \times (N-1)} = (f_k^0 - x_0) W^{-k} + x_N W^{-k} = (f_k^0 - x_0 + x_N) W^{-k}, \quad k = \overline{0, N-1} \quad (6.22)$$

Для следующего шага формула будет выглядеть:

$$f_k^2 = (f_k^1 - x_1 + x_{N+1}) W^{-k} = (f_k^1 - x_1 + x_{N+1}) e^{j\frac{2\pi}{N}k}, \quad k = \overline{0, N-1} \quad (6.23)$$

Следовательно, общая формула имеет вид:

$$f_k^l = (f_k^{l-1} - x_{l-1} + x_{N+l-1}) W^{-k} = (f_k^{l-1} - x_{l-1} + x_{N+l-1}) e^{j\frac{2\pi}{N}k}, \quad k = \overline{0, N-1}, \quad (6.24)$$

где  $l$  – номер этапа вычисления кратковременного ДПФ при сдвиге окна на один отсчет.

Формула для обратного кратковременного ПФ имеет вид:

$$x_k^l = (x_k^{l-1} - f_{l-1} + f_{N+l-1}) W^k = (x_k^{l-1} - f_{l-1} + f_{N+l-1}) e^{-j\frac{2\pi}{N}k}, \quad k = \overline{0, N-1}, \quad (6.25)$$

<sup>1</sup> Т.е.  $W^{(k+mN)(n+lN)} = W^{kn}$ ,  $m, l = 0, \pm 1, \pm 2, \dots$

где  $l$  – номер этапа вычисления кратковременного ДПФ при сдвиге окна на один отсчет.

Во время вычисления кратковременного ДПФ при сдвиге окна необходимо выполнить только  $N$  базовых операций. Вычислительная сложность этого алгоритма  $Q_{\text{кДПФ}} = N \text{ БО}^2$ .

### 6.7. Выполнение фильтрации в частотной области

Идея частотной фильтрации основана на отличии спектров полезного сигнала и помехи [5,16,17]. При этом используются линейные частотные фильтры, позволяющие подавлять помеху и улучшать тем самым соотношение сигнал/помеха. Параметры фильтра определяются спектральными характеристиками сигнала и помехи. На практике наиболее часто встречаются следующие случаи.

1) На вход фильтра поступает узкополосный сигнал и широкополосная помеха. В этом случае эффективен узкополосный фильтр с полосой пропускания  $\Delta\omega_x$ ;

2) На вход фильтра поступает широкополосный сигнал и узкополосная помеха с шириной спектра  $\Delta\omega_r$ . Для подавления подобной помехи фильтр должен обеспечивать подавление помехи в полосе  $\Delta\omega_r$ ;

3) На вход фильтра поступают периодический сигнал и широкополосная помеха.

Рассмотрим случай, когда полезный сигнал является гармоническим, а помеха типа белого шума. Для выделения полезного сигнала в этом случае должен быть использован узкополосный фильтр, настроенный на частоту сигнала. Отношение мощности сигнала к мощности помехи на выходе фильтра при этом

$$\left( \frac{P_x^*}{P_r} \right)_{\text{полн}} = \frac{(P_x)_{\text{полн}}}{P_0 \Delta\omega_{\text{тм}}} = \frac{(P_x)_{\text{полн}}}{2\pi \Delta f_{\text{тм}} P_0}, \quad (6.26)$$

где  $P_0$  - средняя мощность помехи, приходящаяся на единицу полосы;  $\Delta\omega_{\text{ф}}$  - полоса пропускания фильтра. Как видно из выражения (6.26), отношение  $\frac{P_x^*}{P_r}$  можно сделать сколь угодно большим за счет уменьшения полосы пропускания фильтра  $\Delta f_{\text{ф}}$ .

В реальных условиях полезный сигнал поступает лишь в течении определенного времени  $T_x$  и, следовательно, его спектр неограничен.

Известно, что практическая ширина спектра такого сигнала связана с его длительностью соотношением

$$\Delta f_{\text{ф}} T_x = \mu, \quad (6.27)$$

где  $\mu$  - постоянная, зависящая от формы сигнала. Обычно принимается  $\mu \approx 1$ .

<sup>2</sup> Второе и следующие окна.



Длительность сигнала  $T_x$  должна быть выбрана такой, чтобы его спектр был не шире полосы пропускания фильтра  $\Delta f_x \leq \Delta f_\phi$ .

Подставляя в (6.27) вместо  $\Delta f_\phi$  величину  $\Delta f_x$ , получаем

$$\left( \frac{P_x^*}{P_r} \right)_{\text{ВЫХ}} = \frac{(P_x)_{\text{ВХ}}}{P_0 \Delta \omega_\phi} = \frac{(P_x)_{\text{ВХ}}}{2\pi P_0 \mu} T_x. \quad (6.28)$$

Формула (6.28) показывает, что увеличение отношения сигнал/помеха достигается за счет увеличения длительности сигнала  $T_x$ , т.е. времени наблюдения.

Таким образом, при частотной фильтрации улучшение отношения сигнал/помеха окупается ценой увеличения времени наблюдения сигнала.

Кроме указанных видов фильтров в частотной области может выполняться линейная фильтрация более общего вида, описываемая выражением типа свертки. Схема выполнения фильтрации сигнала в частотной области приведена на рис. 6.6. Обобщенно подобный метод частотной фильтрации иногда называют операционной частотной фильтрацией. Таким методом можно выполнять заданные математические преобразования исходного сигнала, например, дифференцирование или интегрирование [14, 25].

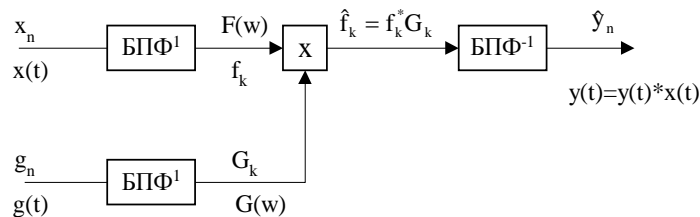


Рис. 6.6. Схема операционной частотной фильтрации

**Согласованные фильтры** предназначены для выделения сигналов известной формы на фоне шумов. Критерием оптимальности таких фильтров является получение на выходе максимально возможного отношения амплитудного значения сигнала к действующему значению помехи [25]. Реакция согласованного фильтра эквивалентна действию корреляционного приемника. Схема выполнения согласованной фильтрации приведена рис. 6.7.

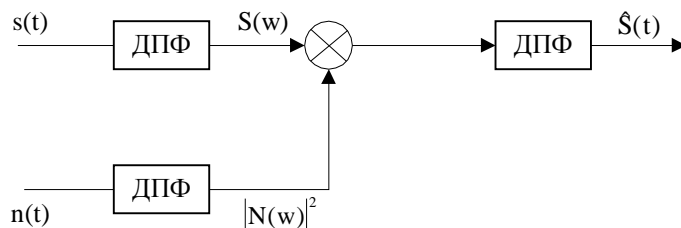


Рис. 6.7. Схема выполнения согласованной фильтрации

Для выделения известного сигнала  $S(t)$  из стационарного шума  $n(t)$  оптимальным является фильтр с передаточной функцией [17, 25]:

$$H(\omega) = \frac{S^*(\omega)}{|N(\omega)|^2} \quad (6.29)$$

$S^{(*)}(\omega)$  - сопряженный спектр Фурье ( или Фурье образ ) исходного сигнала,  $|N(\omega)|^2$  - спектральная плотность шума,  $\omega=2\pi\nu$ . Этот фильтр носит название согласованного фильтра.

Если нужно минимизировать среднеквадратичную ошибку восстановленного сигнала, то:

$$H(\omega) = \frac{1}{F(\omega)} \cdot \frac{\varphi_0 \varphi_n^{-1}}{\varphi_0 \varphi_n^{-1} + |F(\omega)|^{-2}} \quad (6.30)$$

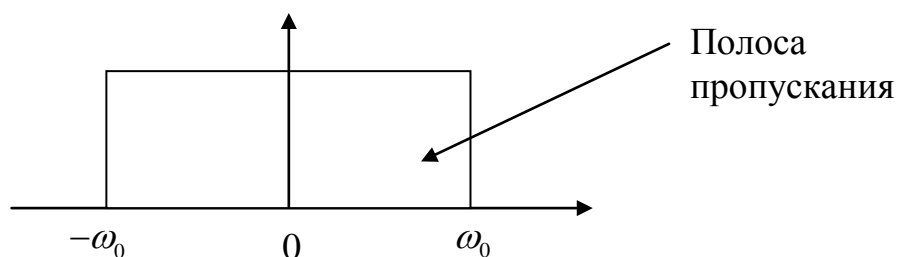
где  $F(\omega)$ - спектр сигнала;  $\varphi_0$  - спектр мощности восстанавливаемого сигнала,  $\varphi_n$  - спектр мощности шума.

При отсутствии шума для минимизации среднеквадратичной погрешности используется инверсный фильтр [28]:

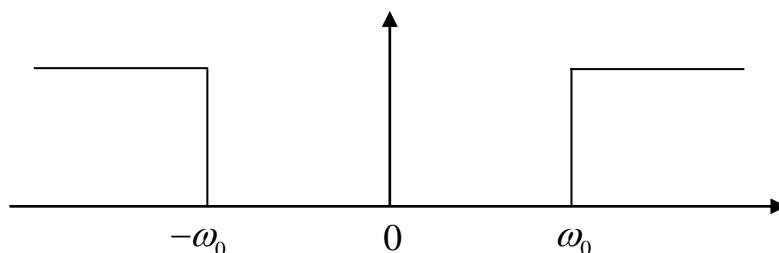
$$H(\omega) = \frac{1}{F(\omega)} = \frac{F^*(\omega)}{|F(\omega)|^2} \quad (6.31)$$

### Виды фильтров

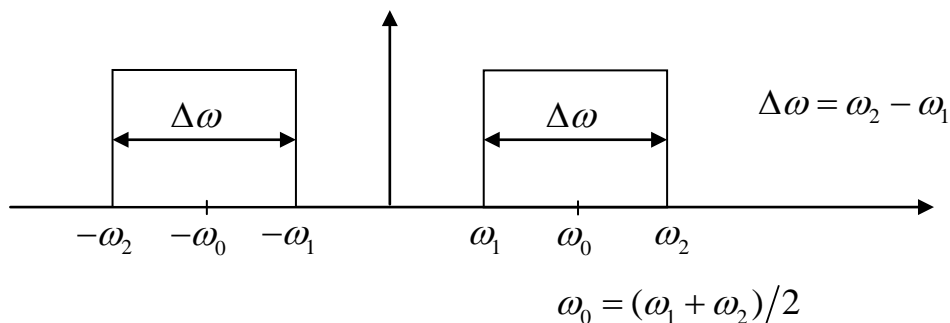
#### 1) Фильтры нижних частот (ФНЧ) low-pass filter



#### 2) Фильтры верхних частот (ФВЧ) high-pass filter

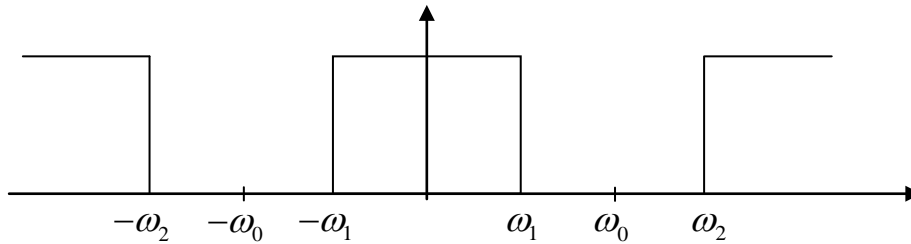


#### 3) Полосовые фильтры (ПФ) band-pass filter



#### 4) Режекторные фильтры (ПФ) band-stop filter

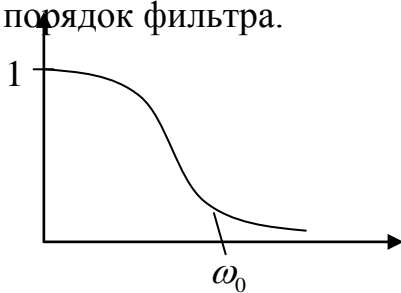
Пропускают все частоты, кроме узкой полосы частот.



**Фильтр Баттерворта:**

$$K(\omega) = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_0}\right)^{2n}}},$$

где  $\omega_0$  – частота среза (у прототипа  $\omega_0 = 1 \text{ рад/с}$ ;  
 $n$  – порядок фильтра.



**Фильтр Чебышева 1-го рода:**

$$K(\omega) = \frac{1}{\sqrt{1 + \varepsilon^2 T_n^2(\omega/\omega_0)}},$$

где  $\omega_0$  – частота среза;  $T_n(x)$  – полином Чебышева  $n$ -го порядка;  $n$  – порядок фильтра;  $\varepsilon$  – параметр, определяющий величину пульсации АЧХ в полосе пропускания.

#### 6.6.1 Алгоритм Герцеля.

Дискретное преобразование Фурье (ДПФ) используется для преобразования сигнала из временной области в частотную. С другой стороны, ДПФ может использоваться для вычисления нескольких частотных точек, например 20, 25 и 30 точек из 256 возможных. Обычно, если необходимо рассчитать более чем  $\log_2 N$  точек из  $N$ , то быстрее рассчитать БПФ, а затем исключить ненужные точки. Если необходимо рассчитать несколько точек, то ДПФ быстрее.

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad \text{где } k = 0, 1, \dots, N-1 \text{ и } W_N^{nk} = e^{-j(2\pi/N)nk}$$

ДПФ вычисляется для одной точки из  $N$ , например, с номером 15:

$$X(15) = \sum_{n=0}^{N-1} x(n)W_N^{15n} \quad \text{где } k = 15 \text{ и } W_N^{15n} = e^{-j(2\pi/N)15n} \quad 0 \leq k \leq N-1$$

Использование алгоритма Герцеля сокращает количество операций и экономит время [11,20]. Для вычисления ДПФ, необходимо вычислить большое количество комплексных коэффициентов. Для ДПФ размером  $N$  используется  $N^2$  комплексных коэффициентов. При использовании алгоритма Герцеля понадобится всего два коэффициента для каждой частоты: один вещественный и один комплексный.

Алгоритм Герцеля можно алгебраически модифицировать, так что результат будем брать в квадрате (тем самым избавляясь от комплексной составляющей). Такая модификация исключает фазовую составляющую, которая не используется во многих реальных приложениях. Так, например, алгоритм Герцеля широко используется для распознавания DTMF сигналов в телефонии. К достоинствам данной модификации можно отнести наличие только одного вещественного коэффициента.

Алгоритм Герцеля позволяет обрабатывать данные в темпе их поступления, при этом нет необходимости ждать, пока заполнится буфер из  $N$  элементов. Схема такого преобразования представлена на рис. 6.8.

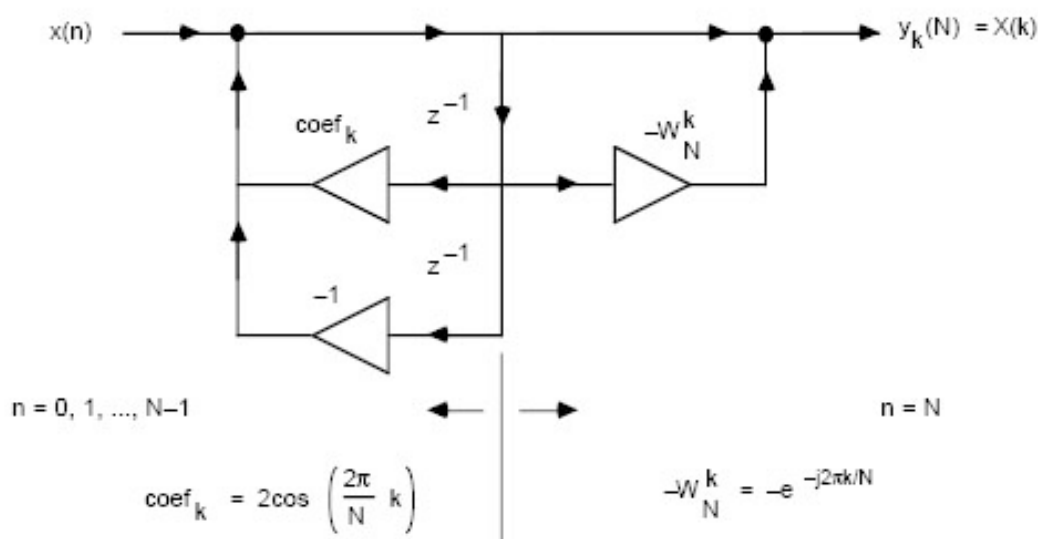


Рис. 6.8 Структурная схема фильтра, реализующего алгоритм Герцеля

Фильтр, реализующий подобный алгоритм может быть представлен, как БИХ фильтр второго порядка.

Алгоритм Герцеля может быть использован для подсчета ДПФ. Однако, его реализация имеет много общего с фильтрами. ДПФ или БПФ получают результат размерности  $N$  из исходных данных размерности  $N$ . Но фильтры БИХ и КИХ получают новое значение на выходе, как только получают новые данные на входе. Алгоритм вычисляет новое значение  $y_k(n)$  (см. рис.), для каждого нового  $x(n)$ . Результат вычисления ДПФ  $X(k)$  будет равен  $y_k(n)$ , если  $n = N$ . Так как каждое новое значение  $y_k(n)$  (где  $n \neq N$ ) не ведет к получению конечного результата  $X(k)$ , нет необходимости вычислять  $y_k(n)$  до тех пор пока  $n = N$ . Это подразумевает, что алгоритм Герцеля функционально эквивалентен БИХ

фильтру второго порядка за исключением того, что выходной результат фильтра появляется только после  $N$  отсчетов данных.

В фильтре Герцеля Вычисления можно выделить две части - правую (рис.6.9) и левую (рис. 6.10).

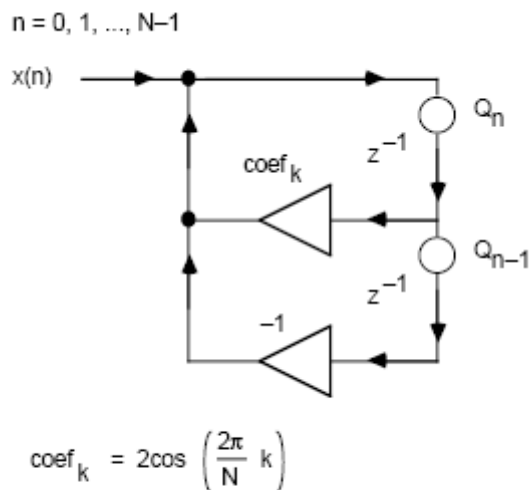


Рис. 6.9. Левая часть фильтра, реализующего алгоритм Герцеля.

В этой схеме при выполнении вычислений вида

$$Q_k(n) = coef_k \times Q_k(n-1) - Q_k(n-2) + x(n)$$

два промежуточных значения  $Q_k(-1) = 0$ ;  $Q_k(-2) = 0$ ;  $n = 0, 1, 2, \dots, N-1$  хранятся в памяти, а  $coef_k = 2 \cos(2\pi k / N)$ ;

Для каждого нового отсчета  $x(n)$   $Q(n-1)$  и  $Q(n-2)$  считываются из памяти данных и используются для вычисления нового значения  $Q(n)$ .

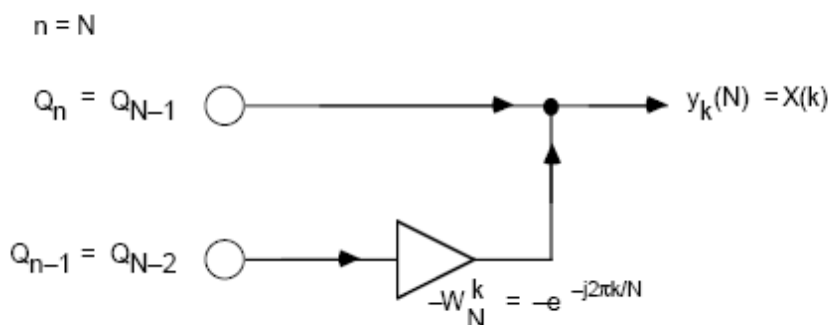


Рис. 6.10. Правая часть схемы фильтра, реализующего алгоритм Герцеля.

## 7. ВЕЙВЛЕТ ПРЕОБРАЗОВАНИЯ ИЛИ РАЗЛОЖЕНИЕ ПО ВСПЛЕСКАМ

Из формулы прямого преобразования Фурье вытекает главный его недостаток: интегральная оценка всех частотных составляющих спектра вне зависимости от времени их существования. Преобразование Фурье прекрасно подходит для стационарных сигналов, но не годится для нестационарных, у которых определённые частотные компоненты существуют только в определённые промежутки времени.

Поэтому можно выделить следующие недостатки преобразования Фурье:

- неприменимость к анализу нестационарных сигналов,
- преобразование Фурье для одной заданной частоты требует знание сигнала в прошлом и будущем,
- из-за неизбежного ограничения числа гармоник или спектра – невозможно точное восстановление сигнала,
- отдельные особенности сигнала: разрывы или пики – вызывают незначительное изменение спектрального образа сигнала во всём интервале частот,
- по составу высших составляющих спектра практически невозможно оценить местоположение особенностей на временной зависимости сигнала.

Проблемы спектрального анализа ограниченных во времени сигналов частично решаются с помощью оконного преобразования Фурье [17].

Идея данного преобразования заключается в разбиении временного интервала на ряд промежутков – окон. Для каждого из окон вычисляется своё Фурье преобразование. Таким образом, можно перейти к частотно-временному представлению сигнала.

Оконное преобразование можно описать следующим образом:

$$A(\omega) = \int_{-\infty}^{+\infty} y(t) \cdot w(t-b) \cdot e^{-i\omega t} dt, \quad (7.1)$$

где  $w(t-b)$  – оконная функция, а параметр  $b$  задаёт сдвиг по временной оси.

У оконного преобразования Фурье есть ряд ограничений: выбирая окно с малой шириной по времени – получают высокое временное разрешение, но низкое частотное, и наоборот, взяв широкое временное окно, получают хорошее частотное разрешение, но малое временное.

Другим видом преобразований сигнала, не обладающих ограничениями, присущими оконному преобразованию Фурье, являются вейвлет преобразования [1].

## 7.1. Понятие о Wavelet-преобразованиях. Преобразование Хаара

В ряде случаев оказывается более удобным в качестве базисов разложения использовать такие системы функций, для которых коэффициенты разложения учитывают поведение исходной функции лишь в нескольких близкорасположенных точках.

