

Министерство образования Республики Беларусь

Учреждение образования  
«Гомельский государственный технический  
университет имени П. О. Сухого»

Кафедра «Промышленная электроника»

**А. В. Ковалев, Д. А. Литвинов**

# **ОБРАБОТКА АУДИО- И ВИДЕОСИГНАЛОВ В СИСТЕМАХ ТЕЛЕКОММУНИКАЦИЙ**

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ  
по курсу «Системы телекоммуникаций»  
для студентов специальности 1-36 04 02  
«Промышленная электроника»  
дневной формы обучения  
В четырех частях  
Часть 1**

Гомель 2011

УДК 621.391(075.8)  
ББК 32.88я73  
К56

*Рекомендовано научно-методическим советом  
факультета автоматизированных и информационных систем  
ГГТУ им. П. О. Сухого  
(протокол № 8 от 28.02.2011 г.)*

Рецензент: доц. каф. «Информационные технологии» ГГТУ им. П. О. Сухого  
д-р техн. наук *И. А. Мурашко*

**Ковалев, А. В.**  
К56      Обработка аудио- и видеосигналов в системах телекоммуникаций : лаборатор. практикум по курсу «Системы телекоммуникаций» для студентов специальности 1-36 04 02 «Промышленная электроника» днев. формы обучения : в 4 ч. Ч. 1 / А. В. Ковалев, Д. А. Литвинов. – Гомель : ГГТУ им. П. О. Сухого, 2011. – 155 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <http://lib.gstu.local>. – Загл. с титул. экрана.

Рассмотрены вопросы обработки аудио- и видеоинформации в системах телекоммуникаций. Дано краткое описание методов и способов обработки и преобразования аудио- и видеоинформации в системах телекоммуникаций.

Для студентов специальности 1-36 04 02 «Промышленная электроника» дневной формы обучения.

УДК 621.391(075.8)  
ББК 32.88я73

© Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», 2011

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. АУДИОИНФОРМАЦИЯ В СИСТЕМАХ ТЕЛЕКОММУНИКАЦИЙ	6
2. МЕТОДЫ СЖАТИЯ РЕЧЕВЫХ СИГНАЛОВ	25
3. ОБРАБОТКА ВИДЕОИНФОРМАЦИИ В СИСТЕМАХ ТЕЛЕКОММУНИКАЦИЙ	64
4. ЛАБОРАТОРНАЯ РАБОТА №1	113
5. ЛАБОРАТОРНАЯ РАБОТА №2	127
6. ЛАБОРАТОРНАЯ РАБОТА №3	146
Список литературы	154

## ВВЕДЕНИЕ

**Телекоммуникация** – связь на расстоянии (Лат.)

**Коммуникация** (процесс обмена информацией) – является необходимым условием существования живых организмов, экологических сообществ и человеческого общества. Общественное развитие сопровождается развитием телекоммуникационных технологий. Особенно интенсивно телекоммуникационные технологии развиваются несколько