Использование такого базиса по своей сути означает переход от частотного анализа к масштабному, т.е. функция  $f(x)$  анализируется с помощью некоторой “стандартной” математической функции, изменяемой по масштабу и сдвигу на некоторую величину.

Первое упоминание об этих функциях появилось в работах Хаара в 1909 году. В 30-е годы начались более детальные исследования возможностей представления сигналов с использованием базисных масштабируемых функций. Пол Леви, используя масштабируемую базисную функцию типа функции Хаара, исследовал разновидность случайного сигнала - броуновское движение. Он обнаружил преимущество в применении базисных функций Хаара перед функциями Фурье.

В период 60-х - 80-х годов Вейс и Кофман исследовали простейшие элементы функционального пространства, названные ими атомами, с целью обнаружить атомы для произвольной функции и найти "правила сборки", позволяющие реконструировать все элементы функционального пространства, используя эти атомы. В 1980 году Гроссман и Марлет определили такие функции как Wavelet-функции. В переводе с английского Wavelet – всплеск, поэтому в отечественной литературе встречается термин «разложение по всплескам» наряду с вейвлет-анализом.

В конце 80-х г.г. Мейер и Добичи на основе исследований Марлета создали ортогональные базисы Wavelet-функций, которые и стали основой современных Wavelet-функций. Сходство Wavelet и Фурье преобразований состоит в следующем:

1. Оба они являются линейными преобразованиями и предназначены для обработки блоков данных, содержащих  $\log_2 N$  элементов.
2. Обратные матрицы ДПФ и DWT (discret wavelet transform) равны транспонированной, причем строки самих матриц содержат функции  $\cos(x)$  и  $\sin(x)$ , а для DWT - более сложные базисные функции - wavelet.

Наиболее важное различие между этими двумя видами преобразований состоит в том, что отдельные функции wavelet локализованы в пространстве, а синусные и косинусные - нет. Благодаря этой особенности, DWT находит большое число применений, в том числе для сжатия данных, распознавания образов и подавления шумовой составляющей принимаемого сигнала.

Преобразование Wavelet состоит из неограниченного набора функций. Семейство Wavelet различают по тому, насколько компактны базисные функции в пространстве и насколько они гладки. Некоторые из них имеют фрактальную структуру. В каждом семействе они могут быть разбиты на подклассы по числу коэффициентов и уровню итераций. Чаще всего внутри семейства функция классифицируется по номеру моментов исчезновения.

Набор дискретных Wavelet-функций в общем виде может быть описан как

$$W_{[s,l]}(x) = 2^{-s/2} W(2^{-s}, x-l), \quad (7.2)$$

где  $s$  и  $l$  - целые числа, которые масштабируют и сдвигают материнскую функцию  $W(x)$  для создания wavelet.

Индекс масштаба  $s$  показывает ширину wavelet, а индекс смещения  $l$  определяет ее позицию. Материнские функции масштабированы или растянуты коэффициентом, кратным степени 2 и приведены к целому. Таким образом, если известна материнская функция, то может быть построен и весь базис.

В свою очередь, само Wavelet-преобразование может быть записано в виде [1]:

$$V(x) = \sum_{k=l}^{N-2} (-1)^k z_{k+l} W(2x+l) \quad (7.3)$$

где  $Z_{k+l}$  – отсчеты исходного сигнала.

Метод разложения по всплескам широко используется для выделения шумовой компоненты при обработке данных.

К wavelet-подобным функциям относятся функции Хаара [6]. Для  $N=4$  и  $N=8$  матрицы ядра преобразования имеют вид:

$$X_4 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{vmatrix}$$

$$X_8 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{vmatrix}$$

Основные свойства матрицы ядра преобразования Хаара состоят в следующем:

а) ее элементы не мультипликативны;

б) матрица не симметрична, откуда следует, что для обратного преобразования матрицу ядра необходимо транспонировать, т.е.

$$X_N^{-1} = X_N^T$$

в) строки матрицы определяют периодические функции с периодом  $N$ .

Однако матрица ядра преобразования Хаара является не ортонормированной, т.е. для данного преобразования не выполняется теорема Парсеваля. Поэтому для выполнения теоремы Парсеваля требуется ввести дополнительную нормировку, такая нормировка заключается в умножении на



$\sqrt{2}$  тех элементов вектора результатов, которые соответствуют строкам матрицы с нулевыми компонентами. Для строк, содержащих  $N/2$  ненулевых элемента такое умножение выполняется один раз, для строк с  $N/4$  ненулевыми элементами - два раза и так далее. Подобную нормировку необходимо выполнять как при выполнении прямого, так и обратного преобразования Хаара.

Близким к преобразованию Хаара является усеченное преобразование Адамара, отличающееся, по своей сути, лишь порядком следования строк матрицы ядра преобразования [6].

## 7.2. Вейвлеты

В математической теории сигналов принято считать, что сигналы определены как векторы в некотором пространстве  $V$ . Бесконечно размерное пространство, часто используемое в теории вейвлетов, называется гильбертовым пространством  $L^2[R]$ . Вейвлет функции  $\Psi$ , принадлежащие  $L^2[R]$  должны иметь нулевое среднее значение и затухать на бесконечности. Ввиду ограниченности действия вейвлетов, они могут покрывать всю вещественную ось, если обладают возможностью сдвига по этой оси, а также свойством масштабирования, которое можно уподобить изменению частоты гармоник в рядах Фурье. Обладая этими свойствами, вейвлеты позволяют представить локальные особенности сигналов [1].

На основании понятий о векторном пространстве общим подходом к анализу сигналов  $s(t)$  стало их представление в виде взвешенной суммы простых составляющих – базисных функций  $\Psi_k(t)$ , умноженных на коэффициенты  $C_k$ :

$$s(t) = \sum_k C_k \Psi_k(t) \quad (7.4)$$

Термин вейвлет, введён впервые Морле, в переводе с английского означает «короткая волна» или «всплеск». Грубо вейвлеты можно представить как некоторые волновые функции, способные осуществлять преобразование Фурье не по всей временной оси, а локально по месту своего расположения. Базисными функциями вейвлетов могут быть различные функции, в том числе, напоминающие модулированные импульсами синусоиды, функции со скачками уровня и т.д. Это обеспечивает лёгкое представление сигнала со скачками и разрывами.

Вейвлеты характеризуются своими временными и частотными образами. Временной образ представляет собой некую функцию  $\Psi(t)$  времени, а частотный образ определяется её Фурье-образом:  $\hat{\Psi}(w) = F(w)$ .

Таким образом, с помощью вейвлетов сигнал представляется совокупностью волновых пакетов, образованных на основе некоторой исходной базовой функции  $\Psi_0(t)$ . Эта совокупность, разная в различных частях

временного интервала определения сигнала и корректируемая множителями, и представляет с той или иной степенью детализации. Такой подход называют вейвлет анализом сигналов.

Число используемых вейвлетов, при разложении сигнала, задаёт уровень декомпозиции сигнала. За нулевой уровень декомпозиции часто принимают сам сигнал, а последующие уровни декомпозиции образуют ниспадающее вейвлет дерево.

Одна из основополагающих идей вейвлет-представления вейвлет представлении сигнала заключается в разбивке приближения сигнала на две составляющих: грубую (*аппроксимирующую*) и приближенную (*детализирующую*), с последующим их уточнением итерационным методом. Каждый шаг такого уточнения соответствует определённому уровню декомпозиции и реставрации сигнала.

Как и преобразование Фурье, вейвлет преобразования можно применять как к непрерывным сигналам – непрерывные вейвлет преобразования, так и к цифровым сигналам – дискретные вейвлет преобразования.

### 7.2.1. Непрерывные вейвлет преобразования

В основе непрерывного вейвлет преобразования *НВП* лежит использование двух непрерывных и интегрируемых по всей оси  $x$  функций:

- вейвлет функция *psi*  $\Psi(t)$  с нулевым значением интеграла ( $\int_{-\infty}^{\infty} \Psi(t) dt = 0$ ), определяющая детали сигнала и порождающая детализирующие коэффициенты;
- масштабирующая или скейлинг-функция *phi*  $\phi(t)$  с единичным значением интеграла ( $\int_{-\infty}^{\infty} \phi(t) dt = 1$ ), определяющая грубое приближение сигнала и порождающая коэффициенты аппроксимации.

$\Phi$ -функции  $\phi(t)$  присущи не всем вейвлетам, а только тем, которые являются ортогональными.

$\Psi$ -функции  $\Psi(t)$  создаются на основе той или иной базисной функции  $\Psi_0(t)$ , которая как и  $\Psi(t)$  определяет тип вейвлета. Базисная функция должна удовлетворять всем требованиям, которые были отмечены для  $\psi$ -функции. Она должна обеспечивать выполнение двух основных операций:

- смещение по оси времени:  $\Psi_0(t-b)$  при  $b \in \mathfrak{R}$ ;
- масштабирования:  $a^{-1/2} \Psi_0(\frac{t}{a})$  при  $a \in \mathfrak{R}^+ - \{0\}$ .

Параметр  $a$  задаёт ширину данного вейвлета, а  $b$  – его положение:

$$\Psi(t) \equiv \Psi(a, b, t) = a^{-1/2} \Psi_0\left(\frac{t-b}{a}\right) \quad (7.5)$$

Вейвлеты, обозначаемые как  $\Psi(t)$ , называют материнскими вейвлетами, поскольку они порождают целый ряд вейвлетов определённого рода.

Прямое непрерывное вейвлет преобразование (ПНВП) сигнала  $s(t)$  задаётся путём вычисления вейвлет коэффициентов по формуле:

$$C(a, b) = \langle s(t), \psi(a, b, t) \rangle = \int_{-\infty}^{\infty} s(t) a^{-1/2} \psi\left(\frac{t-b}{a}\right) dt, \quad (7.6)$$

где обозначение  $\langle \dots, \dots \rangle$  означает скалярное произведение соответствующих множителей. С учётом ограниченной области определения сигналов и  $a, b \in \mathfrak{R}$ ,  $a \neq 0$  формула (7.6) приобретает вид:

$$C(a, b) = \int_{\mathfrak{R}} s(t) a^{-1/2} \psi\left(\frac{t-b}{a}\right) dt \quad (7.7)$$

Прямое вейвлет преобразование можно рассматривать как разложение сигнала по всем возможным сдвигам и растяжением/сжатиям сигнала  $\Psi(t)$ , при этом параметры  $a$  и  $b$  могут принимать любые значения в области их определения.

Обратное непрерывное вейвлет преобразование осуществляет по формуле реконструкции во временной области:

$$f = C_{\psi}^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{dadt}{a^2} \langle f, \psi^{a,b} \rangle \psi^{a,b}, \quad \text{где} \quad (7.8)$$

$$\psi^{a,b}(x) = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right)$$

$C_{\psi}$  зависит только от  $\Psi$  и определяется как:

$$C_{\psi} = 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(w)|^2}{|w|^{-1}} dw, \quad \text{где } \hat{\psi}(w) - \text{Фурье образ вейвлета.}$$

Для выполнения прямого и обратного вейвлет преобразования нужно иметь вейвлеты на основе ортогональных базисных функций. Функция Хаара – простейший пример ортогонального вейвлета. Функция  $\phi$  у него имеет значение 1 на интервале  $[0,1]$  и 0 за пределами этого интервала, а функция  $\psi$  имеет вид прямоугольных импульсов: 1 на интервале  $[0,0.5]$  и -1 в интервале  $[0.5,1]$ .

К другим хорошо известным ортогональным вейвлетам относятся вейвлеты Добеши.

Поскольку непрерывное вейвлет преобразование требует больших вычислительных затрат, то для практического его применения необходима его дискретизация параметром  $a$  и  $b$ . Для избежания избыточности вейвлет преобразования можно задать значения  $a$  и  $b$  на множестве  $Z = \{\dots, -1, 0, 1, \dots\}$  равные:

$$a = 2^j \text{ и } b = k2^j, \quad (7.9)$$

где  $j$  и  $k$  – целые числа.

Подобная дискретизация наиболее распространена, а сама сетка дискретизации называется диадической. Её важной особенностью является исключение перекрытия носителей вейвлетов, т.е. устранение избыточности вейвлет преобразования.

## 7.2.2. Частотный подход к вейвлет преобразованиям

Частотная область вейвлетов может быть разбита на две составляющие: низкочастотную и высоко частотную. Их частота раздела равна половине частоты дискретизации сигнала. Для их разделения достаточно использовать два фильтра: низко частотный  $Lo$  и высокочастотный  $Hi$ . Фильтр  $Lo$  даёт частотный образ для аппроксимации (грубого приближения), а фильтр  $Hi$  – для его детализации.

Коэффициенты фильтров  $Hi$  и  $Lo$  представляют собой детализирующие коэффициенты вейвлет декомпозиции сигнала и их коэффициенты аппроксимации.

Поскольку фильтры представляют половину всех частотных компонент сигнала, то не попавшие в полосу прозрачности компоненты, могут быть удалены. Данная операция имеет название децимация и обозначается как:  $\downarrow 2$ . Если просто сложить полученные на выходах фильтров сигналы, то получится исходный сигнал, т.е. будет иметь место полная реконструкция сигнала.

Но низкочастотный фильтр можно снова подвергнуть разложению через два фильтра и провести децимацию полученных результатов, тем самым изменить уровень реконструкции сигнала. Таким образом, может быть сформирована система вейвлет-фильтров, реализующих операцию декомпозиции сигнала того или иного уровня.

## 7.2.3. Вейвлет-ряды дискретного времени

В большинстве приложений работа осуществляется с дискретными сигналами. Поэтому представляют интерес дискретные аналоги НВП, которые преобразуют дискретный сигнал в непрерывный и дискретный сигналы, соответственно.

Пусть имеется некоторая непрерывная функция  $f_0(x) \in V_0$ . Дискретный сигнал  $c_n$  представим как последовательность коэффициентов при масштабирующих функциях, по которым раскладывается  $f_0(x)$ :

$$f_0(x) = \sum_n c_{0,n} \phi_{0,n}(x), \quad (7.10)$$

Где  $c_{0,n} = c_n$ .

Другими словами, сигнал интерпретируется как последовательность коэффициентов разложения, полученную в ходе кратномасштабного анализа функции  $f_0(x)$ . Тогда можно вычислить аппроксимацию этой функции, принадлежащие пространствам  $V_1, V_2, \dots$ . Пространства  $V_{-1}, V_{-2}, \dots$  не имеют значения при данной интерпретации.

Согласно концепции кратномасштабного анализа функция  $f_0(x)$  декомпозируется на две функции  $f_1(x) \in V_1$  и  $e_1(x) \in W_1$ :

$$f_0(x) = f_1(x) + e_1(x) = \sum_k c_{1,k} \phi_{1,k}(x) + \sum_k d_{1,k} \psi_{1,k}(x). \quad (7.11)$$

Таким образом, получили две новые последовательности  $c_{1,n}$  и  $d_{1,n}$ . Этот процесс может быть продолжен по  $f_1(x)$ , и функция  $f_0(x)$  (а также и последовательность  $c_n$ ) будет представлена совокупностью коэффициентов  $d_{m,n}, m \in Z^+, n \in Z$ .

Рассмотрим, как вычисления ДВП могут быть выполнены с использованием операций только над дискретными сигналами. С учетом того, что масштабирующая функция образует базис соответствующего пространства, можно получить

$$\begin{aligned} c_{1,k} &= \langle \phi_{1,k}(x), f_1(x) \rangle = \langle \phi_{1,k}, f_0(x) - e_1(x) \rangle = \left\langle \phi_{1,k}(x), \sum_n c_{0,n} \phi_{0,n}(x) \right\rangle = \\ &= \sum_{n \in Z} c_{0,n} \langle \phi_{1,k}(x), \phi_{0,n}(x) \rangle = 2^{1/2} \sum_{n \in Z} c_{0,n} h_{n+2k}. \end{aligned} \quad (7.12)$$

Отсюда оказывается возможным итеративное вычисление коэффициентов  $c_{j,k}$  и  $d_{j,k}$  без непосредственного использования функций  $\phi(x)$  и  $\psi(x)$ .

$$\begin{aligned} c_{j,k} &= 2^{1/2} \sum_n c_{j-1,n} h_{n+2k}, \\ d_{j,k} &= 2^{1/2} \sum_n c_{j-1,n} g_{n+2k}, \end{aligned} \quad (7.13)$$

получив, таким образом, полностью дискретный процесс декомпозиции. Последовательности  $h_n$  и  $g_n$  называются фильтрами. Следует отметить, что  $c_{j,k}$  и  $d_{j,k}$  имеют «половинную» длину по сравнению с  $c_{j-1,k}$ . За счет этого не вводится избыточности.

Обратный процесс заключается в получении  $c_{j-1}$  из  $c_j$  и  $d_j$ :

$$\begin{aligned} c_{j-1,n} &= \langle \phi_{j-1,n}, f_{j-1}(x) \rangle = \langle \phi_{j-1,n}(x), f_j(x) + e_j(x) \rangle = \\ &= \left\langle \phi_{j-1,n}, \sum_k c_{j,k} \phi_{j,k}(x) \right\rangle + \left\langle \phi_{j-1,n}, \sum_k d_{j,k} \psi_{j,k}(x) \right\rangle = \\ &= \sum_k c_{j,k} \langle \phi_{j-1,n}, \phi_{j,k}(x) \rangle + \sum_k d_{j,k} \langle \phi_{j-1,n}, \psi_{j,k}(x) \rangle = \\ &= 2^{1/2} \sum_k c_{j,k} h_{n+2k} + 2^{1/2} \sum_k d_{j,k} g_{n+2k}. \end{aligned} \quad (7.14)$$

Длина последовательности  $c_{j-1}$  вдвое больше длины последовательности  $c_j$  или  $d_j$ .

Для фильтров  $h_n$  и  $g_n$  существуют следующие ограничения:

$$2 \sum_k (h_{n+2k} h_{p+2k} + g_{n+2k} g_{p+2k}) = \delta_{n,p}, \quad (7.15)$$

$$2 \sum_n h_{n+2k} h_{n+2p} = 2 \sum_n g_{n+2k} g_{n+2p} = \delta_{k,p}, \quad (7.16)$$

$$2 \sum_n h_{n+2k} h_{n+2p} = 0. \quad (7.17)$$

## 7.2.4. Дискретное вейвлет-преобразование

Опишем ДВП в матричном виде, а затем – на основе банков фильтров, что наиболее часто используется при обработке сигналов.

В обоих случаях предполагают, что базисные функции  $\phi(x)$  и  $\psi(x)$  компактно определены. Это автоматически гарантирует финитность последовательностей  $h_n$  и  $g_n$ . Далее предположим, что сигнал, подвергаемый преобразованию, имеет длину  $N = 2^d, d \in Z^+$ .

Обозначим через вектор  $v^j$  последовательность конечной длины  $c_{j,n}$  для некоторого  $j$ . Этот вектор преобразуется в вектор  $v^{j+1}$ , содержащий последовательности  $c_{j+1,n}$  и  $d_{j+1,n}$ , каждая из которых половинной длины. Преобразование может быть записано в виде матричного умножения  $v^{j+1} = M_j v^j$ , где матрица  $M_j$  - квадратная и состоит из нулей и элементов  $h_n$ , умноженных на  $\sqrt{2}$ . В силу свойств  $h_n$  матрица  $M_j$  является ортонормированной, и обратная ей матрица равна транспонированной. В качестве иллюстрации рассмотрим следующий пример. Возьмем фильтр длиной  $L = 4$ , последовательность длиной  $N = 8$ , а в качестве начального значения -  $j = 0$ . Последовательность  $g_n$  получим из  $h_n$  по формуле (7.16), где  $t = L/(2-1) = 4$ . Тогда операция матрично-векторного умножения будет представлена в виде:

$$\begin{bmatrix} c_{1,0} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ d_{1,0} \\ d_{1,1} \\ d_{1,2} \\ d_{1,3} \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & \\ & & & & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & & & & & h_0 & h_1 \\ h_3 & -h_2 & h_1 & -h_0 & & & & \\ & & h_3 & -h_2 & h_1 & -h_0 & & \\ & & & & h_3 & -h_2 & h_1 & -h_0 \\ h_1 & -h_0 & & & & & h_3 & -h_2 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{0,4} \\ c_{0,5} \\ c_{0,6} \\ c_{0,7} \end{bmatrix} \quad (7.18)$$

Обратное преобразование тогда выполняется как умножение  $v^{j+1}$  на обратную матрицу  $M_j^T$ :



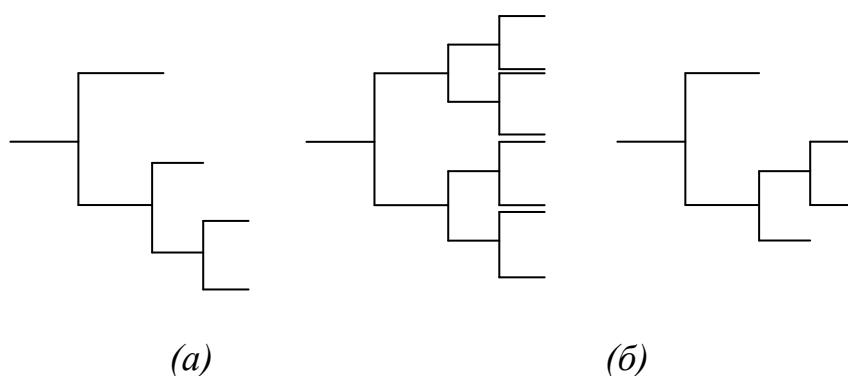




### 7.2.5. Пакеты вейвлетов (алгоритм одиночного дерева)

Вейвлет-преобразование сигнала выполняется путем его пропускания через каскадно соединенные двухканальные схемы А-С. При этом каскадирование производится по низкочастотной области. Причина этого в неявном предположении, что эта область содержит больше информации об исходном сигнале. В результате получается «однобокое» дерево (рис. 7.3(а)). Данное предположение оправданно для многих реальных сигналов. В самом деле, оно означает, что наш сигнал является низкочастотным на большом интервале времени, а высокочастотные составляющие появляются на коротком интервале. Однако для некоторых сигналов это предположение не выполняется.

Метод пакетов вейвлетов основан на определении того, по какой области на данном уровне выгоднее производить каскадирование. Для этого вначале производится каскадирование по обеим субполосам. В результате получается так называемое «полное», «сбалансированное» дерево (рис. 7.3(б)), напоминающее дерево, присущее кратковременному преобразованию Фурье. Далее, на основе введенной функции стоимости определяется наилучший путь по этому дереву (рис. 7.3(в)).



*Рис. 7.3. Разбиение частотно-временной плоскости при помощи пакетов вейвлетов: (а) вейвлет-декомпозиция; (б) полная, аналогичная STFT декомпозиция; (в) пример декомпозиции при помощи пакетов вейвлетов*

Если исходный блок вейвлет - фильтров был ортогональным, то и схема, соответствующая любой конфигурации дерева, будет ортогональной, так как она есть не что иное, как каскадное соединение ортогональных блоков. Таким образом, получается базис, адаптированный к сигналу.

### 7.2.6. Целочисленное вейвлет-преобразование

Рассмотрим методы получения целочисленных вейвлет-коэффициентов изображения. Эти методы могут быть применены для сжатия изображения, как без потерь, так и с потерями. В основе рассматриваемых методов лежит

некоторая модификация вейвлет-преобразования, позволяющая производить все вычисления в целочисленном виде. Полученное преобразование не является, строго говоря, вейвлет-преобразованием, но обладает всеми его свойствами [2]. Теоретически при вейвлет-преобразовании потери информации не происходит. Однако при реализации возникают неизбежные ошибки округления вейвлет-коэффициентов. Вместе с тем, в некоторых приложениях обработки изображений полная обратимость преобразования является важной.

Целочисленное вейвлет-преобразование позволяет достичь полного контроля над точностью вычислений. Поэтому оно получило название обратимого вейвлет-преобразования. Кроме того, целочисленность вычислений ускоряет выполнение преобразования.

Рассмотрим два примера, поясняющие обсуждаемые далее методы. Для простоты все выкладки производятся для одного уровня разложения и для одномерного сигнала четной длины. Пусть  $\{c_n^0\}_{n=0}^{N-1}$  - исходный сигнал, где верхний индекс показывает уровень разложения (0), нижний – конкретную точку сигнала. Пусть  $\{c_n^1\}_{n=0}^{N_1-1}$  и  $\{d_n^1\}_{n=0}^{N_1-1}$  - составляющие его разложения на первом уровне (низкочастотная и высокочастотная части, соответственно). Здесь  $N_1 = N/2$ .

**Целочисленное вычисление вейвлет-преобразование (2,2).** Это преобразование эквивалентно вейвлет-преобразованию Хаара, использующему следующие фильтры декомпозиции:

$$\tilde{h}_0 = \tilde{h}_1 = \tilde{g}_0 = -\tilde{g}_1 = 1/2 \quad (7.20)$$

Вычисление ведется следующим образом:

$$d_k^1 = c_{2k}^0 - c_{2k+1}^0, \quad k = 0, \dots, N_1 - 1, \quad (7.21)$$

$$c_k^1 = \text{int}\left(\frac{d_k^1}{2}\right) + c_{2k+1}^0, \quad k = 0, \dots, N_1 - 2, \quad (7.22)$$

$$c_{N_1-1}^1 = \text{int}\left(\frac{d_{N_1-1}^1}{2} + c_{N-1}^0\right) \quad (7.23)$$

В выражениях (7.22), (7.23)  $\text{int}$  означает операцию выделения целой части. Таким образом, все элементы  $\{c_n^1\}_{n=0}^{N_1-1}$  и  $\{d_n^1\}_{n=0}^{N_1-1}$  будут целыми числами. Из (7.21)-(7.23) легко получить алгоритм реконструкции:

$$c_{2k+1}^0 = c_k^1 - \text{int}\left(\frac{d_k^1}{2}\right), \quad k = 0, \dots, N_1 - 1, \quad (7.24)$$

$$c_{2k}^0 = d_k^1 + c_{2k+1}^0, \quad k = 0, \dots, N_1 - 1. \quad (7.25)$$

**Вейвлет-преобразование Лэйзи.** Вейвлет-преобразование Лэйзи заключается в простом разбиении входного сигнала на четную и нечетную части. На этапах декомпозиции и реконструкции используются одни и те же формулы:

$$c_k^1 = c_{2k}^0, \quad k = 0, \dots, N_1 - 1, \quad (7.26)$$

$$d_k^1 = c_{2k+1}^0, \quad k = 0, \dots, N_1 - 1.$$

Преобразования, используемые в этих примерах, не подходят для кодирования изображений. Однако на их основе могут быть получены значительно более эффективные преобразования. Они могут рассматриваться как стартовая точка для получения алгоритмов целочисленного обратимого вейвлет-преобразования.

Отметим интересное свойство вышеприведенных преобразований. Оно заключается в том, что если пиксели изображения представляются некоторым числом бит, то такое же число бит может быть использовано в компьютере для представления значений вейвлет-коэффициентов. Данное свойство преобразования получило название свойства сохранения точности.

**Целочисленное вычисление вейвлет-преобразования (1,3)**. Это нелинейное преобразование является разновидностью преобразования, использующего биортогональную пару фильтров:  $\tilde{h}_n = \{1,0,0\}$ ,  $\tilde{g}_n = \{1/4,-1/2,1/4\}$ . Вычисления начинаются с вейвлета Лэйзи (7.26) с последующим изменением высокочастотных коэффициентов:

$$\begin{cases} d_k^1 = \text{int}\left(\frac{c_k^1 + c_{k+1}^1}{2}\right) - d_k^{1,0}, & k = 0, \dots, N_1 - 2, \\ d_{N_1-1}^1 = c_{N_1-1}^1 - d_{N_1-1}^{1,0}. \end{cases} \quad (7.27)$$

Реконструкция выполняется следующим образом:

$$c_{2k}^0 = c_k^1, \quad k = 0, \dots, N_1 - 1, \quad (7.28)$$

$$\begin{cases} c_{2k+1}^0 = \text{int}\left(\frac{c_{2k}^0 + c_{2k+2}^0}{2}\right) - d_k^1, & k = 0, \dots, N_1 - 2, \\ c_{N_1-1}^0 = c_{N_1-2}^0 - d_{N_1-1}^1. \end{cases} \quad (7.29)$$

**Целочисленное вычисление вейвлет-преобразования (2,6)**. Данное преобразование эквивалентно использованию следующих фильтров анализа:

$$\tilde{h}_n = \{0,0,1/2,1/2,0,0\}, \quad \tilde{g}_n = \{-1/16,-1/16,1/2,-1/2,1/16,1/16\}$$

Декомпозиция выполняется аналогично преобразованию Хаара с добавлением еще одного шага. Вначале производятся вычисления по формулам (7.21)-(7.23). Вместо  $d_0^1$  в данных формулах теперь используется обозначение  $d_0^{1,0}$ . Затем производится изменение высокочастотных коэффициентов по формулам:

$$\begin{cases} d_0^1 = \text{int}\left(\frac{c_0^1 - c_1^1}{4}\right) - d_0^{1,0}, \\ d_k^1 = \text{int}\left(\frac{c_{k-1}^1 - c_{k+1}^1}{4}\right) - d_k^{1,0}, & k = 1, \dots, N_1 - 2, \end{cases} \quad (7.30)$$

$$d_{N_1-1}^1 = \text{int}\left(\frac{c_{N_1-2}^1 - c_{N_1-1}^1}{4}\right) - d_{N_1-1}^{1,0}. \quad (7.31)$$

Алгоритм реконструкции аналогичен алгоритму декомпозиции. Он выполняется в «обратном» порядке:

$$\begin{cases} d_0^{1,0} = \text{int}\left(\frac{c_0^1 - c_1^1}{4}\right) - d_0^1, \\ d_k^{1,0} = \text{int}\left(\frac{c_{k-1}^1 - c_{k+1}^1}{4}\right) - d_k^1, \quad k = 1, \dots, N_1 - 2, \end{cases} \quad (7.32)$$

$$d_{N_1-1}^{1,0} = \text{int}\left(\frac{c_{N_1-2}^1 - c_{N_1-1}^1}{4}\right) - d_{N_1-1}^1 \quad (7.33)$$

и, далее, по формулам (7.24)-(7.25) с заменой в них  $d_i^1$  на  $d_i^{1,0}$ .

**Целочисленное вычисление вейвлет –преобразования (5,3).** Такое преобразование также является разновидностью биортогонального преобразования и использует следующую пару фильтров:

$$\tilde{h}_n = \{-1/8, 1/4, 3/4, 1/4, -1/8\}, \quad \tilde{g}_n = \{1/4, -1/2, 1/4, 0, 0\}.$$

Декомпозиция производится следующим образом:

$$c_k^{1,0} = c_{2k}^0, \quad k = 0, \dots, N_1 - 1, \quad (7.34)$$

$$\begin{cases} d_k^1 = \text{int}\left(\frac{c_{2k}^0 + c_{2k+2}^0}{2}\right) - c_{2k+1}^0, \quad k = 0, \dots, N_1 - 2, \\ d_{N_1-1}^1 = c_{N_1-2}^0 - c_{N_1-1}^0, \end{cases} \quad (7.35)$$

$$\begin{cases} c_0^1 = c_0^{1,0} - \text{int}\left(\frac{d_0^1}{2}\right), \\ c_k^1 = c_k^{1,0} - \text{int}\left(\frac{d_{k-1}^1 + d_k^1}{4}\right), \quad k = 0, \dots, N_1 - 2, \\ c_{N_1-1}^1 = c_{N_1-2}^{1,0} - \text{int}\left(\frac{d_{N_1-2}^1 + d_{N_1-1}^1}{4}\right). \end{cases} \quad (7.36)$$

Реконструкция осуществляется по следующим формулам :

$$c_0^0 = c_0^1 + \text{int}\left(\frac{d_0^1}{2}\right), \quad (7.37)$$

$$c_{2k}^0 = c_k^1 + \text{int}\left(\frac{d_{k-1}^1 + d_k^1}{4}\right), \quad k = 1, \dots, N_1 - 2, \quad (7.38)$$

$$c_{N_1-2}^0 = c_{N_1-1}^1 + \text{int}\left(\frac{d_{N_1-2}^1 + d_{N_1-1}^1}{4}\right), \quad (7.39)$$

$$c_{2k+1}^0 = \text{int}\left(\frac{c_{2k}^0 + c_{2k+2}^0}{2}\right) - d_k^1, \quad k = 0, \dots, N_1 - 2, \quad (7.40)$$

$$c_{N_1-1}^0 = c_{N_1-2}^0 - d_{N_1-2}^1. \quad (7.41)$$

### 7.3. Применение вейвлет-преобразований для сжатия изображения

Вейвлет-кодер изображения устроен так же, как и любой другой кодер с преобразованием. Он состоит из трех основных частей: декоррелирующее преобразование, процедура квантования и энтропийное кодирование. В настоящее время во всем мире проводятся исследования по усовершенствованию всех трех компонент базового кодера [4,26].

Выбор оптимального базиса вейвлетов для кодирования изображения является трудной и вряд ли решаемой задачей. Известен ряд критериев построения «хороших» вейвлетов, среди которых наиболее важными являются: гладкость, точность аппроксимации, величина области определения, частотная избирательность фильтра. Тем не менее, наилучшая комбинация этих свойств неизвестна [1,4,10,18,26].

Для выбора наилучшего (по соотношению вычислительная сложность метода / размер сжатых данных после вторичного сжатия) вейвлет преобразования был проведен следующий эксперимент. К одному и тому же изображению типа портрет («Lena»), были применены следующие одноуровневые вейвлет преобразования: преобразование Хаара, преобразование 1.3., преобразование 2.6. и преобразование 5.3. .

На каждом шаге преобразования выполняется два разбиения по частоте, а не одно. Предположим, имеем изображение размером  $N \times N$ . Сначала каждая из  $N$  строк изображения делится на низкочастотную и высокочастотную половины. Получается два изображения размерами  $N \times N/2$ . Далее, каждый столбец делится аналогичным образом. В результате получается четыре изображения размерами  $N/2 \times N/2$ : низкочастотное по горизонтали и вертикали, высокочастотное по горизонтали и вертикали, низкочастотное по горизонтали и высокочастотное по вертикали и высокочастотное по горизонтали и низкочастотное по вертикали. Первое из вышеназванных изображений делится аналогичным образом на следующем шаге преобразования, как показано на рис.7.4.

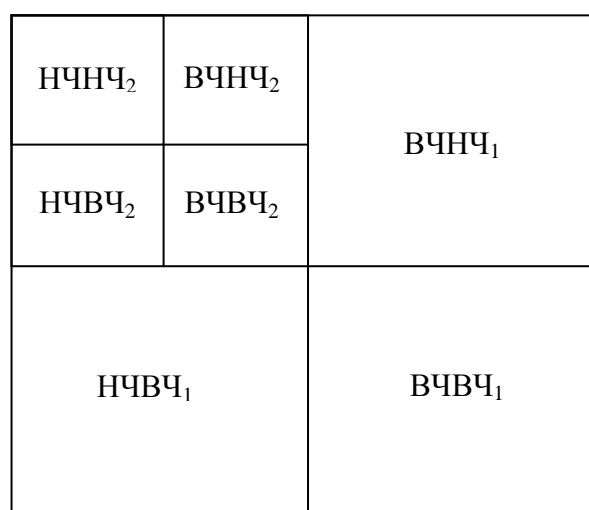


Рис.7.4. Два уровня вейвлет-преобразования изображения

В результирующий файл записывались коэффициенты всех субполос, кроме диагональных, поскольку именно диагональные субполосы содержат шумовые составляющие изображения. Полученные файлы сжимались вторичным методом сжатия (алгоритм ZIP сжатия). Результаты эксперимента приведены в таблице 7.1.

Таблица 7.1.

Выбор наилучшего целочисленного вейвлет преобразования

Метод вейвлет преобразования	Размер файла с частотными коэффициентами (kb)	Размер файла после вторичного сжатия (kb)	Коэф-нт. корреляции между ориг. и восст-ым изобр.
Преобразование Хаара	169	73	0.9990
Преобразование 2.2	172	74	0.9991
Преобразование 1.3	172	74	0.9988
Преобразование 2.6	178	77	0.9991
Преобразование 5.3	176	75	0.9992

На рисунке 2.5. приведены разностные изображения для преобразований Хаара, преобразования 2.2 и преобразования 5.3.

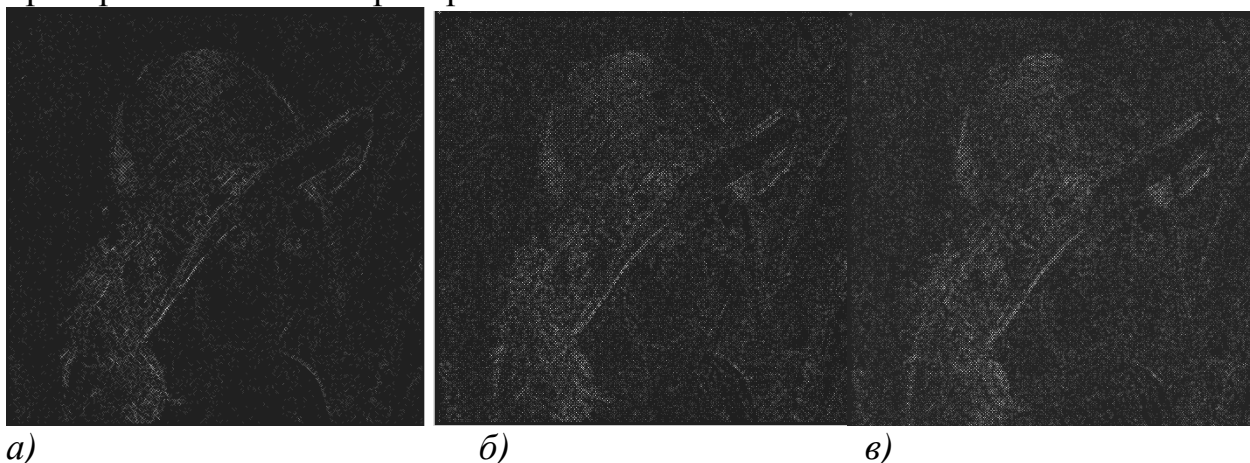


Рис. 7.5. Разностные изображения для преобразований Хаара (а), 2.2.(б) и 5.3(в)

Данные преобразования вносят меньше потерь при исключении диагональной субполосы и образуют хорошо «пакуемые» частотные коэффициенты.

Таким образом, метод Хаара обладает наименьшей вычислительной сложностью и получает хорошо «пакуемые» частотные коэффициенты. Именно этот метод вейвлет преобразования используется во всех разработанных методах сжатия изображений.

Следующим этапом в алгоритмах сжатия изображений является этап квантования частотных коэффициентов. В большинстве вейвлет-кодеров применяется скалярное квантование. Существуют две основные стратегии выполнения скалярного квантования. Если заранее известно распределение коэффициентов в каждой полосе, оптимальным будет использование

квантователей Ллойда-Макса с ограниченной энтропией для каждой субполосы. В общем случае подобным знанием мы не обладаем, но можем передать параметрическое описание коэффициентов путем посылки декодеру дополнительных бит. Априорно известно, что коэффициенты высокочастотных полос имеют обобщенное гауссовское распределение с нулевым матожиданием. На практике обычно применяется намного более простой равномерный квантователь с «мертвой» зоной. Как показано на рис. 7.6, интервалы квантования имеют размер  $\Delta$ , кроме центрального интервала (возле нуля), чей размер обычно выбирается  $2\Delta$ .

Коэффициенту, попавшему в некоторый интервал, ставится в соответствие значение центра этого интервала. В случае асимптотически высоких скоростей кодирования равномерное квантование является оптимальным. Хотя в практических режимах работы квантователи с «мертвой» зоной субоптимальны, они работают почти так же хорошо, как квантователи Ллойда-Макса, будучи намного проще в исполнении. Кроме того, они устойчивы к изменениям распределения коэффициентов в субполосе. Дополнительным их преимуществом является то, что они могут быть вложены друг в друга для получения вложенного битового потока.



Рис. 7.6. Равномерный квантователь с «мертвой» зоной

В заключении необходимо отметить о возможность применения вейвлет преобразований к цветным изображениям. Обычно, цветные изображения представлены в RGB системе цветопрдставления.

Вейвлет преобразования Хаара можно применять непосредственно к отдельным составляющим RGB изображения, но можно, как и в алгоритме JPEG, использовать YCrCb цветопрдставление. В данном случае составляющие Cr и Cb могут непосредственно подвергаться вейвлет преобразованию, а могут, с целью сокращения информационной избыточности, предварительно быть прорежены, т.е. от исходных Cr и Cb сохраняются значения через строчку и через столбец.

Эксперименты показывают, что при использовании преобразования Хаара для RGB плоскостей и для YCrCb составляющих, восстановленные изображения практически идентичны.

Вейвлет преобразования обладают многими полезными свойствами, применимыми как при обработке, так и при сжатии изображений.

Для сжатия изображений наиболее полезными являются следующие свойства:

- возможность целочисленного обратимого преобразования,
- частотные коэффициенты преобразования обладают той же точностью, что и отчёты изображения.

- вейвлет коэффициенты одновременно локализованы как в пространственной, так и в частотной областях,
- вейвлет коэффициенты сдвинуты и распространены по вложенным масштабированным подуровням,
- если данный вейвлет коэффициент большой либо маленький, то смежный с ним вейвлет коэффициент также либо большой, либо маленький,
- масштаб значений вейвлет коэффициентов сохраняется от уровня к уровню.

Исследования показывают, что наилучшим преобразованием по соотношению вычислительная сложность метода / размер сжатых данных после вторичного сжатия является преобразование Хаара. Данное преобразование применимо не только для изображений в градациях серого, но и к цветным изображениям. Для цветных изображений данное преобразование можно применять как к RGB плоскостям, так и к другим производным цветопредставления, например, YCrCb.

Актуальными при сжатии изображений на основе вейвлет преобразования являются задачи:

- оптимального обхода плоскости вейвлет коэффициентов,
- поиска наилучшего метода вторичного сжатия вейвлет коэффициентов,
- поиска метода оптимального кодирования значимых вейвлет коэффициентов,
- совмещения алгоритмов сжатия с алгоритмами защиты авторских прав на изображения.



## 8. БЫСТРЫЕ АЛГОРИТМЫ ОРТОГОНАЛЬНЫХ ПРЕОБРАЗОВАНИЙ

### 8.1. Вычислительная сложность ДПФ и способы её сокращения

Во второй главе мы пришли к выводу, что любое ортогональное преобразование в матричной форме может быть описано как процедура умножения вектора исходных данных на матрицу ядра (при прямом преобразовании)

$$F = kB_N X$$

или вектора результатов разложения исходного сигнала по тем или иным базисным ортогональным функциям на обратную матрицу (при обратном преобразовании):

$$X = k B_N^{\dagger} F$$

В общем случае вычислительная сложность такой процедуры составляет

$$Q = N^2 (Б.О.),$$

где отдельная базовая операция включает операцию умножения и сложения действительных чисел или те же операции умножения и сложения, но для комплексных чисел, в случае преобразования Фурье. С учётом того, что для комплексных чисел

$$A_1 * A_1 = (a + jb)(c + jd) = (ac - bd) + j(bc + ad)$$

сложность вычислительной процедуры ДПФ

$$Q_{\text{ДПФ}} = 4N^2 \text{ Б.О.}, \quad (8.1)$$

если под Б.О. понимать те же операции, как и в базисах действительных функций.

В целях сокращения объёма вычислений можно выполнить формирование отсчётов вектора  $F$  с учётом вырождения (т.е. тривиальности) первой строки и первого столбца матрицы ДПФ:

$$\begin{cases} f_0 = \sum_{k=0}^{N-1} x_k \\ f_n = x_0 + \sum_{k=1}^{N-1} x_k * e^{2\pi knj/N}; n = \overline{1, N-1} \end{cases} \quad (8.2)$$

Отметим особенности представления чётных и нечётных функций при ДПФ и ДПХ. Пусть исходный сигнал описан как суперпозиция чётной и нечётной составляющей:

$$X = X_{\text{sim}} + X_{\text{asim}}$$

причем для каждой из составляющих можно записать

$$\begin{cases} X_{\text{sim}_n} = X_{\text{sim}_{-n}} = X_{\text{sim}_{N-n}} \\ X_{\text{asim}_N} = -X_{\text{asim}_{-N}} = -X_{\text{asim}_{N-n}} \end{cases} \quad (8.3)$$

С учётом такого представления сигнала можно записать, что

$$\begin{cases} X_{sim_n} * \cos \frac{2\pi}{N} kn = X_{sim_{N-n}} * \cos \frac{2\pi}{N} (N-n)k \\ X_{asim_n} * \cos \frac{2\pi}{N} kn = -X_{asim_{N-n}} * \cos \frac{2\pi}{N} (N-n)k \end{cases} \quad (8.4a)$$

$$\begin{cases} X_{sim_n} * \sin \frac{2\pi}{N} k = -X_{sim_{N-n}} * \sin \frac{2\pi}{N} (N-n)k \\ X_{asim_n} * \sin \frac{2\pi}{N} k = X_{asim_{N-n}} * \sin \frac{2\pi}{N} (N-n)k \end{cases} \quad (8.4б)$$

поэтому для чётного исходного сигнала “синусная” компонента спектра будет равна 0, а для нечётного сигнала - “косинусная” компонента спектра равна 0.

Поэтому спектр действительного сигнала в базисах Фурье и Хартли описывается чётной функцией и для его задания требуется только косинусная компонента, причём в силу чётности для задания спектра достаточно только  $\frac{N}{2}$  отсчётов  $\left\{ f_0 \div f_{\frac{N}{2}-1} \right\}$ .

Кроме того, матрицы  $E_N$  и  $C_N$  обладают свойством симметрии и периодичности (цикличности) следования элементов. Из симметрии матриц указанных ядер следует, что в общем случае для действительных последовательностей (т.е. вектора  $X$ , содержащего только действительные элементы) может быть выполнено только для  $\frac{N}{2}$  компонент спектра, (например, с номерами  $\left\{ 0, \frac{N}{2}-1 \right\}$ ), поскольку компоненты “положительной” и “отрицательной” областей частот (компоненты с номерами  $\left\{ 0, \frac{N}{2}-1 \right\}$  и  $\left\{ \frac{N}{2}, N-1 \right\}$  имеют одинаковую амплитуду, но противоположные фазы. Поэтому вторую половину компонент можно легко достроить из вычисленных.

Аналогично можно поступить и при вычислении ДПХ - cos-составляющая “положительных” и “отрицательных” спектральных компонент одинакова, а sin-составляющие для “положительных” и “отрицательных” имеют противоположные знаки.

## 8.2. Запись алгоритма БПФ в векторно-матричной форме

Матрицы указанных ортогональных преобразований позволяют значительно сократить объём вычислений, поскольку из  $N^2$  элементов матрицы имеется лишь  $N$  (для ДПФ и ДПХ) различных значений.

Действительно, рассмотрим ДПФ для  $N=4$

$$F = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} * \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_0 + x_1 + x_2 + x_3 \\ x_0 - jx_1 - x_2 + jx_3 \\ x_0 - x_1 + x_2 - x_3 \\ x_0 + jx_1 - x_2 - jx_3 \end{bmatrix} = \begin{bmatrix} (x_0 + x_2) + (x_1 + x_3) \\ (x_0 - x_2) - j(x_1 - x_3) \\ (x_0 + x_2) - (x_1 + x_3) \\ (x_0 - x_2) + j(x_1 - x_3) \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

После выделения встречающихся неоднократно блоков данных можно заметить, что  $(x_0 + x_2)$  и  $(x_0 - x_2)$  есть не что иное, как ДПФ для  $N=2$  и

подвектора  $(x_0, x_2)$ . Аналогичные выводы можно сделать и для других блоков данных, что позволяет записать :

$$\left| E_2 \begin{array}{c} x_0 \\ x_2 \end{array} \right| = \begin{array}{c} x_0 + x_2 \\ x_0 - x_2 \end{array}; \quad \left| E_2 \begin{array}{c} x_1 \\ x_3 \end{array} \right| = \begin{array}{c} x_1 + x_3 \\ x_1 - x_3 \end{array}$$

Попробуем использовать как типовой элемент преобразования процедуру вида  $E_2 * X_2$  и перекомпоновку векторов.

1) Переставим элементы исходного вектора, чтобы сформировать указанные подвектора. Очевидно, что для такой перестановки необходимо двоично-инверсное переупорядочивание вектора  $X$ .

Такая процедура может быть записана как операция умножения вектора  $X$  на перестановочную матрицу  $S_4^1$ .

$$\hat{X}^1 = S_4^1 X \rightarrow \begin{array}{c} \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right| \begin{array}{c} x_0 \\ x_1 \\ x_2 \\ x_3 \end{array} \end{array} * \begin{array}{c} \left| \begin{array}{c} x_0 \\ x_1 \\ x_2 \\ x_3 \end{array} \right| = \begin{array}{c} x_0 \\ x_2 \\ x_1 \\ x_3 \end{array}$$

2) Запишем матрицу

$$\hat{E}_4 = (E_2 \otimes I_2^1) = \left[ \begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{array} \right] \otimes \left[ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] = \left| \begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{array} \right|$$

и получим вектор  $\hat{F}_1$  как:

$$\hat{F}^1 = \hat{E}_4 * \hat{X}^1 = \hat{E}_4 * [S_4^1 X] = \left| \begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{array} \right| \begin{array}{c} x_0 + x_2 \\ x_0 - x_2 \\ x_1 + x_3 \\ x_1 - x_3 \end{array}$$

$$\hat{F}^1 = [E_2 \otimes I_2] S_4^1 * X$$

3) Умножим вектор  $\hat{F}_1$  на диагональную матрицу  $D_4^1$ :

$$\hat{F}^2 = D_4^1 * \hat{F}^1 = \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -j \end{array} \right| \begin{array}{c} x_0 + x_2 \\ x_0 - x_2 \\ x_1 + x_3 \\ x_1 - x_3 \end{array} * \begin{array}{c} \left| \begin{array}{c} x_0 + x_2 \\ x_0 - x_2 \\ x_1 + x_3 \\ x_1 - x_3 \end{array} \right| = \begin{array}{c} x_0 + x_2 \\ x_0 - x_2 \\ x_1 + x_3 \\ -j(x_1 - x_3) \end{array}$$

$$\hat{F}^2 = D_4^1 [E_2 \otimes I_2] * (S_4^1 X)$$

4) Получим вектор  $\hat{X}^2$ , вновь переставив элементы вектора  $\hat{F}^2$  по правилу двоичной инверсии с помощью матрицы  $S_4^1$ :

$$\hat{X}^2 = S_4^1 * \hat{F}^2 = \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right| \begin{array}{c} x_0 + x_2 \\ x_0 - x_2 \\ x_1 + x_3 \\ -j(x_1 - x_3) \end{array} * \begin{array}{c} \left| \begin{array}{c} x_0 + x_2 \\ x_0 - x_2 \\ x_1 + x_3 \\ -j(x_1 - x_3) \end{array} \right| = \begin{array}{c} x_0 + x_2 \\ x_1 + x_3 \\ x_0 - x_2 \\ -j(x_1 - x_3) \end{array}$$

$$\hat{X}^2 = S_4^1 [D_4^1 * (E_2 \otimes I_2)] * (S_4^1 X)$$

5) Умножим вектор  $\hat{X}^2$  на матрицу  $\hat{E}_4$ :

$$F^{\oplus} = E_4^{\oplus} * X^{\oplus} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ & 1 & 1 \\ & 1 & -1 \end{bmatrix} * \begin{bmatrix} x_0 + x_2 \\ x_1 + x_3 \\ x_0 - x_2 \\ -j(x_1 - x_3) \end{bmatrix} = \begin{bmatrix} (x_0 + x_2) + (x_1 + x_3) \\ (x_0 + x_2) - (x_1 + x_3) \\ (x_0 - x_2) - j(x_1 - x_3) \\ (x_0 - x_2) + j(x_1 - x_3) \end{bmatrix} = \begin{bmatrix} f_0 \\ f_2 \\ f_1 \\ f_3 \end{bmatrix}$$

$$\hat{F}^3 = [E_2 \otimes I_2] * S_4^1 [D_4^1 * (E_2 \otimes I_2)] * (S_4^1 X)$$

6) Переставим элементы вектора  $\hat{F}^3$  с помощью матрицы  $S_4^1$ :

$$F = S_4^1 * \hat{F}^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

иначе говоря:

$$F = S_4^1 \{ [(E_2 \otimes I_2) D_4^1] * S_4^1 [(E_2 \otimes I_2) D_4^0] * S_4^1 X \} \quad (8.5)$$

где на первой итерации диагональная матрица

$$D_4^0 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

введена формально, для симметрии матричной записи обеих итераций.

Рассуждая аналогичным образом, можно прийти к алгоритму, состоящему в выполнении последовательности итераций. На рис.8.1 приведен граф алгоритма БПФ, иллюстрирующий выполнение итеративно - связанных вычислений по описанной схеме.

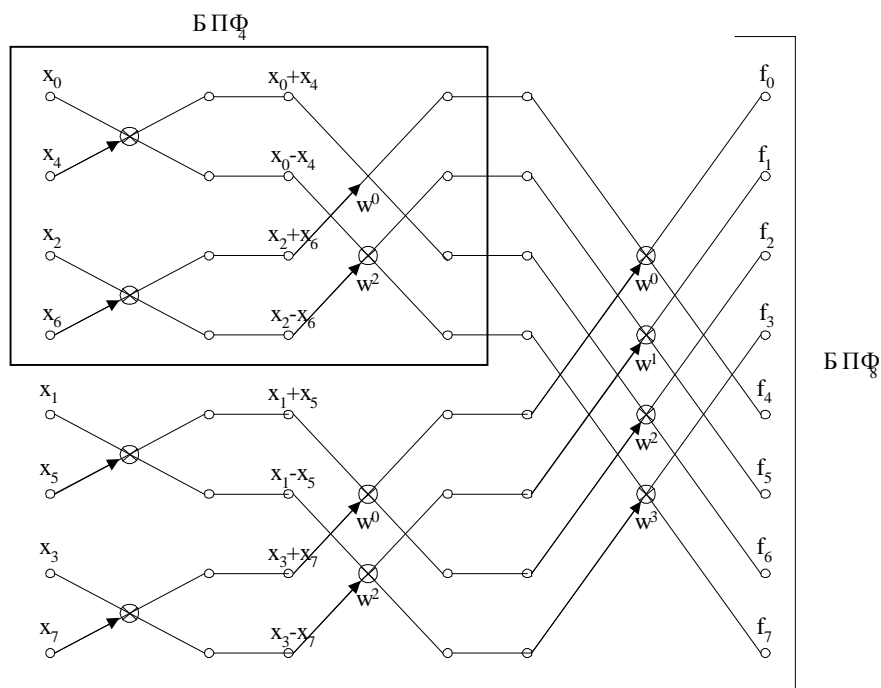


Рис.8.1. Граф процедуры БПФ для  $N = 8$ .

На рисунке обозначены:

$$W^0 \equiv 1 \equiv e^{-j\frac{2\pi \cdot 0}{8}}; W^1 = e^{-j\frac{\pi}{4}} = \frac{\sqrt{2}}{2}(1-j); W^2 = e^{-j\frac{\pi}{2}} = -j; W^3 = e^{-j\frac{3\pi}{4}} = -\frac{\sqrt{2}}{2}(1+j)$$

Число итераций составляет  $m = \log_2 N$  ( $N = 2^m$ ). Перед первой итерацией выполняется г-ично инверсная перестановка элементов вектора исходных данных. В свою очередь каждая итерация сводится к схожим действиям:

1. Перестановка элементов вектора результатов предшествующей итерации;
2. Умножение вектора данных на диагональную матрицу  $D_N^m$  ( $m$  - номер итерации), причём  $D_N^1 \equiv I_N$
3. Умножение вектора результатов (по п.1) на матрицу блочной структуры

$$E_1^{\wedge} (E_N^{\wedge} = E_2 \otimes I_2 \otimes I_2 \otimes I_2 \otimes I_2 \dots)$$

4. Выполнение перестановок элементов вектора, содержащего результаты п.2.

В матричном виде это можно записать [7, 17,25]:

$$F^{(m)} = S_N^{(m)} (B_N^m \tilde{S}_N^m F^{m-1}) \quad (8.6),$$

где  $F^{(0)} = [x_0, x_1, x_2 \dots x_{N-1}]$  - вектор исходных данных,

$F^{(m)} = [f_0, f_1, f_2 \dots f_{N-1}]$  - вектор результатов БПФ,

$\tilde{S}_N^m$  - матрица перестановок вектора выходных данных (т.е.  $F^{(m)}$ ),

$S_N^m$  - матрица перестановок вектора входных данных (т.е.  $F^{(m-1)}$ ).

В свою очередь, матрица  $B_N^{(m)}$  является блочной матрицей, процедура формирования которой определяется выражением:

$$B_N^{(m)} = I_2^{M-m} \otimes \left[ \bigoplus_{K=0}^{2^{m-1}} (E_2 D_2^{K \cdot 2^{m-1}}) \right] \quad (8.7),$$

здесь  $I_2^{M-m}$  - единичная матрица порядка  $2^{M-m}$ ; ( $m = \overline{1, M}$ ),

$\otimes$  - знак прямого произведения матриц (кронекеровского произведения),

$\oplus$  - знак прямой суммы матриц. Заметим, что прямой суммой матриц  $A_N$  и  $B_N$  является диагональная матрица  $C_{2N}$  на главной диагонали которой расположены блоки из исходных матриц:

$$C_{2N} = A_N \oplus B_N = \begin{vmatrix} A_N & \\ & B_N \end{vmatrix} \quad (8.8).$$

$D_2^P = \text{diag}\{W_N^{(0)}, W_N^{(1P)}\}$  - диагональная матрица поворачивающих множителей ( $P = K \cdot 2^{m-1}$ ),

$E_2 = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$  - матрица ДЭФ порядка,

$$W_N^{iP} = \exp(-j \frac{2\pi i P}{N}); P = K \cdot 2^{m-1}; i = \overline{0,1}$$

Такое представление алгоритма БПФ в виде матричных операций носит в большей мере характер формального описания. Действительно, выполнение перестановок как операций умножения вектора на матрицу явно не является

оптимальным путем их реализации, также как и описываемые через аналогичные операции умножения на поворачивающие множители.

Серьезным недостатком такого представления алгоритма БПФ является трудность перехода к программной реализации алгоритма.

Для практических целей более удобно описание алгоритма БПФ через систему рекуррентных выражений.

### 8.3. Представление алгоритма БПФ в виде рекурсивных соотношений

Вернемся к рассмотрению процедуры БПФ. Мы говорили, что на каждой ступени вычисляются группы однородных операций:

- умножение на матрицу  $D$  поворачивающих множителей;
- умножение на матрицу  $\hat{E}_2$ ;
- перестановка элементов вектора результатов.

Основная сложность в записи алгоритма БПФ состоит именно в описании перестановок элементов вектора.

Если рассмотреть граф БПФ (например, для  $N=8$ , рис3.1), то можно увидеть, что на первой итерации как бы независимо обрабатывается 4 фрагмента исходного вектора, содержащие по 2 элемента: такие фрагменты назовем подвекторами. В данном случае это подвектор  $X=\{x_0, x_4\}$ ;  $X=\{x_1, x_5\}$ ;  $X=\{x_2, x_6\}$ ;  $X=\{x_3, x_7\}$ . На второй итерации обрабатывается 2 подвектора по 4 элемента, а на третьей - один вектор из восьми элементов.

Таким образом, суть алгоритма БПФ состоит в разбиении на подвектора по 2 элемента, их независимой обработке, и формированию из промежуточных подвекторов вдвое большей длины (очевидно, что число таких подвекторов уменьшается от итерации к итерации), что выполняется до тех пор, пока не образуется один вектор длиной  $N$  элементов.

Подобная процедура может быть описана как [21]:

$$F^m = \begin{vmatrix} f_{r,m} \\ f_{k+1,m} \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} \begin{vmatrix} W_{N_m}^0 & 0 \\ 0 & W_{N_m}^k \end{vmatrix} \begin{vmatrix} f_{k,m-1} \\ f_{k+1,m-1} \end{vmatrix} \quad (3.9)$$

где  $l=2^{m-1}$ ,  $m \in \overline{1, M}$ ,  $k=(n)_{\text{mod } l}$ ,  $k \in \overline{0, 2^{m-1} - 1}$ ,  $n \in (0, N-1)$  - текущий индекс элемента вектора,  $k$  - соответствует номеру БО при обработке подвектора,  $W_{N_m}^k = \exp(-j \frac{2\pi k}{N_m})$ ,  $N=2^m$ - длина подвектора, причём для  $m=0$  вектор  $F^{(0)} \equiv X$ .

Номер обрабатываемого на данной итерации блока данных определяется целой частью  $(n)_{\text{mod } l}$

Выражение (3.9) определяет обработку отдельного, но любого подвектора на каждой итерации из общего их числа  $N / N_m = 2^{M-m}$ . Индекс  $n$  определяется следующим образом:

$$n_i = N_{m-i} \quad (i = \overline{0, 2^{M-m} - 1}),$$

где  $n_i$  - начальный индекс для  $i$ -го подвектора.

В свою очередь, (3.9) может быть переписано в виде системы из двух уравнений:

$$\begin{cases} f_{k,m} = f_{k,m-1} + f_{k+1,m-1} \cdot W_{N_m}^k \\ f_{k+1,m} = f_{k,m-1} - f_{k+1,m-1} \cdot W_{N_m}^k \end{cases} \quad (8.10)$$

Это выражение соответствует базовой операции, для которой указаны правила формирования текущих индексов по “входным” и “выходным” бабочкам, а также правила вычисления поворачивающего множителя  $W_{N_m}^k$

#### 8.4. Алгоритмы БПФ с прореживанием по времени и по частоте

Быстрые алгоритмы БПФ являются “обратимыми”, т.е. вычисления по графу БПФ (см., например, рис. 3.1) могут выполняться как “слева направо”, так и “справа налево”.

В первом случае алгоритм БПФ основан на “бабочке” вида:

$$\begin{cases} P = A + W^k B \\ Q = A - W^k B \end{cases} \quad (8.11)$$

и называется алгоритмом с прореживанием по времени (алгоритм Кули-Тьюки) [17,21].

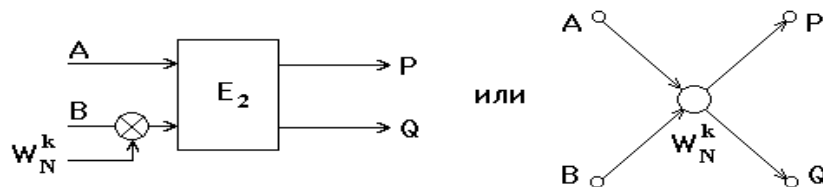
Во втором случае (при вычислении по графу “справа налево”) алгоритм БПФ производится на основе бабочки вида:

$$\begin{cases} P = A + B \\ Q = (A - B) \cdot W^k \end{cases} \quad (8.12)$$

и называется алгоритмом БПФ с прореживанием по частоте (алгоритмом Сэнди-Тьюки). Такие алгоритмы были впервые опубликованы в середине 60-х годов [17,25].

Графы подобных базовых операций приведены на рис. 8.2, а) и б) соответственно.

а)



б)

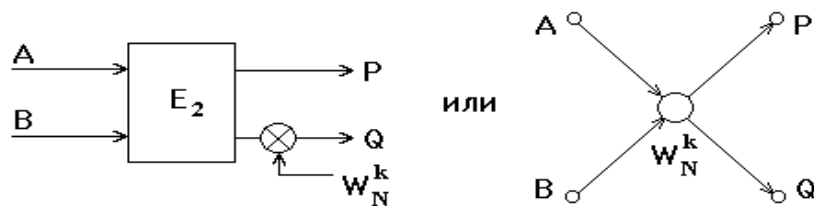


Рис. 8.2. Графы базовых операций БПФ с прореживанием по времени (а) и по частоте (б)

Заметим, что для алгоритма БПФ с прореживанием по частоте не выполняется двоично-инверсная перестановка элементов вектора  $X$  перед первой итерацией, но зато необходимо переставлять элементы векторов результата по закону двоичной инверсии.

### 8.5. Алгоритм БПФ по основанию $r$ ( $N = r^m, r \geq 3$ )

До сих пор мы рассматривали БПФ только для случая, когда  $N = 2^m$ . На практике, однако, достаточно часто возникает необходимость вычисления ДПФ при  $N = r^m$ , где  $r$  отлично от 2 (например,  $N = 125 = 5^3, N = 625 = 5^4; N = 27$  и т.д.).

Для таких случаев несколько позднее Сэнди и Радемахором были разработаны алгоритмы БПФ, основу которых составляют базовые операции “бабочка” следующего вида для алгоритмов с прореживанием по времени:

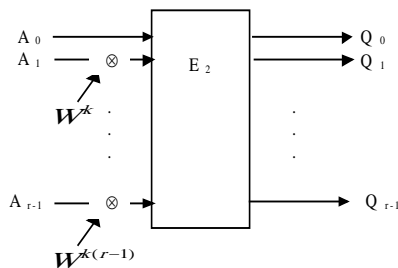
$$\begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ \dots \\ Q_{r-2} \\ Q_{r-1} \end{pmatrix} = [E_r] \cdot \begin{pmatrix} W^0 & & & & & \\ & W^{1K} & & & & \\ & & W^{2K} & & & \\ & & & \dots & & \\ & & & & W^{K(r-1)} & \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ \dots \\ A_{r-2} \\ A_{r-1} \end{pmatrix} \quad (8.13)$$

и для алгоритмов с прореживанием по частоте соответственно:

$$\begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ \dots \\ Q_{r-2} \\ Q_{r-1} \end{pmatrix} = \begin{pmatrix} W^0 & & & & & \\ & W^{1K} & & & & \\ & & W^{2K} & & & \\ & & & \dots & & \\ & & & & W^{K(r-1)} & \end{pmatrix} \cdot [E_r] \cdot \begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ \dots \\ A_{r-2} \\ A_{r-1} \end{pmatrix}$$

Графически каждая из представленных базовых операций может быть изображена так, как это указано на рис.8.3.

а)



б)

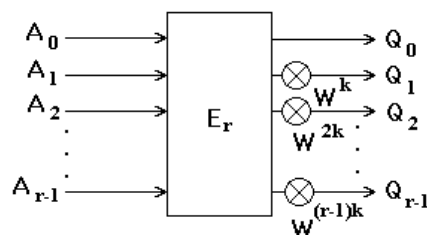


Рис. 8.3. Графы базовых операций БПФ по основанию  $r$

Остановимся на правилах перестановки элементов в таком случае. На входе выполняется перестановка по закону  $r$ -ичной инверсии [13], т.е. на



входе элементы вектора  $X$  объединяются в подвектора из  $r$  - элементов, причём шаг выборок составит:  $\Delta = r^{M-1}$

С помощью системы рекуррентных соотношений, подобных выражению (8.9), алгоритм БПФ для  $N=r^M$  с прореживанием по времени можно описать следующим образом [7,17]:

$$F^m = \begin{pmatrix} f_{k,m} \\ f_{k+l,m} \\ f_{k+2l,m} \\ \dots \\ f_{k+(r-1)l,m} \end{pmatrix} = [E_r] \cdot \begin{pmatrix} W_{N_m}^0 & 0 & 0 & \dots & 0 \\ 0 & W_{N_m}^k & 0 & & 0 \\ 0 & 0 & W_{N_m}^{2k} & & 0 \\ \dots & & \dots & \dots & 0 \\ 0 & 0 & 0 & & W_{N_m}^{(r-1)k} \end{pmatrix} \cdot \begin{pmatrix} f_{k,m-1} \\ f_{k+l,m-1} \\ f_{k+2l,m-1} \\ \dots \\ f_{k+(r-1)l,m-1} \end{pmatrix} \quad (8.14)$$

где  $E_r = \|\exp(-jsq2\pi/r)\|$ ;  $s, q \in \overline{0, r-1}$  - матрица ДЭФ порядка  $r$  (т.е. ДПФ для вектора  $N$  сводится к ряду ДПФ размера  $r$  над блоками),  $k=(n)_{\text{mod } l}$ ,  $n$  текущий индекс элемента вектора ( $n \in \overline{0, N-1}$ ),  $k \in \overline{0, l-1}$ ,

$$l = r^{m-1}, W_{N_m}^{pk} = \exp(-j \frac{2\pi}{N_m} \cdot pk), p \in \overline{0, r-1}, N_m = r^m \quad (8.15)$$

Граф БПФ для  $N=9$  имеет вид, представленный на рис.8.4

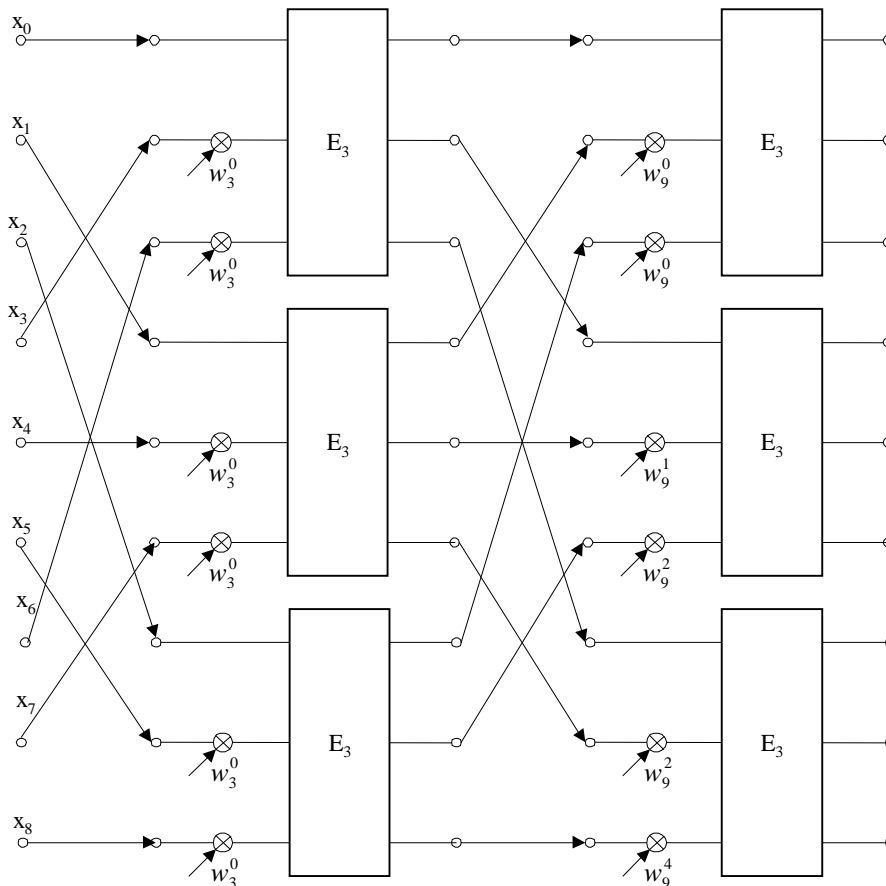


Рис.8.4. Граф БПФ

## 8.6. Вычислительная сложность алгоритмов БПФ

Рассмотрим вначале алгоритмы БПФ для  $N = 2^m$  с прореживанием по времени.

Такой алгоритм является итерационным и включает  $M = \log_2 N$  итераций, причем на каждой стадии выполняется  $N/2$  базовых операций вида (7.11), откуда нетрудно получить, что общая трудоемкость алгоритма БПФ [17]:

$$Q = \left[ \frac{N}{2} \log_2 N \right] q_{\text{BO}}, \quad (3.16)$$

где  $q_{\text{BO}}$  - сложность базовой операции.

В свою очередь, базовая операция требует для своего выполнения 1 операцию умножения и 2 операции сложения комплексных чисел или, в пересчете на операции с действительными числами, 4 операции умножения и 4 операции сложения действительных чисел.

Для ЭВМ предыдущих поколений, где операции умножения выполнялись главным образом программным (а не аппаратным) способом, особую важность имело прежде всего сокращение числа операций умножения как наиболее длительных.

В современных ЭВМ эта задача не столь актуальна, так как длительность практически всех операций близка, особенно в специализированных процессорах.

Поэтому будем полагать, что сложность базовой операции при работе с комплексными числами равна

где  $q_D = q_x + q_+$

$$q_{\text{BO},k} \approx 4 q_D$$

Здесь и далее под  $q$  подразумевается сложность отдельной машинной команды, например, в числе тактов работы процессора, необходимой для выполнения конкретной арифметической операции. Следовательно, для БПФ сложность алгоритма можно определить как:

$$Q_2 = 2 \left[ N \log_2 N \right] q_D \quad (3.17)$$

При обработке вектора данных длиной  $N = r^m$  на каждой из  $M$  итераций выполняется по  $N/r$  базовых операций, т.е.

Однако сложность базовой операции, как это можно видеть из (8.15), составит:

$$Q_1 = \left[ \frac{N}{r} \log_2 N \right] q_{\text{BO}} \quad (3.18)$$

$$q_{\text{BO}} \approx r^2 q_k$$

где  $q_k = q_{\text{УМН}} + q_{\text{СЛ}}$  - для комплексных чисел, или, как и в случае  $N = 2^m$ , посчитаем, что  $q_k = 4q_D$  и получим, подставив в (8.15) значения  $q_{\text{BO}}$ :

$$Q_r = 4N[\log_2 N]r^2 q_D \quad (3.19)$$

В таблице 8.1 приведены значения для времени выполнения ДПФ и БПФ по различным основаниям при условии, что  $t_{cl} = t_{умн} = 100 \text{ нсек.}$

Таблица 8.1.

Размер вектора N		Время выполнения, сек.	
		БПФ	ДПФ
16	$2^4$	$2,56 \cdot 10^{-5}$	$2,0 \cdot 10^{-4}$
25	$5^2$	$2 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
27	$3^3$	$6,5 \cdot 10^{-5}$	$6,5 \cdot 10^{-5}$
32	$2^5$	$6,4 \cdot 10^{-5}$	$8 \cdot 10^{-4}$
81	$3^4$	$7,8 \cdot 10^{-4}$	$5,3 \cdot 10^{-3}$
125	$5^3$	$1,5 \cdot 10^{-3}$	$1,2 \cdot 10^{-3}$
128	$2^7$	$3,6 \cdot 10^{-4}$	$1,3 \cdot 10^{-2}$
512	$2^9$	$1,8 \cdot 10^{-3}$	$2 \cdot 10^{-1}$
625	$5^4$	$10^{-2}$	$3 \cdot 10^{-1}$
1024	$2^{16}$	$4,1 \cdot 10^{-3}$	$8,3 \cdot 10^{-1}$
16384	$2^{14}$	$\approx 1 \text{ сек.}$	$\approx 36 \text{ мин.}$

Поэтому с точки зрения сокращения вычислительных затрат выгодны алгоритмы БПФ по основанию 2 (или 4), причем с ростом основания  $r$  выигрыш уменьшается.

### 8.7. Выполнение БПФ для случаев $N \neq r^M$

Если  $N = m_1 * m_2 * m_3 * m_4 \dots m_M$ , то в этом случае выполняется алгоритм БПФ по смешанному основанию. Он может быть реализован как алгоритм с прореживанием по частоте или как алгоритм с прореживанием по времени. В обоих случаях такой алгоритм выполняется за  $M$  итераций. Однако на каждой итерации выполняется различное число базовых операций, причём размерность матрицы ядра ДПФ базовой операции на каждой итерации различна и составляет  $E_{mi}$ , а число БО соответственно на каждой итерации составляет  $N/m_i$ , где до первой итерации элементы вектора переупорядочиваются с шагом  $N/m_i$  [13]. На рис. 8.5. приведен в качестве примера граф БПФ для  $N = 6 = 3 \times 2$ .

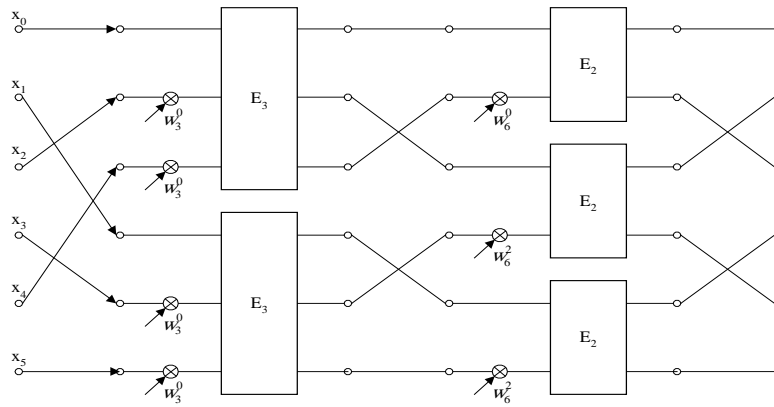


Рис.8.5. Граф БПФ для  $N=6$

Если же  $N$  является простым числом и не может быть разложено на взаимно-простые множители, то в этом случае также можно использовать алгоритм быстрого преобразования, но особого вида - так называемый алгоритм Винограда [15]. Этот алгоритм позволит значительно сократить число операций умножения, но зато число операций сложения сокращается не столь существенно, причем требуются достаточно сложные процедуры перекomпоновки векторов промежуточных результатами.

Поэтому для современных ЭВМ, и особенно спецпроцессоров, когда  $\Delta t_{умн} \approx \Delta t_{сл}$  (а в некоторых из них одинаково время любой операции), такие алгоритмы, как и БПФ по смешанному основанию, не приносят слишком большого выигрыша по времени.

С точки зрения экономии времени целесообразно дополнить вектор до  $N = r^M$  (желательно  $N = 2^M$  или  $N = 4^M$ ) нулями. Однако такой прием приводит к некоторому искажению результата, что связано с введением так называемой оконной функции. Пусть исходный сигнал задан в виде вектора из  $N_1$  отсчетов (рис.8.6.).

Для того, чтобы иметь возможность использовать алгоритмы БПФ дополним сигнал нулевыми отсчетами до общего числа отсчетов  $N = 2^M$ , причем  $N$  - ближайшее большее к  $N_1$ . В этом случае исходный сигнал оказывается заданным с помощью функции

$$\hat{f}(x) = f(x) * g(x),$$

где  $g(x)$  - функция окна вида

$$g(x) = \text{rect}\left(\frac{x * \frac{N_1}{2}}{N_1}\right)$$

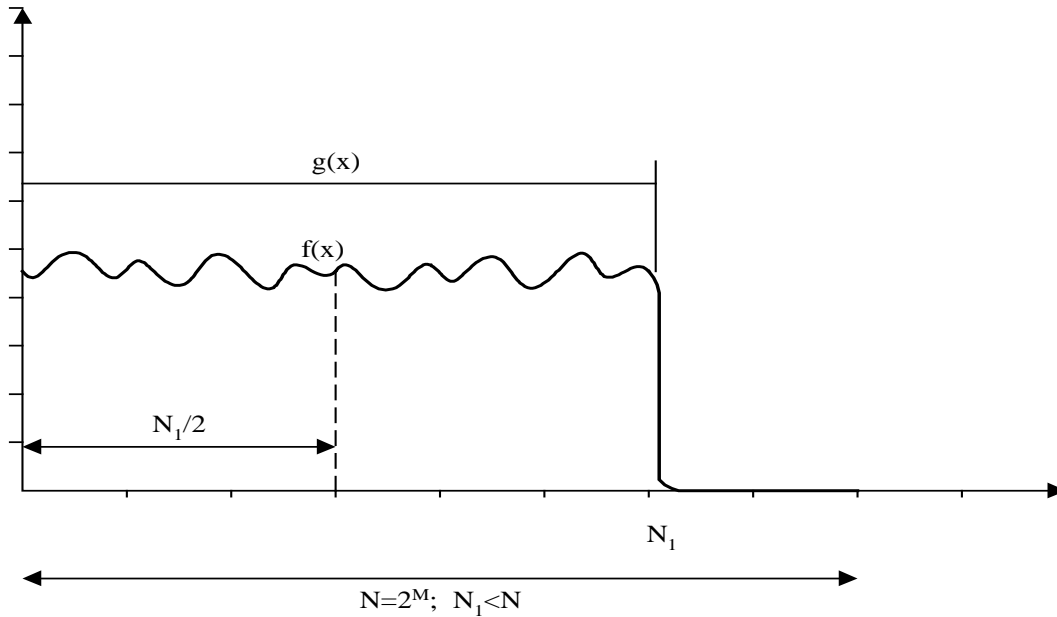


Рис. 8.6. Проявление эффекта окна

Поэтому при вычислении преобразования Фурье от  $f(x)$  проявляется эффект, получивший название “эффекта окна”.

$$F_{(\zeta)} = \int f(x) e^{-j2\pi\zeta x} dx = \int [f(x) * g(x)] e^{-j2\pi\zeta x} dx = F_{(\zeta)} \otimes G_{(\zeta)},$$

где  $G_{(\zeta)} = \text{sinc}(\zeta/N_1)$  при этом функция  $G_{(\zeta)}$  будет тем ближе к  $\delta$ -функции,

чем шире  $\text{rect}[\frac{2x}{N_1\Delta x}]$  или чем  $N_1$  ближе к  $N$ . Проявляется “эффект окна” в

искажении периферийных областей спектра сигнала [21]. Поэтому, во избежание проявления “эффекта окна” удобнее функцию  $f(x)$  при переходе от  $N_1$  к  $N$  сделать периодической с периодом  $N_1$ , дополнив вектор исходных данных до длины  $N$ , повторив первые  $(N - N_1)$  отсчетов за последние элементы вектора  $X$ .

## 8.8. Быстрое преобразование Хартли

Как уже отмечалось в разделе 6.2, множители ядра преобразования Харли не обладают свойством мультипликативности. Это не позволяет достаточно просто записать выражения для итерации алгоритма БПХ в матричной форме.

Выполним выкладки для получения выражений, описывающих итерацию БПХ. Для этого возьмем за основу выражения для описания “бабочки” БПФ ( $N = 2^m$ ) произвольной итерации согласно (8.10):

$$\begin{cases} f_{k,m} = f_{k,m-1} + f_{k+l,m-1} \cdot \left[ \cos \frac{2\pi k}{N_m} - j \sin \frac{2\pi k}{N_m} \right] \\ f_{k+l,m} = f_{k,m-1} - f_{k+l,m-1} \cdot \left[ \cos \frac{2\pi k}{N_m} - j \sin \frac{2\pi k}{N_m} \right] \end{cases} \quad (8.20)$$

здесь  $l=2^{m-1}$ ,  $k=(n) \bmod l$ ,  $N=2^m$ .

Воспользуемся выражениями, связывающими отсчёты ДПФ и ДПХ [3]:

$$\begin{cases} f_{k,m-1} = (h_{k,m-1} + h_{p,m-1}) / 2 - j(h_{k,m-1} - h_{p,m-1}) / 2 & (*) \\ f_{k+l,m-1} = (h_{k+l,m-1} + h_{p+l,m-1}) / 2 - j(h_{k+l,m-1} - h_{p+l,m-1}) / 2 & (**) \\ f_{k,m} = (h_{k,m} + h_{p,m}) / 2 - j(h_{k,m} - h_{p,m}) / 2 & (***) \end{cases}$$

подставим (\*), (\*\*) и (\*\*\*) в формулу (1) выражения (8.20) и получим:

$$\begin{aligned} (h_{k,m} + h_{p,m}) - j(h_{k,m} - h_{p,m}) &= [(h_{k,m-1} + h_{p,m-1}) - j(h_{k,m-1} - h_{p,m-1})] + \\ &+ [(h_{k+l,m-1} + h_{p+l,m-1}) - j(h_{k+l,m-1} - h_{p+l,m-1})] \cdot \left[ \cos \frac{2\pi k}{N_m} - j \sin \frac{2\pi k}{N_m} \right] \end{aligned} \quad (8.21)$$

В свою очередь, из выражения (8.21) можно получить выражения для

$$h_{k,m} - h_{p,m} = (h_{k,m-1} - h_{p,m-1}) + (h_{k+l,m-1} + h_{p+l,m-1}) \sin \frac{2\pi k}{N_m} + (h_{k+l,m-1} + h_{p+l,m-1}) \cos \frac{2\pi k}{N_m} \quad (3.23)$$

$$(h_{k,m} + h_{p,m}) = (h_{k,m-1} + h_{p,m-1}) + (h_{k+l,m-1} + h_{p+l,m-1}) \cos \frac{2\pi k}{N_m} - (h_{k+l,m-1} - h_{p+l,m-1}) \sin \frac{2\pi k}{N_m} \quad (3.22)$$

$$h_{k,m} = h_{k,m-1} + h_{k+l,m-1} \cos \frac{2\pi k}{N_m} + h_{p+l,m-1} \sin \frac{2\pi k}{N_m}$$

действительной и мнимой частей:

Сложив выражения (3.22) и (3.23), можно получить:

Заметим, что  $(p+l) \bmod l \equiv p$ , откуда следует:

$$h_{k+m} = h_{k,m-1} + h_{k+l,m-1} \cdot \cos \frac{2\pi k}{N_m} + h_{p,m-1} \cdot \sin \frac{\pi k}{N_m} \quad (8.24)$$

Проделав аналогичные выкладки, можно получить, что:

$$h_{k+l,m} = h_{k,m-1} - h_{k+l,m-1} \cos \frac{2\pi k}{N_m} - h_{p,m-1} \sin \frac{\pi k}{N_m} \quad (3.25)$$

здесь  $p = l + (l-k) \bmod l$ ,  $N_m = 2^m$ ,  $e = r^{m-1}$ ,  $k=(n) \bmod l$ ,  $n$  - текущий индекс элемента вектора ( $n = 0, N-1$ ).

Таким образом, при  $N_m = 2^m$  “бабочка” БПХ подобна “бабочке” БПФ, но обладает следующим отличием: правило вычисления индекса элемента синусной компоненты иное, чем у индекса косинусного элемента. Индекс же элемента косинусной компоненты и “свободной” компоненты, а также индекса у результирующего элемента и аргументы косинуса и синуса вычисляются так же, как и для “бабочки” БПФ [21].

На рис. 8.7. приведен граф базовой операции бабочка для алгоритма БПХ.

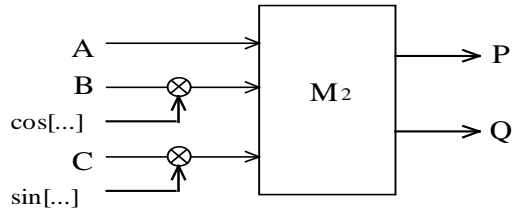


Рис.8.7. Граф базовой операции БПХ, где

$$\begin{cases} P = A + B \cdot \cos[\dots] + C \cdot \sin[\dots] \\ Q = A - B \cdot \cos[\dots] - C \cdot \sin[\dots] \end{cases}$$

Граф алгоритма БПХ для  $N = 2^3$  приведен на рис.8.8.

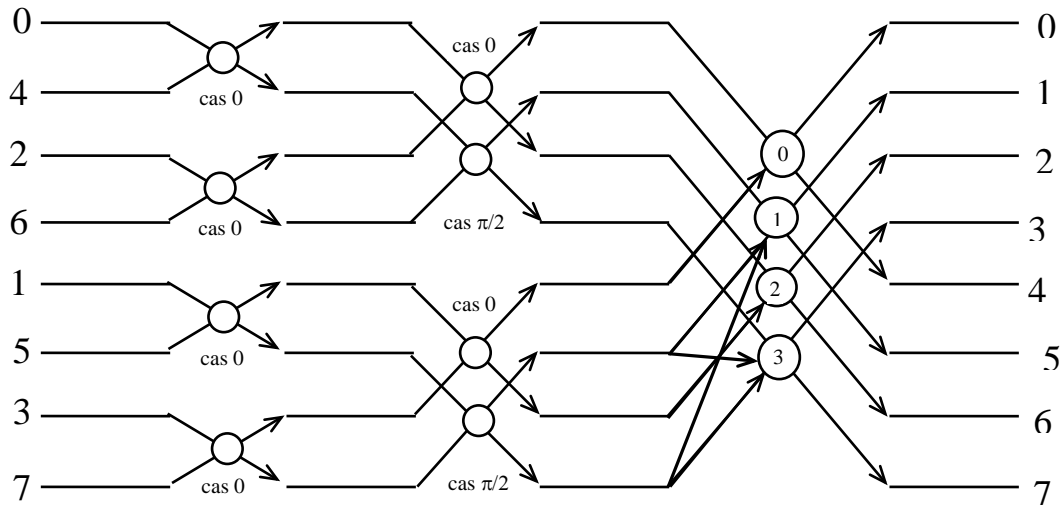


Рис.8.8. Граф алгоритма БПХ

Из представленного графа видно, что на третьей итерации при вычислении  $H_1 \div H_5$  и  $H_3 \div H_7$  в “бабочках” участвуют  $\cos[\dots]$  и  $\sin[\dots]$  компоненты с разными элементами.

### 8.9. Быстрое преобразование Адамара

Как отмечалось в разделе 6.4, матрица ядра Адамара

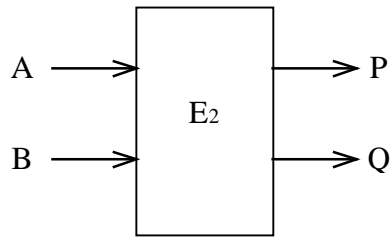
$$A_N = E_2 \otimes E_2 \otimes E \dots \otimes E_2 = \otimes_{i=2}^{M-1} E_2$$

Поэтому алгоритм быстрого преобразования Адамара может быть легко получен на основе алгоритма БПФ<sub>2</sub>, если из него удалить операцию умножения на матрицу D поворачивающих элементов. На основании этого из выражения (3.10) можно получить:

$$\begin{cases} a_{k,m} = a_{k,m-1} + a_{k+l,m-1} \\ a_{k+l,m} = a_{k,m-1} - a_{k+l,m-1} \end{cases} \quad (8.26)$$

где, как и в (8.10)  $l = 2^{m-1}$ ,  $k = (n)_{\text{mod } l}$ ,  $n$  - текущий индекс вектора ( $n = \overline{0, N-1}$ ).

Иначе говоря, “бабочка” в таком алгоритме имеет вид как это показано на рис.8.9



*Рис.8.9. Бабочка алгоритма быстрого преобразования Адамара, где*

$$\begin{cases} P = A + B \\ Q = A - B \end{cases}$$

Такая же точно “бабочка”, как и представленная на рис.7.7, служит основой алгоритмов быстрого преобразования Пэли и Уолша, отличие которых состоит лишь в упорядочении исходных данных перед первой итерацией (см. раздел 6.2).

Граф быстрого алгоритма Уолша – Пэли для  $N = 16$  приведен на рис.8.10. Сложность же алгоритма быстрых ортогональных преобразований Уолша - Адамара в числе БО алгоритма  $Q_{\text{БПФ}}$  и  $Q_{\text{БПХ}}$  для  $N = 2^M$ , однако сложность самой базовой операции значительно меньше и составит только 2 операции сложения, что значительно меньше чем для алгоритма БПХ (не говоря уже о БПФ). Поэтому можно записать, что:

$$Q_{\text{БДОП}} = [N \log_2 N] q_{\text{сл}}.$$



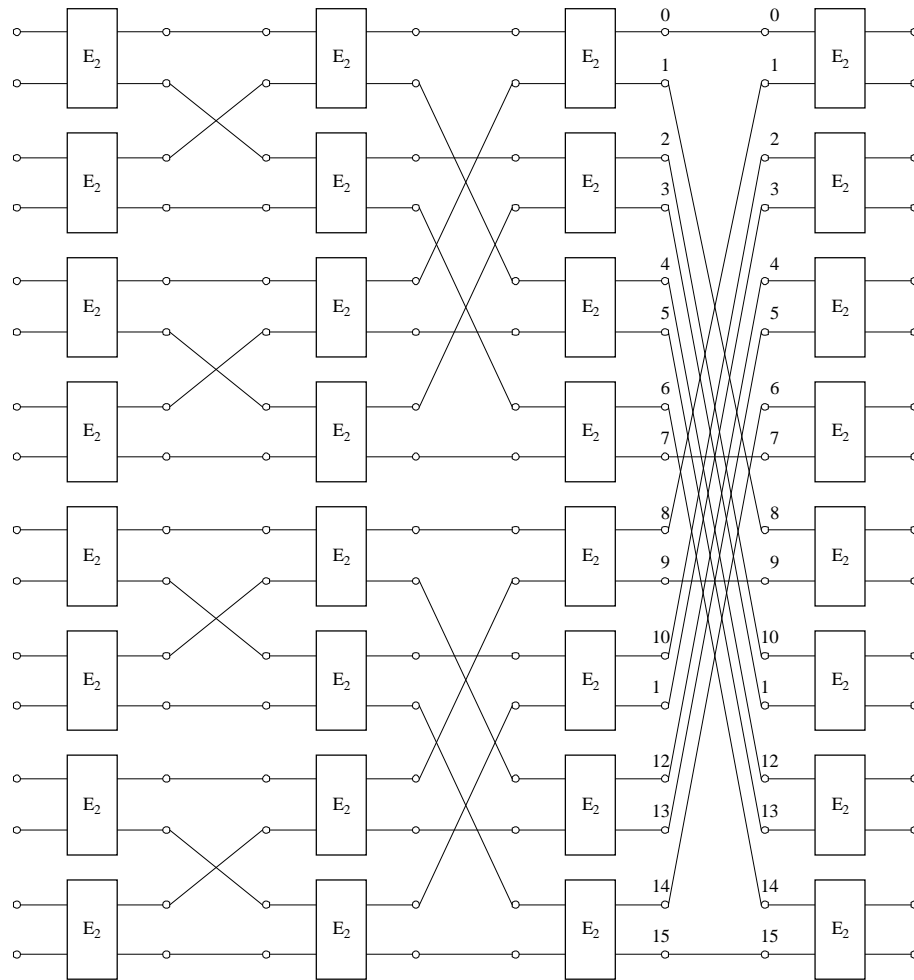


Рис.8.10. Граф алгоритма быстрого преобразования Адамара-Уолша для  $N=16$

Быстрые алгоритмы БДОП Адамара-Уолша являются «обратимыми», т.е. вычисления по графу (см., например, рис.8.10) могут выполняться как «слева направо», так и «справа налево» [6].

Заметим, что для алгоритма БДОП с прореживанием по частоте не выполняется двоично-инверсная перестановка элементов вектора  $X$  перед первой итерацией, но зато необходимо переставлять элементы векторов результата по закону двоичной инверсии.

На рис. 8.11 представлен граф быстрого преобразования Хаара. Нетрудно видеть, что он является частично вырожденным по отношению к графу БДОП Адамара-Уолша.

Сам алгоритм быстрого преобразования Хаара (за исключением итоговой нормировки) будет описываться выражением (8.26), за тем лишь исключением, что при обработке каждого блока данных выполняется только первая базовая операция ( $k=0$ ). Для БП Хаара сложность отдельной базовой операции такая же, как и у БПУ. Однако общее число базовых операций меньше, чем в алгоритмах БПХ и БПУ, и составляет всего  $Q = (2N-1) БО$ .

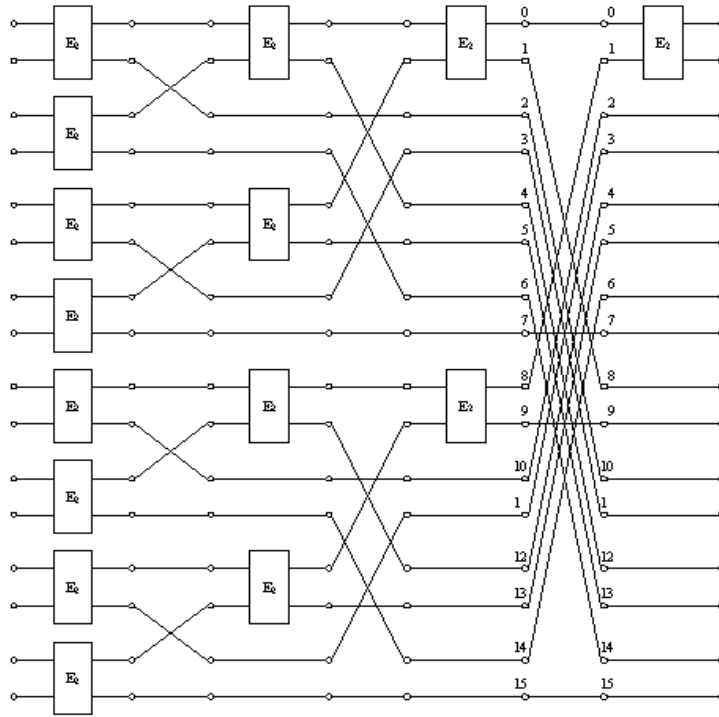


Рис.8.11. Граф быстрого преобразования Хаара для  $N = 16$ .

### 8.10. Выбор метода вычисления свертки / корреляции

Таким образом, в основе работы не рекурсивного фильтра лежит вычисление свертки вектора исходных данных

$$X = [x_0, x_1, x_2, \dots, x_{N-1}]$$

и импульсной характеристики фильтра (ядра свертки)

$$G = [g_0, g_1, g_2, \dots, g_{M-1}],$$

где  $N \gg M$ . Подобные вычисления могут быть вычислены как на основе прямого алгоритма (см. раздел 1.7), так и через спектральные дискретные преобразования Фурье и Хартли (см. разделы 6.1 и 6.2). Возникает естественный вопрос, какой метод предпочтителен, исходя из меньшего объема вычислений, а следовательно, и времени, необходимого для реализации данного метода.

Если размер “окна” или ядра свертки равен  $M$ , а длина вектора исходных данных, то для получения  $N$  отсчетов результата аperiodической свертки необходимо выполнить:

$$Q_{св} = MN (БО), \quad (8.27)$$

где БО – базовая операция, включающая операцию умножения и сложения действительных чисел. Сложность БО составляет:

$$q_{БО} = q_{умн} + q_{сл}.$$

Отсюда можно получить, что время вычисления свертки по прямому алгоритму

$$T_{св} = Q_{св} * \Delta t = MN(q_{умн} + q_{сл}) * \Delta t$$

где  $\Delta t$  - длительность такта В.У.

При этом необходимо отметить, что задаваемые данные и результаты являются действительными числами, т.е. при расчёте прямого алгоритма  $q_{БО}$  определяется суммой операций умножения и сложения действительных чисел.

При вычислении свёртки на основе спектрального преобразования выполняются следующие действия:

1. Прямые спектральные преобразования Фурье или Хартли исходного вектора.
2. При вычислении свёртки на основе ДПХ выполняются перекомпоновка полученного спектра (см. раздел 6.2, т.е. формируется  $\hat{H}_{(\zeta)}$  и вычисляются  $[H_{(\zeta)} + H_{(-\zeta)}]$  и  $[H_{(\zeta)} - H_{(-\zeta)}]$ .
3. Поэлементное перемножение спектров для ДПФ:  $\Phi_{(\zeta)} = H_{(\zeta)} * G_{(\zeta)}$  или вычисление в  $\Phi_{(\zeta)}$  согласно методике в разделе 2.2 - для ДПХ.
4. Обратное преобразование для функции  $\Phi_{(\zeta)}$  на основе ДПФ или ДПХ.

Таким образом, если свёртка при  $N = 2^N$  вычисляется через ДПФ на основе быстрых алгоритмов:

$$\begin{aligned} \mathcal{E}_{св} &= Q_{БПФ}^1 + Q_{умн} + Q_{БПФ}^{-1} = 2Q_{БПФ} + Nq_{умн} = [N \log_2 N]q_{БО} + Nq_{умн} = \\ &= [N \log_2 N](2\hat{\epsilon}_{сл} + \hat{\epsilon}_{умн}) + N\hat{\epsilon}_{умн} \end{aligned} \quad (8.28)$$

где  $q_{БО} = 2\hat{q}_{сл} + \hat{q}_{умн}$

При расчёте  $\hat{Q}_{св}$  свёртки необходимо учитывать, что

$$q_{БО} = 2\hat{q}_{сл} + \hat{q}_{умн} \approx 2(\hat{q}_{сл} + \hat{q}_{умн}),$$

где  $\hat{q}_{сл}$  и  $\hat{q}_{умн}$  - сложность операций сложения и умножения комплексных чисел, причём

$$\begin{aligned} \hat{q}_{сл} &= 2q_{сл} \\ \hat{q}_{умн} &= 4q_{умн} + 2q_{сл} \end{aligned}$$

аналогичных действий с действительными числами, т.е.

$$q_{БО} = 4(q_{умн} + 4q_{сл}) = 4(q_{умн} + q_{сл})$$

Тогда получим

$$\begin{aligned} \hat{Q}_{св} &= [4N \log_2 N](q_{сл} + q_{умн}) + 4N(q_{сл} + q_{умн}) = 4N[1 + \log_2 N](q_{сл} + q_{умн}) \quad (8.29) \\ \hat{T}_{св} &= 4N[1 + \log_2 N](q_{умн} + q_{сл})\Delta t \end{aligned}$$

Отсюда нетрудно получить с учетом выражения (8.27), что прямой метод вычисления свёртки предпочтителен, если:

$$\underline{M < 4(1 + \log_2 N)} \quad (8.30)$$

Из (8.30) следует, например, что для  $N = 1024$  предельный размер  $M = 44$  для вычисления по прямому алгоритму, а для  $N = 128$  предельный размер окна составляет  $M = 32$

Если свёртка вычисляется на основе БПХ, то при том же числе БО их сложность умножается:

$$q_{BO} = 4q_{сл} + 2q_{умн} = 2(q_{умн} + q_{сл})$$

для действительных чисел, однако требуется выполнить операции по вычислению функции  $\hat{\phi}_{(c)}$ , что потребует примерно  $\approx 2N(2q_{сл} + q_{умн})$ . С учетом сложности базовых операций при прямом и спектральном алгоритме можно получить, что прямой алгоритм обладает меньшими вычислительными затратами по сравнению с алгоритмом вычисления свертки через БПХ при условии:

$$M \leq \frac{3}{2}[1 + \log_2 N] \quad (8.31)$$

Что же касается БДОП типа Уолша - Адамара, то базис этих функций для вычисления свёртки мало пригоден, поскольку в отличие от ДПХ для подобных преобразований нет простых формул для связи с ДПФ и не справедлива теорема о свёртке. Известны алгоритмы вычисления свёртки и на основе указанных ДОП [18, 19], однако отсутствие операции умножения при выполнении БДОП не компенсируется сложностью вычисления  $\hat{\phi}$ .

## 9. АЛГОРИТМЫ НЕЛИНЕЙНОЙ ОБРАБОТКИ СИГНАЛОВ

Ранее рассматривались алгоритмы, относящиеся к типу линейных преобразований. Характерная черта этих преобразований состоит в том, что они могут быть описаны в терминах матричной алгебры. При этом, если  $Y=B_N X$  и  $B_N$  является невыраженной, т.е.  $\det[B_N] \neq 0$ , (замети, что такое требование всегда соблюдается для ортогональной матрицы  $B_N$ ), то  $X=B_N^{-1} Y$ .

Это позволяет однозначно перейти от вектора  $Y$ , являющегося результатом преобразования, к вектору  $X$ . Поэтому, если результат обработки сигнала не удовлетворяет по тем или иным соображениям пользования, вновь можно получить исходный сигнал  $X$  и выбрать другое ядро преобразования  $B_N$  или заменить алгоритм.

Базовые операции любых линейных преобразований требуют выполнения операций умножения (деления) и сложения (вычитания) чисел. При выполнении линейных преобразований данные и результаты могут быть как в формате с фиксированной точкой, так и в формате с плавающей точкой. Однако при использовании формата с фиксированной точкой возникает необходимость увеличения разрядности данных в ходе обработки. Поясним это на примере.

Пусть вычисляется аperiodическая свертка с ядром размером  $M=9$  элементов при разрядности исходных данных и коэффициентов ядра, равной 8 бит (причем исходные данные – положительные числа, а коэффициенты ядра знакопеременны):

$$q\{X\}=8; \quad (x \geq 0) \quad q\{G\}=8 \text{ (включая знак)}$$

Вычисление свертки производится по прямому алгоритму:

$$Y_n = \sum_{m=-4}^4 x_{n-m} q_{n-m}$$

В этом случае разрядность любого частичного произведения составляет  $q\{MPL\}=15$ , а с учетом вычисления суммы таких произведений разрядность конечного результата равна  $q\{Y_n\}=15+4=19=20$  бит.

Еще сложнее обстоит дело с увеличением разрядности при использовании формата с фиксированной точкой при выполнении быстрых ортогональных преобразований Фурье и Хартли. Это связано с необходимостью последовательного выполнения операции умножения формируемого результата на различные поворачивающие множители. Поэтому при большой длине вектора разрядность результата становится недопустимо большой. Усечение же его может внести недопустимо большие погрешности или потерять значимость части спектральных коэффициентов при большем динамическом диапазоне сигнала.

Поэтому формат чисел с фиксированной точкой целесообразно использовать для вычисления свертки с малым ядром или для ортогональных преобразований Уолша-Адамара. Для циклической свертки при большом  $N$  и для реализации алгоритмов БПФ и БПХ целесообразно использование формата чисел с плавающей точкой. Однако такой формат существенно усложняет выполнение арифметических операций.

Помимо линейных преобразований для обработки сигналов в настоящее время используются так называемые нелинейные преобразования. К числу нелинейных преобразований в частности, относится [5,16]:

- ранговая фильтрация;
- взвешенная ранговая фильтрация;
- гистограммные преобразования.

Указанные преобразования не могут быть описаны в терминах операций матричной алгебры, хотя, как и операции алгебраической свертки, рекурсивной и не рекурсивной фильтрации, они относятся к группе локальных преобразований и выполняются в “скользящем” режиме при последовательном перемещении окна сканирования размером  $M$  элементов вдоль вектора из  $N$  элементов ( $M \ll N$ ,  $M$  - нечетное).

Вторая характерная особенность указанных нелинейных алгоритмов состоит в том, что они могут быть применены к исходным данным только формата с фиксированной точкой при ограниченной разрядности операндов. Увеличения разрядности результатов при выполнении подобных преобразований не происходит, что существенно упрощает вычисления.

Кроме указанных особенностей, нелинейные преобразования обладают, к сожалению, негативным свойством – они необратимы, т.е. от результата обработки невозможно вернуться к исходным данным. Поэтому при выполнении нелинейной обработки приходится сохранять исходный сигнал во избежание риска его потери до получения удовлетворяющих пользователя результатов.

### 9.1. Ранговая фильтрация

Рассмотрим в начале выполнение ранговой фильтрации в одномерном случае. При ранговой фильтрации для  $i$ -го положения окна сканирования требуется выполнить следующие действия [16]:

1. Все элементы вектора  $X$ , попавшие в окно сканирования, переупорядочиваются в порядке возрастания:

$$\left\{ X_{i-(M/2+1)}; \dots; X_{i-(M/2+1)} \right\} \rightarrow \left\{ \hat{X}_{\min}; \dots; \hat{X}_{\max} \right\}$$

2. В качестве результата  $y_i$  выбирается значение:  $y_i = \hat{X}_R$ , где  $R$  – заданный номер позиции элемента в упорядоченном списке.

Производится смещение окна сканирования на одну позицию:  $i=i+1$

3. Повторяются пункты 1-3.

Очевидно, что если  $R=1$ , то выполняется так называемая минимальная фильтрация, поскольку всегда в качестве результата выбирается наименьший элемент в окне; если же  $R=M$ , - то за результат выбирается максимальный элемент окна.

При  $R=(M+1)/2$  выполняется медианная или срединная фильтрация. При ранговой фильтрации при размере окна  $M$  происходит искажение фильтруемого сигнала - высокочастотные компоненты сигнала с периодом  $<R$  будут пропущены.

Поэтому для того, чтобы сигнал при фильтрации не был подвержен существенному искажению, необходимо, чтобы:

$$T_{\max} \geq R \text{ (при } R = med \rightarrow T_{\min} \geq \frac{M+1}{2} \text{)}. \quad (9.1)$$

Если размер окна относительно невелик, то ранговую фильтрацию можно выполнить посредством упорядочения, т.е. сортировки элементов окна. Однако при больших размерах окна и жестких ограничениях на время фильтрации сортировку следует выполнять только при наличии специальных аппаратных средств - сортирующих сетей [20].

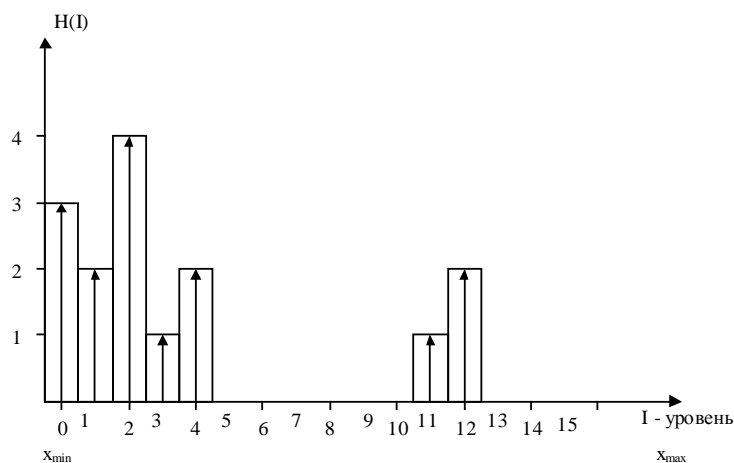
Возможны ещё два алгоритма выполнения РФ:

- гистограммный алгоритм;
- разрядно-срезовой алгоритм.

При гистограммном алгоритме строится гистограмма  $H(I)$ , т.е. распределение числа элементов окна по уровням (т.е. по значениям):

$$R \leq \sum_{I=0} H(I) \quad (9.2)$$

При выполнении ранговой фильтрации согласно (9.2) строится гистограмма и вычисляется её площадь (сумма значений), до тех пор, пока число элементов учтённых уровней в гистограмме не превысит значение ранга  $R$ . Последний учтённый уровень и определяет значение элемента заданного ранга. На рис.9.1. показан пример нахождения элемента заданного ранга по построенной гистограмме.



$$\begin{aligned} R_1 = 5 &\rightarrow x = 1 \\ R_2 = 7 &\rightarrow x = 2 \\ R_3 = 10 &\rightarrow x = 3 \\ R_4 = 12 &\rightarrow x = 4 \\ R_5 = 13 &\rightarrow x = 11 \end{aligned}$$

Рис. 9.1. Гистограмма распределения элементов по уровням

Гистограммный алгоритм ранговой фильтрации для окна размером  $M \times M$  может быть представлен в следующем виде [16,21]:

начало;

$H(I) := 0$ ; для  $\forall i = 0, 2^Q - 1$

для  $m = 1, M$  цикл:

для  $n = 1, M$  цикл:

$I := B_{mn}$ ;  $H(I) := H(I) + 1$ ;

конец цикла по  $n$ ;

конец цикла по  $m$ ;

$D := 0$ ;

для  $i = 0, 2^Q - 1$  цикл:

если  $D \geq R$  то выход из цикла;

иначе  $D := D + H(i)$ ;

конец цикла по  $i$ ;

$I_R := i$ ;

конец

Здесь  $B_{mn}$  - значение отсчета с номером  $(m,n)$ ,  $Q$  - разрядность отсчетов. Поскольку фильтрация выполняется в скользящем режиме, то при перемещении окна вдоль строки изображения достаточно модифицировать имеющуюся гистограмму предшествующего положения окна, включив в нее элементы нового столбца и исключив элементы, соответствующие "ушедшему" столбцу. Модификация гистограммы сводится к следующим операциям:

- выполнить  $M$  раз вычитание  $H(I) := H(I) - 1$  для элементов  $I = B_{mn}^i$  столбца, выходящего из окна сканирования;
- выполнить  $M$  раз сложение  $H(I) := H(I) + 1$  для элементов  $I = B_{mn}$  нового столбца полосы.

Очевидно, что такой подход позволяет значительно сократить время формирования гистограммы. Дальнейшее сокращение времени фильтрации может быть достигнуто путем сокращения времени поиска элемента заданного ранга. Для этого оказывается удобным строить не одну, а  $Q$  гистограмм, причем  $H_1(I)$  строится только по старшему разряду,  $H_2(I)$  - по двум старшим разрядам и так далее, до  $H_Q(I)$ , которая строится по всем разрядам элементов окна. В результате поиск по  $Q$  гистограммам организуется как поиск по  $Q$  - уровневому бинарному дереву. Поиск результата начинается с первой гистограммы  $H_1(I)$ , по которой определяется старший разряд. Анализ каждой следующей гистограммы уточняет очередной младший разряд результата, в результате за  $Q$  циклов будет определен  $Q$ -разрядный результат фильтрации.

Разрядно-срезовой алгоритм РФ может быть интерпретирован как поиск необходимого значения по двоичному дереву и сведен к следующему ( пусть разрядность данных – 4 бита) :

1. Анализируем биты старшего среза. Если "0" больше, чем  $R$ , то это значит, что элемент ранга  $R$  лежит среди отсчетов с нулевым старшим битом;



2. Анализируем второй срез, исключив при его рассмотрении те отсчеты  $x_m$ , для которых старший бит  $=1$  (т.е. модифицируем второй срез). Если “0” во втором срезе больше, чем  $R$ , то это значит, что элемент заданного ранга лежит среди отсчетов со старшими битами “00”;
3. Анализируем третий срез, исключив при его рассмотрении те отсчеты  $x_m$ , у которых во втором модифицированном срезе бит  $=1$ , т.е. у которых старшие два бита равны “01”. Если “0” в третьем срезе больше, чем  $R$ , то это значит, что элемент заданного ранга лежит среди отсчетов со старшими битами “000”;
4. Анализируем четвертый срез, исключив при его рассмотрении те отсчеты  $x_m$ , у которых старшие биты равны “001”. Если “0” в младшем срезе больше, чем  $R$ , то это значит, что элемент заданного ранга лежит среди отсчетов со старшими битами “0000”, иначе среди тех, у которых младший бит  $=1$ , т.е.  $x_m = “0001”$ .

Заметим, что “1” в последнем младшем срезе указывает и местоположение, и число элементов заданного ранга

Подобный алгоритм получил название разрядно-срезового алгоритма со сквозным маскированием. Очевидно, что отдельно взятый разрядный срез не позволяет определить искомый элемент  $I_R$ . Для этого необходимо дополнительно иметь результаты анализа на предыдущих итерациях поиска. Поэтому в разрядно - срезовом алгоритме требуется, чтобы при переходе к анализу текущего разрядного среза были известны все элементы, исключенные из поиска на предыдущих этапах, т.е. требуется задание срезовой маски. В результате обработки текущего разрядного среза определяется очередной разряд результата  $I_R$  и маска, содержащая всю информацию об исключенных элементах списка на  $q$ -ом этапе поиска.

Запишем подобный алгоритм (для фиксированного положения окна) [21]:

*начало:*

$$D_0 = -R;$$

для  $q = 1, Q$  цикл:

$$n_0 := 0;$$

для  $m = 1, M$  цикл:

для  $n = 1, M$  цикл:

$$\text{если } b_{mn} = 1 \text{ то } n_0 := n_0,$$

$$\text{иначе } n_0 := n_0 + 1;$$

конец цикла по  $n$ ;

конец цикла по  $m$ ;

$$D_q := D_{q-1} + n_0;$$

если  $D_q \geq 0$ , то

{

$d_q := 0,$   
 для  $k = q, Q - 1$  цикл:  
     для  $m = 1, M$  цикл:  
         для  $n = 1, M$  цикл:  
              $b_{mn, k+1} := b_{mn, k+1} \cup b_{mn, q};$   
             конец цикла по  $n$ ;  
         конец цикла по  $m$ ;  
     конец цикла по  $k$ ;  
      $D_q := D_{q-1};$

иначе }  
 {

$d_q := 1;$   
 для  $k = q, Q - 1$  цикл:  
     для  $m = 1, M$  цикл:  
         для  $n = 1, M$  цикл:  
              $b_{mn, k+1} := b_{mn, k+1} \cup \bar{b}_{mn, q};$   
             конец цикла по  $n$ ;  
         конец цикла по  $m$ ;  
     конец цикла по  $k$ ;  
 }

конец цикла по  $q$ ;  
 $I_R := d_1 d_2 d_3 \dots d_Q;$

конец.

Недостатком алгоритма со сквозным маскированием является необходимость после анализа текущего среза модифицировать все последующие, что резко увеличивает объем вычислений и практически не позволяет распараллелить вычисления по срезам. Указанный недостаток преодолевается путем ввода маски, воздействующей только на один последующий срез и не изменяющей другие разрядные срезы.

Такая маска  $S_q$  может быть найдена в соответствии с выражением [21]:

$$S_q = \overline{S_{q-1} \& (B_q \# d_{q-1})}, \quad (9.3)$$

где  $B_q$  -  $q$ -й разрядный срез элементов окна, а символ  $\#$  обозначает логическую операцию неравнозначности. Накапливаемая сумма  $D_q$  определяется в этом случае из соотношения:

$$D_q = \begin{cases} D_{q-1} + \Delta D_q, & d_{q-1} = 1. \\ D_{q-1} - \Delta D_q, & d_{q-1} = 0. \end{cases} \quad (9.4)$$

$$\Delta D_q = \sum S_q \& (B_{q-1} \# B_q)$$

Такой подход и является основой алгоритма с последовательным маскированием. Таким образом, для обработки последующего разрядного среза  $B_q$  необходимо сформировать маску  $S_q$  и передать предыдущий разрядный срез  $B_{q-1}$ .

Возможны три варианта процедуры разрядно-срезовой ранговой фильтрации: четный, нечетный и четно-нечетный алгоритмы. На каждой  $q$ -й итерации четного алгоритма анализируются элементы  $q$ -го разрядного среза, имеющие четные значения, т.е. элементы,  $q$ -ый разряд которых равен нулю. Для нечетного алгоритма на  $q$ -й итерации учитываются те элементы, соответствующий  $q$ -й разряд которых равен единице, а в четно-нечетном алгоритме на различных итерациях могут учитываться как четные, так и нечетные элементы.

В разрядно-срезовых алгоритмах каждый разрядный срез должен быть представлен как битовая строка длиной  $M^2$  бит, что при размере окна более чем  $5 \times 5$  превышает разрядность большинства существующих операционных элементов. Поэтому в разрядно-срезовых алгоритмах следует предусмотреть возможность обработки разрядных срезов по фрагментам (например по столбцам). В этом случае обрабатывается вектор длиной лишь  $M$  бит.

Модифицированный четный разрядно-срезовой алгоритм ранговой фильтрации, позволяющий оптимизировать конвейерный режим вычислений с возможностью обработки срезов по столбцам, может быть записан в следующем виде [21]:

*Начало :*

$$D_0 := -R; I_R = 00000\dots 0; S_0 := 11111\dots 11;$$

*цикл для  $q = 1, Q$ :*

$$D_q := D_{q-1};$$

$$\Delta := 0;$$

*цикл для  $i = 1, M$*

$$\Delta := \Delta + \sum_q^1 (S_{q-1}^i \& \bar{B}_q^i);$$

*конец цикла по  $i$*

$$D_q := D_q + \Delta;$$

*если  $D_q \geq 0$ , то*

$$\{ d_q := 0, D_q := D_{q-1};$$

*цикл для  $i = 1, M$*

$$S_q^i := S_{q-1}^i \& \bar{B}_q^i$$

*конец цикла по  $i$  };*

иначе  $\{d_q := 1,$   
цикл для  $i = 1, M$   
 $S_q^i := S_{q-1}^i \& B_q^i$   
конец цикла по  $i \}$ ;

конец цикла по  $q$ ;

$I_R := d_1 d_2 d_3 \dots d_Q$ ;

конец.

Здесь  $Q$  - разрядность данных;  $S_q$  -  $q$ -я разрядно-срезовая бинарная маска ;  
 $B_q$  -  $q$ -й разрядный срез матрицы данных ;  $\sum_q^1$  - число единичных бит в  
маскированном разрядном срезе  $\bar{B}_q$ ,  $d_q$  - вычисленное значение  $q$ -го разряда  
результата;  $I_R$  - результат ранговой фильтрации.

Следует отметить, что при использовании подобного алгоритма может  
быть определена не только величина элемента заданного ранга  $R$ , но и его  
местоположение в окне. Информация о местоположении элемента  $I_R$   
содержится в бинарной маске  $S_q$ , сформированной на последней итерации  
алгоритма. Положение единичного бита (или единичных бит) определяет  
положение элементов заданного ранга  $R$  в окне сканирования.

Рассмотрим примеры выполнения ранговой фильтрации по разрядно-  
срезовому алгоритму со сквозным маскированием.

1. Пусть  $M \times M = 5 \times 5$ ;  $Q = 4$  и надо определить элемент ранга  $R = 13$   
(выполняется медианная фильтрация)

9	7	6	5	8
10	11	7	6	3
1	2	13	7	5
1	5	14	12	4
2	13	15	14	6

$B_1 =$   
(старшие  
разряды  
пикселей)

1				1
1	1			
		1		
		1	1	
	1	1	1	

$B_2 =$

	1	1	1	
		1	1	
		1	1	1
	1	1	1	1
	1	1	1	

0)  $S_0 = 11 \dots 1111$ ;

$D_0 = -13$ ;

$Y_{i0} = I_R$ ;

1)  $\Sigma_1 = S_0 \& \bar{B}_1 = 15$ ;

$D_1 = -13 + 15 = 2$ ;  $D_1 > 0$ ;  $d_1 = 0$ ;

$D_1 = -13$ ;  $S_1 = S_0 \& B_1$ ;

2)  $\Sigma_2 = S_1 \& \bar{B}_2$ ;

$D_2 = -13 + 6 = -7$ ;

$D_2 = -7$ ;  $d_2 = 1$ ;  $D_2 < 0$ ;

$D_2 = -7$ ;  $S_2 = S_1 \& \bar{B}_2$ ;

$B_3 =$

	1	1		
1	1	1	1	1
	1		1	
		1		
1		1	1	

$B_4 =$   
(младшие  
разряды  
пикселей)

1	1	1		
	1	1	1	1
		1	1	1
1	1			
1	1			

Элементы  
13-го ранга

3)  $\Sigma_3 = S_2 \& \bar{B}_3$ ;

$D_3 = -7 + 4 = -3$ ;  $D_3 = -3$ ;  $d_3 = 1$ ;  $D_3 < 0$ ;

$S_3 = S_2 \& B_3$ ;

4)  $\Sigma_4 = S_3 \& \bar{B}_4$ ;

$D_4 = -3 + 2 = -1$ ;  $d_4 = 1$ ;

$S_4 = S_3 \& \bar{B}_4$ ;

“1” в S4 показывает, где расположен  $I_R$ .

2. Пусть для того же окна надо найти элемент ранга  $R=18$ .

$R=18$ ;  $I_{R18}=10$ .

$B_1 =$   
(старшие  
разряды  
пикселей)

1				1
1	1			
		1		
		1	1	
	1	1	1	

$B_2 =$

	1	1	1	
		1	1	
		1	1	1
	1	1	1	1
	1	1	1	

0)  $S_0=11\dots1111$ ;

$D_0=-18$ ;

1)  $\Sigma_1=S_0 \& \bar{B}_1=15$ ;

$D_1=-18+15=-3$ ;  $D_1<0$ ;  $d_1=1$ ;

$D_1=-3$ ;  $S_1=S_0 \& B_1$ ;

2)  $\Sigma_2=S_1 \& \bar{B}_2=4$ ;

$D_2=-3+4=1$ ;

$d_2=0$ ;  $D_2>0$ ;  $D_2=-3$ ;

$S_2=S_1 \& \bar{B}_2$ ;

$B_3 =$

	1	1		
1	1	1	1	1
	1		1	
		1		
1		1	1	

$B_4 =$   
(младшие  
разряды  
пикселей)

1	1		1	
	1	1		1
		1	1	1
1	1			
	1	1		

3)  $\Sigma_3=S_2 \& \bar{B}_3=2$ ;

$D_3=-3+2=-1$ ;  $D_3<0$ ;  $D_3=-1$ ;  $d_3=1$ ;

$S_3=S_2 \& B_3$ ;

4)  $\Sigma_4=S_3 \& \bar{B}_4=1$ ;

$D_4=-1+1=0$ ;  $d_4=0$ ;  $D_4=0$ ;

$S_4=S_3 \& \bar{B}_4$ ;

“1” в S4 показывает, где расположен  $I_R$ .

## 9.2. Взвешенная ранговая фильтрация

Дальнейшим развитием метода ранговой фильтрации является процедура взвешенной ранговой фильтрации (ВРФ). При выполнении ВРФ задаётся вектор  $Z_M$  (вектор взвешивания). Каждый элемент  $Z_m$ ,  $m \in \overline{1, M}$  указывает, сколько раз должен быть повторён элемент вектора, занимающий ту же позицию в окне сканирования (в неупорядоченном). Поэтому процедура ВРФ состоит из 2-х этапов [16]:

1. “Взвешивание” элементов с помощью весовых коэффициентов;

2. Поиск элемента с рангом R.

При этом наиболее просто выполнение указанных процедур осуществляется разрядно-срезовыми алгоритмами. Разрядно-срезовой алгоритм взвешенной ранговой фильтрации аналогичен алгоритму ранговой фильтрации, за исключением операции модификации текущей суммы, которая принимает вид :

$$D_q = D_{q-1} + \sum_m \sum_n z_{mn} (s_{mn}^{q-1} \& \bar{b}_{mn}^q) \quad (9.5)$$

Таким образом, ранговая фильтрация может рассматриваться как частный случай взвешенной ранговой фильтрации, когда  $z_{mn} = 1$  для всех значений  $m$  и  $n$ . Если  $z_{mn} = \{0; 1\}$ , то выполняется процедура ранговой фильтрации с произвольной формой окна.

Согласно литературе, ВРФ приводит к меньшим искажениям мелких деталей изображений.

### 9.3. Скользящая эквализация гистограмм

Скользящая эквализация гистограмм является процедурой, обратной по своему алгоритму процедуре ранговой фильтрации. Элемент преобразованного изображения при скользящей эквализации определяется рангом центрального элемента окна исходного изображения.

Выполнение процедуры скользящей эквализации при использовании гистограммного алгоритма сводится к вычислению следующей суммы [16]:

$$R = \sum_{q=1}^{B_{\Pi}} H(q) \quad (5.6)$$

Здесь  $H(q)$  -  $q$ -ый отсчет гистограммы,  $R$  - ранг центрального элемента окна  $B^{\Pi}$ .

Алгоритм ранговой фильтрации при незначительном видоизменении базовой операции может быть сведен к алгоритму скользящей эквализации гистограмм. При выполнении такой процедуры элемент преобразованного изображения определяется рангом центрального элемента окна исходного изображения.

Однако важное отличие процедуры скользящей эквализации от ранговой и взвешенной ранговой фильтрации заключается в увеличении размеров окна сканирования. При обработке двумерных сигналов (изображений) размер окна сканирования выбирается из условия

$$M^2 \geq 2^Q,$$

где  $Q$  – разрядность отсчетов сигнала (желательно при этом выбирать ближайшее значение  $M$ , при котором соблюдается указанное условие).

Разрядно-срезовой алгоритм обратной ранговой фильтрации с послынным маскированием приобретает вид (для некоторого произвольного положения окна) [16]:

*Начало :*

$$D_0 := -M^2;$$

$$b^M := d^1 d^2 d^3 \dots d^M;$$

$$S_0 := 1111 \dots 1;$$

для  $q = 1, Q$  цикл: )

$$\Delta D := \sum_q^1 \bar{B}_q \& S_{q-1};$$

$$\text{если } d_q^u = 0, \text{ то } \left\{ D_q := D_{q-1} + \Delta D; S_q := S_{q-1} \& \bar{B}_q \right\}$$

$$\text{иначе } \{ D_q := D_{q-1}; S_q := S_{q-1} \& B_q \};$$

конец цикла по  $q$ ;

$$R^H := \text{abs}(D_Q);$$

конец.

Здесь  $B^H$  - центральный элемент окна изображения. Заметим, что алгоритмы обратной ранговой фильтрации могут быть получены на базе алгоритмов как обычной, так и взвешенной ранговой фильтрации.

Ниже приведены примеры выполнения скользящей эквализации гистограмм.

**Пример1.** Выполнение эквализации на основе гистограммного алгоритма (рис. 5.2).

9	7	6	5	8
10	11	7	6	3
1	2	13	7	5
1	5	14	12	4
2	13	15	14	0

$B_{cp}$

$$I_R = 13$$

$$R_{13} = 22$$

$$y_{ij} = R_{13} = 22$$

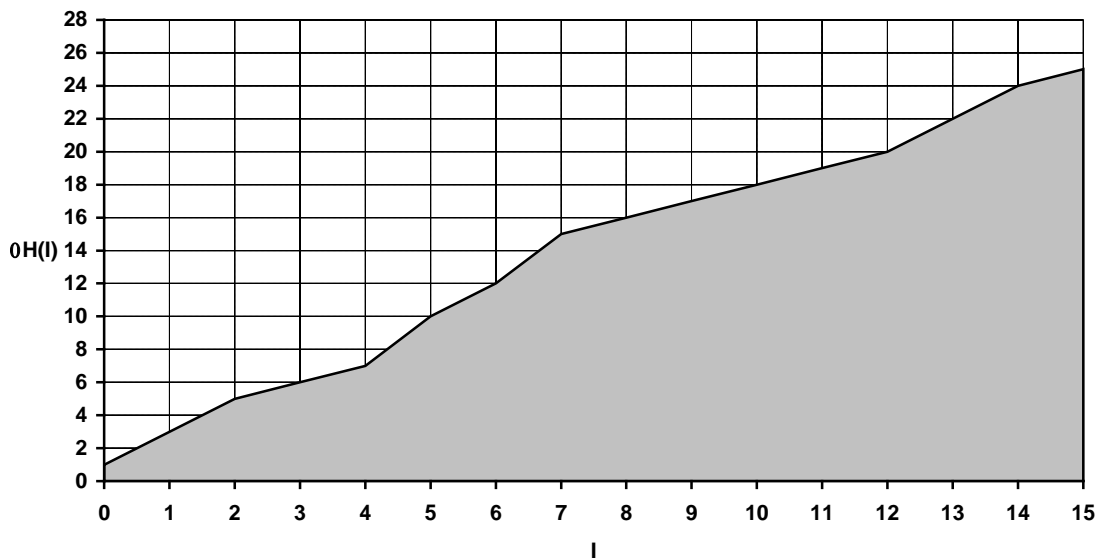
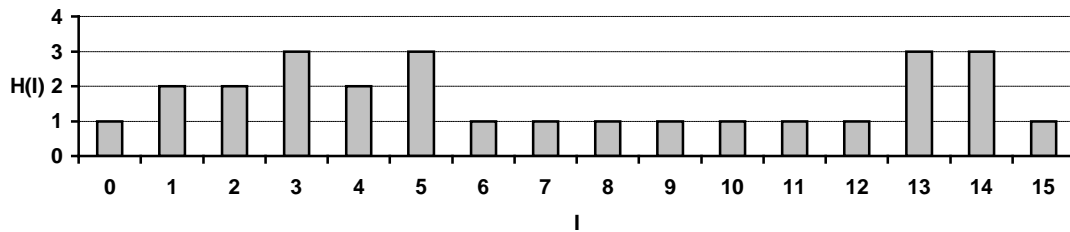


Рис. 9.2. Скользящая эквализация гистограмм по гистограммному алгоритму

**Пример 2.** Эквиализация на основе разрядно-срезового алгоритма с послойным маскированием.

9	7	6	5	8
10	11	7	6	3
1	2	13	7	5
1	5	14	12	4
2	13	15	14	0

0)  $D_0 = -25$   
 $b^u = 1101$   
 $S_0 = 111...111$

1)  $D_1 = -25 + 10 = -15$   
 $b_1 = 1; D_1 = -25$   
 $S_1 = S_0 \& B_1$

$B_1 =$

1				1
1	1			
		1		
		1	1	
	1	1	1	

2)  $D_2 = -25 + 6 = -19$   
 $b_2 = 1; D_2 = -25$   
 $S_2 = S_1 \& B_2$

$B_2 =$

	1	1	1	
		1	1	
		1	1	1
	1	1	1	1
	1	1	1	

$B_3 =$

	1	1		
1	1	1	1	1
	1		1	
		1		
1		1	1	

$B_4 =$

1	1		1	
	1	1		1
1		1	1	1
1	1			
	1	1		

3)  $D_3 = -25 + 3 = -22$   
 $b_3 = 0; D_3 = -22$   
 $S_3 = S_2 \& B_3$

4)  $D_4 = -22 + 2 = -20$   
 $b_4 = 1; D_4 = D_3$   
 $S_4 = S_2 \& B_3$

$R = |D_4| = 22$

### 9.4. Преобразование гистограмм распределения

Процедура модификации гистограммы обеспечивает изменение параметров гистограммы (глобально или локально) по различным законам [16]:



- по экспоненциальному закону (задается параметр  $\alpha$  - показатель степени экспоненты);
- по закону гиперболической функции;
- по закону распределения Рэлля или распределения степени  $2/3$ ;
- возможна также эквализация (лианеризация) и степенная интенсификация гистограммы.

Суть подобной процедуры иллюстрируется ниже на рис.9.3. На рис.9.3,а) приведена первоначальная гистограмма распределения яркостей (амплитуд) отсчетов изображения (пунктиром показаны желаемые распределения). На рис.9.3,б) - гистограмма распределения яркостей после ее модификации для результирующего изображения.

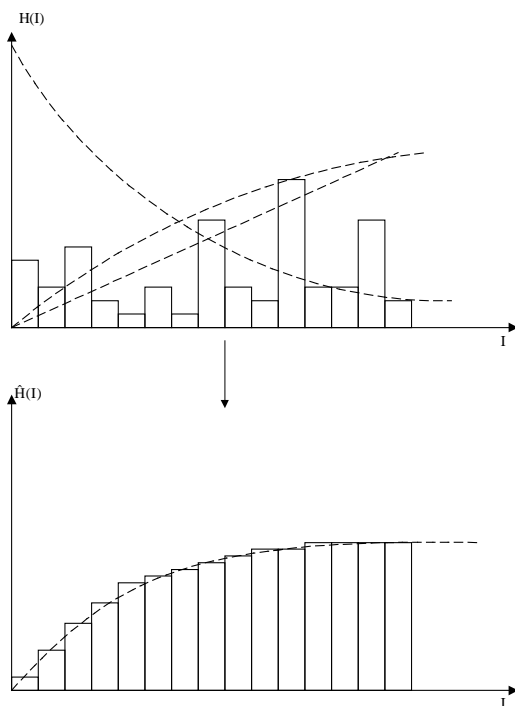


Рис.9.3. Глобальная эквализация гистограмм

При выполнении процедуры глобальной эквализации производится выравнивание гистограммы уровней яркости изображения на основе формирования гистограммы всего изображения.

$$H = [h_g], \quad g = 0, 255.$$

Преобразование выполняется в два этапа [21]:

1. Вычисляется таблица преобразования  $T = [t_g]$ , обеспечивающая выравнивание гистограммы  $H$  изображения:

$$t_k = \sum_{i=0}^k h_g, \quad k = 0, 255.$$

2. Элементы изображения преобразуются по таблице  $T$  и нормируются для приведения яркостей результирующего изображения в диапазон  $[0,255]$ :

$$y_{ij} = (255 * T[a_{ij}]) / (I * J),$$

где  $I \times J$  - размеры изображения  $A$ .

## Контрольные вопросы и задания.

### Разделы 1-3.

1. Построить функцию пропускания фильтра высоких частот с граничной частотой  $f_{гр} = 100$  кГц, при условии, что число отсчетов спектра  $N=100$  и разрешение по частоте  $\Delta f = 20$  кГц. Привести рисунок.
2. Каким образом могут быть представлены в частотной области периодические сигналы?
3. Что позволяет определить корреляционная функция двух детерминированных сигналов? Каковы ее свойства?
4. Как изменится полоса частот сигнала при уменьшении его длительности в 10 раз?
5. Укажите основные методы повышения помехоустойчивости при приеме сигналов.
6. Статистически независимые случайные величины являются некоррелированными. Верно ли обратное утверждение? Ответ поясните.
7. Несет ли усредненная спектральная плотность случайного процесса информацию о собственно случайной составляющей такого процесса?. Ответ поясните.
8. Каким образом можно отфильтровать низкочастотную помеху?
9. Поясните взаимосвязь между корреляционной функцией случайного процесса и его спектральной плотностью мощности?
10. Каким образом может быть определен спектральный состав периодического сигнала?
11. Возможно ли точное восстановление фазы, амплитуды и частоты дискретного гармонического сигнала, если его частота дискретизации равна частоте Найквиста?
12. Каким образом могут быть определены параметров квантователя при равномерном квантовании отсчетов сигнала?

### Раздел 4

1. Сигнал имеет частотный спектр, ограниченный частотой  $F_{max} = 10$  КГц, причем разрешение по частоте составляет 100 Гц. В течении какого промежутка времени должен наблюдаться сигнал? Через какие промежутки времени должны сниматься отсчеты сигнала?

2. Сигнал наблюдается в течении 10 сек., причем отсчеты сигнала снимаются через 10 мксек. Какова предельная частота сигнала  $F_{max}$  может быть зафиксирована. Какое разрешение по частоте будет обеспечиваться в этом случае?

3. Заданы последовательности  $G = [0; 1; 2]$  и  $X = [0; 1; 2]$ . Вычислить аперриодическую свертку и корреляцию. Выполнить оценку вычислительной сложности разрядно-срезового алгоритма свертки/корреляции в сравнении с вычислительной сложностью прямого алгоритма свертки/корреляции.

4. Задан вектор  $X = [0, 0, 1, 1, 2, 3, 2, 1, 0, 1, 0, 0]$ . Определить вектор  $Y$  с отсчетами отфильтрованного сигнала при использовании рекурсивного линейного фильтра с коэффициентами  $H = [1, 3, 1]$  и  $B = [-1/2, 1]$  ("краевыми эффектами пренебречь").

5. Реализация линейных пространственных фильтров требует перемещения центра маски по изображению и вычисления, для каждого из положений маски, суммы произведений коэффициентов маски на значения соответствующих пикселей. В случае низкочастотной фильтрации все коэффициенты равны 1, и можно использовать *однородный усредняющий фильтр* или алгоритм *скользящего среднего*, основанный на том, что при переходе от точки к точке обновляется только часть вычисляемых элементов.

(а) Сформулируйте такой алгоритм для фильтра размерами  $n \times n$ , демонстрирующий характер взаимосвязи вычислений с последовательностью сканирования, использующейся при передвижении маски по изображению.

(б) Отношение числа операций, требуемых для реализации метода «в лоб» к числу операций, используемых алгоритмом скользящего среднего называется *эффективностью алгоритма*. Подсчитайте эффективность алгоритма для данного случая и изобразите ее в виде графика зависимости от  $n$  для  $n > 1$ . Коэффициент  $1/n^2$  является общим для обоих случаев, и поэтому не должен приниматься во внимание. Считайте, что изображение окружено бордюром из нулей достаточной ширины, чтобы не учитывать влияние граничных эффектов при вычислениях.

### Разделы 5 и 6

1. Проверить, является ли ортогональным ядро преобразования для  $N = 4$

$$\begin{aligned} \text{а) } f_k &= \sum x_n (-1)^{kn} \\ \text{б) } f_k &= \sum x_n \cos[2\pi kn/N] \\ \text{в) } f_k &= \sum x_n \sin[2\pi kn/N] \end{aligned}$$

2. Покажите, что преобразование Фурье и обратное преобразование суть линейные операции (относительно линейности см. Раздел 2.6).

3. Пусть  $F(u, v)$  — ДПФ изображения  $f(x, y)$ . Из обсуждения в Разделе 4.2.3 нам известно, что умножение  $F(u, v)$  на функцию фильтра  $H(u, v)$  и вычисление обратного преобразования Фурье изменит вид изображения в соответствии с характером используемого фильтра. Пусть  $H(u, v) = A$ , где  $A$  — некоторая положительная константа. Используя теорему о свертке, объясните математически, почему элементы изображения в пространственной области умножаются на ту же самую константу.

4. Постройте перестановочную матрицу, с помощью которой можно было бы сгруппировать вначале нечетные элементы вектора, а затем четные (и те и другие - в порядке возрастания индексов) для  $N = 8$ .

5. Выполнить двоично-инверсную перестановку вектора данных

$$X = [x_0; x_1; x_2; x_3; \dots; x_{N-2}; x_{N-1}]^T \quad (N = 16)$$

6. Выполнить перестановку вектора данных

$$X = [x_0; x_1; x_2; x_3; \dots; x_{N-2}; x_{N-1}]^T$$

с троичным прореживанием ( $N = 9$ ).

## Раздел 5

1. Построить функцию пропускания фильтра низких частот с граничной частотой  $f_{гр} = 750$  КГц, при условии, что число отсчетов спектра  $N=100$  и разрешение по частоте  $\Delta f = 50$  КГц. Привести рисунок.

2. Задан вектор  $X = [0, 0, 1, 1, 2, 3, 2, 1, 0, 1, 0, 0]$ . Определить вектор  $Y$  с отсчетами отфильтрованного сигнала при использовании рекурсивного линейного фильтра с коэффициентами  $H = [1, 3, 1]$  и

$B = [-1/2, 1]$  ("краевыми" эффектами пренебречь).

3. Задана матрица взаимокорреляции:

$$R = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & -1 \\ -2 & -2 & 2 \end{bmatrix}$$

Найти для нее обратную матрицу.

4. В одном из промышленных приложений рентгеновская съемка используется для контроля внутренней части некоторой сложной отливки. Целью является обнаружение пустот в отливке, которые обычно выглядят как маленькие пятна на изображении. Однако, из-за особенностей материала отливки и уровня энергии рентгеновских лучей, возникает высокий уровень шума, часто затрудняющий процесс контроля. В качестве решения проблемы используется усреднение серии изображений для подавления шума и тем самым улучшения видимых контрастов. При вычислении среднего важно максимально уменьшить число изображений, чтобы сократить общее время экспозиции, в течение которого деталь должна оставаться неподвижной. После многочисленных экспериментов было выяснено, что достаточным является уменьшение дисперсии шума в 10 раз. Если устройство ввода изображений работает с частотой 30 кадров в секунду, как долго отливка должна оставаться неподвижной при съемке, чтобы достичь требуемого уменьшения дисперсии шума? Считайте шум некоррелированным и имеющим нулевое среднее.

5. Исследуйте предельные эффекты многократного применения низкочастотного сглаживающего фильтра размерами  $3 \times 3$  к дискретному изображению. Можете игнорировать влияние границ изображения.

6. Покажите, что если передаточная функция фильтра  $H(u, v)$  вещественная и центрально-симметричная, то и соответствующий пространственный фильтр  $h(x, y)$  также вещественный и центрально-симметричный.

7. Пусть дано изображение размерами  $M \times N$ , и проводится эксперимент, состоящий в последовательном применении к изображению процедур низкочастотной фильтрации с использованием одного и того же гауссова фильтра низких частот с заданной частотой среза  $D_0$ . Ошибкой округления можно пренебречь. Пусть  $k_{min}$  — наименьшее положительное число, представимое в вычислительной машине, на которой проводится эксперимент.

(а) Обозначим через  $K$  число произведенных в эксперименте процедур фильтрации. Можете ли Вы предсказать, без проведения эксперимента, каков будет его результат (изображение) при достаточно большом значении  $K$ ?

(б) Получите выражение для *минимального* значения  $K$ , гарантирующего получение предсказанного результата.

8. Предположим, что Вы сформировали низкочастотный пространственный фильтр, действие которого в каждой точке  $(x, y)$  сводится к усреднению значений в четырех ближайших к ней точках, исключая ее саму.

(а) Найдите эквивалентный фильтр в частотной области.

(б) Покажите, что это действительно низкочастотный фильтр.

9. Предположим, что Вы имеете набор изображений, полученных в результате астрономических наблюдений. Каждое изображение состоит из множества ярких широко разбросанных точек, соответствующих звездам в малонаселенной части Вселенной. Проблема заключается в том, что звезды едва различимы на фоне дополнительного освещения, возникающего в результате рассеяния света в атмосфере. Исходя из модели, согласно которой изображения представляют собой суперпозицию постоянной яркостной составляющей и множества импульсов, предложите основанную на гомоморфной фильтрации процедуру для выявления составляющих изображения, которые непосредственно связаны со звездами.

10. Опытному медицинскому эксперту поручено просмотреть некоторую группу изображений, полученных при помощи электронного микроскопа. Для того чтобы облегчить себе задачу, эксперт решает воспользоваться методами цифровой обработки изображений. С этой целью он исследует ряд характерных изображений и сталкивается со следующими трудностями. (1) Наличие на изображениях отдельных ярких точек, не представляющих интерес. (2) Недостаточная резкость изображений. (3) Недостаточный уровень контрастности некоторых изображений. И, наконец, (4) сдвиг среднего уровня яркости, который для корректного проведения некоторых измерений яркости должен принимать значение  $V$ . Эксперт хочет преодолеть эти трудности и затем выделить белым все точки изображения, яркость которых находится в диапазоне от  $I_1$  до  $I_2$ , сохранив яркость всех остальных точек без изменения. Предложите последовательность шагов обработки, придерживаясь которой эксперт достигнет поставленных целей.

11. Белые полосы на представленном тестовом изображении имеют ширину 7 пикселей и высоту 210 пикселей. Расстояние между полосами составляет 17 пикселей. Как будет выглядеть это изображение после применения

(а) Среднеарифметического фильтра размерами  $3 \times 3$ ?

(б) Среднеарифметического фильтра размерами  $7 \times 7$ ?

(в) Среднеарифметического фильтра размерами  $9 \times 9$ ?



*Замечание:* Эта и следующие задачи, связанные с фильтрацией этого изображения, могут показаться несколько утомительными. Однако они стоят того, чтобы потратить усилия на их решение, поскольку способствуют выработке настоящего понимания того, как действуют соответствующие фильтры. После того как Вам будет ясно, как именно конкретный фильтр видоизменяет данное изображение, Ваш ответ может представлять собой короткое словесное описание результата. Например, «результатирующее изображение будет состоять из вертикальных полос шириной 3 пикселя и высотой 206 пикселей». Не забудьте описать изменение формы полос, такое как округление углов. Эффекты, возникающие на краях, где маски лишь частично накладываются на изображение, можно не принимать во внимание.

12. Решите задачу 11 для случая среднегеометрического фильтра.
13. Решите задачу 11 для случая среднегармонического фильтра.
14. Решите задачу 11 для случая медианного фильтра.
15. Решите задачу 11 для случая фильтра максимума.
16. Решите задачу 11 для случая фильтра минимума.
17. Исследуйте предельные эффекты многократного применения низкочастотного сглаживающего фильтра размерами  $3 \times 3$  к дискретному изображению. Можете игнорировать влияние границ изображения.

## Раздел 8

2. Построить граф БПФ для  $N = 9$  с прореживанием по времени.
3. Определить вычислительную сложность алгоритма БПФ для  $N = 3^6$  (в числе операций умножения действительных чисел).
4. Показать, что с точки зрения обеспечения минимума вычислительных затрат предпочтителен алгоритм БПФ по основанию 4.
5. Определить максимальный размер  $M$  окна сканирования, при котором предпочтителен прямой алгоритм вычисления свертки, если  $N = 2048$ , а исходные данные  $X = [x_0; x_1; x_2; x_3; \dots; x_{N-2}; x_{N-1}]$  и ядро свертки  $G = [g_0, g_1, \dots, g_{M-1}]$  комплексные.
6. Существуют ли быстрые алгоритмы БПА по основанию, отличному от двух или четырех?
7. Провести сравнительную оценку сложности алгоритма БПФ, БПХ и БПА для  $N=1024$  и  $N = 4048$

## Раздел 9

1. Для некоторого  $j$ -го положения окна сканирования нелинейным фильтром обрабатывается фрагмент вектора исходных данных

$$X(j) = [0, 1, 12, 1, 15, 13, 10, 3, 2, 2, 0, 15, 13].$$

Обработка выполняется методом ранговой фильтрации. Используя гистограммный алгоритм, найти элементы следующих рангов:

а)  $R = 9$ ;

б)  $R = 5$ .

2. Предложите набор функций градационных преобразований, позволяющих получить все отдельные битовые плоскости 8-битового одноцветного изображения. (Например, функция преобразования с параметрами  $T(r) = 0$  для  $r$  в диапазоне  $[0, 127]$  и  $T(r) = 255$  для  $r$  в диапазоне  $[128, 256]$  дает изображение 7-й битовой плоскости).

3. (а) К какому эффекту на гистограмме приведет обнуление младшей битовой плоскости изображения?

(б) Как будет выглядеть гистограмма, если, наоборот, обнулить старшую битовую плоскость изображения?

4. Объясните, почему операция эквализации гистограммы в дискретном виде не приводит, вообще говоря, к равномерной гистограмме.

5. Предположим, что дискретное изображение было подвергнуто операции эквализации гистограммы. Покажите, что второй проход операции эквализации гистограммы даст в точности тот же результат, что уже был получен после первого прохода.

6. Рассмотрите два 8-битовых изображения (а) и (б), уровни яркостей которых занимают весь диапазон значений от 0 до 255.

1) Обдумайте влияние эффекта ограничения при многократном вычитании изображения (б) из изображения (а).

2) Изменится ли результат, если поменять изображения местами?



## ЛИТЕРАТУРА

1. Блаттер К. Вейвлет-анализ. Основы теории.// – М. – Техносфера. - 2006, - 279 с.
2. Бородакий Ю.В. Информационные технологии. Методы, процессы, системы. М., Радио и связь, 2004.
3. Брейсуэлл Р. Преобразование Хартли. Теория и приложения.// -М.- Мир. -1990. -217 с.
4. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: ДИАЛОГ-МИФИ, 2003. - 384 с.
5. Гонсалес Р., Вудс Р. Цифровая обработка изображений.// Пер. с англ.- Москва.- Техносфера. – 2006. -1072 с.
6. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB: пер. с англ. – М.: Техносфера, 2006. – 616 с.
7. Дагман Э.Е., Кухарев Г.А. Быстрые дискретные ортогональные преобразования.// -Новосибирск. – Наука. -1983. -232 с.
8. Губин А.В., Шарыпин Е.В. Архитектура персональных систем обработки изображений.// Системы и средства информатики. -Ежегодник.- Вып.1. -1989. -с.74-81.
9. Дымков В.И., Сеницын Е.В. Элементы концепции персональных систем обработки изображений.//В сб. Системы и средства информатики. -Вып.1. -1989.-с.74-81.
10. Ковалгин Ю.А., Вологодин Э.И. Цифровое кодирование звуковых сигналов. СПб.: КОРОНА-принт, 2004. – 240 с
11. Куприянов М.С., Матюшкин Б.Д. Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования. СПб., Политехника, 1999.
12. Кухарев Г.А., Тропченко А.Ю., Шмерко В.П. Системные процессоры для обработки сигналов.// -Минск. – Беларусь. -1988. -127 с.
13. Маккелан Д.Х., Рейдер Ч.М. Применение теории чисел в цифровой обработке сигналов.// - М. - Радио и связь. -1983. – 265 с.
14. Под ред. Соифера В.А. Методы компьютерной обработки изображений. – М., Физматлит, 2001.
15. Применение цифровой обработки сигналов.// - Под.ред. А.Оппенгейма.- М. –Мир. -1980.-550 с.
16. Прэт У. Цифровая обработка изображений.// -М. –Мир.-1982. -т.1. - 495 с. -т.2. - 480с.
17. Рабинер Л., Гоулд Б. Теория и применения цифровой обработки сигналов.// -М. – Мир. -1978. - 545 с.
18. Ричардсон Я. Видеокодирование. –М., Техносфера, 2005.
19. Сергиенко А.Б. Цифровая обработка сигналов. - СПб, Питер, 2007.
20. Солонина А., Улахович Д., Яковлев Л. Алгоритмы и процессоры цифровой обработки сигналов. – СПб., «БЧВ-Петербург», 2002.

21. Тропченко А.Ю., Тропченко А.А. Цифровая обработка сигналов. Методы предварительной обработки. Учебное пособие по дисциплине "Теоретическая информатика" // – СПб: СПбГУ ИТМО, 2009. – 88 с.
22. Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудио-сигналов и видео. Учебное пособие по дисциплине "Теоретическая информатика" // СПб: СПбГУ ИТМО, 2009. – 108 с
23. Уидроу Б., Стирнз С. Адаптивная обработка сигналов // -М. - Радио и связь. - 1989, - 440 с.
24. Фрумкин М.А. Систематические вычисления. - М.: Наука, -1990. - 191 с.
25. Ярославский Л.П. Цифровая обработка сигналов в оптике и голографии: Введение в цифровую оптику // - М. - Радио и связь. -1987.- 296 с.
26. Fourati W., Bouhlef M. A Novel Approach to Improve the Performance of JPEG2000 // ICGST International Journal on Graphics, Vision and Image Processing. – 2005. – Vol. 5, N. 5. – P. 1 – 9.

СПбНИУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы и успешно реализовал инновационную образовательную программу «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий», что позволило выйти на качественно новый уровень подготовки выпускников и удовлетворять возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях науки. Реализация этой программы создала основу формирования программы дальнейшего развития вуза до 2015 года, включая внедрение современной модели образования.

---

## КАФЕДРА РЕЧЕВЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

### О кафедре

Кафедра речевых информационных систем (РИС) создана в 2011 году на факультете Информационных технологий и программирования (ФИТиП).

Организатором создания кафедры выступает «Центр речевых технологий» ([www.speechpro.ru](http://www.speechpro.ru)). Заведующий – генеральный директор ООО «Центр речевых технологий», кандидат технических наук Хитров Михаил Васильевич, вице-президент консорциума «Российские речевые технологии», член ISCA, IEEE.

Кафедра РИС обеспечивает подготовку докторантов, аспирантов и магистров. Для тех, кто имеет высшее образование, но хотел бы связать свое будущее с речевыми технологиями, имеются курсы дополнительного профессионального образования.

### Обучение на кафедре

Кафедра «Речевые информационные системы» (базовая кафедра «Центра речевых технологий») Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (ИТМО) в рамках направления 230400.68 «Информационные системы и технологии» открывает прием в магистратуру по новой образовательной программе 230400.68.04 «Речевые информационные системы».

Срок обучения 2 года. Обучение завершается защитой магистерской диссертации.

Целевая установка магистратуры – подготовка специалистов, способных участвовать в исследовательской и проектной работе в области речевых информационных технологий со специализацией в направлениях распознавания и синтеза речи, распознавания личностей по голосу, мультимодальной биометрии, в области проектирования и разработки информационных систем и программного обеспечения.

Область профессиональной деятельности выпускников кафедры РИС включает:

- исследование, разработка, внедрение речевых информационных технологий и систем;
- методы и алгоритмы цифровой обработки речевых сигналов;
- автоматизированные системы обработки речевых сигналов;
- программное обеспечение автоматизированных речевых информационных систем;
- системы автоматизированного проектирования программных и аппаратных средств для речевых информационных систем и информационной поддержки таких средств.

Объектами профессиональной деятельности выпускников кафедры РИС являются:

- информационные процессы, технологии, системы и сети, предназначенные для обработки, распознавания, синтеза речевых сигналов;
- инструментальное (математическое, информационное, техническое, лингвистическое, программное, эргономическое, организационное и правовое) обеспечение речевых информационных систем;
- способы и методы проектирования, отладки, производства и эксплуатации информационных технологий и систем в областях обработки, распознавания, синтеза речевых сигналов, телекоммуникации, связи, инфокоммуникации, медицины.

Широкий профиль подготовки, знание универсальных методов исследования и проектирования информационных систем, практические навыки работы с современным программным обеспечением – все это позволяет выпускникам кафедры найти работу в научных институтах и университетах, в фирмах, на производственных предприятиях, а также в коммерческих структурах. Студентам, которые хорошо проявляют себя в учебе, предлагается работа в ООО «Центр речевых технологий».

Учебный план предусматривает, в частности, следующие курсы:

- Информационные технологии;
- Системный анализ и моделирование информационных процессов и систем;
- Проектирование информационных систем;
- Организация проектирования и разработки программного обеспечения распределенных систем;
- Организация проектирования и разработки программного обеспечения встроенных систем;
- Тестирования программного обеспечения;
- Управление качеством разработки программного обеспечения.

Речевые технологии:

- Цифровая обработка сигналов;
- Цифровая обработка речевых сигналов;
- Математическое моделирование и теория принятия решений;

- Распознавание образов;
- Распознавание и синтез речи;
- Распознавание диктора (говорящего по голосу);
- Мультимодальные биометрические системы.

К преподаванию привлекаются ведущие специалисты «Центра речевых технологий», преподаватели НИУ ИТМО, а также специалисты, работающие в известных научных, производственных и коммерческих организациях.

**Редакционно-издательский отдел**  
Санкт-Петербургского национального  
исследовательского университета  
информационных технологий, механики и оптики  
197101, Санкт-Петербург, Кронверкский пр., 49

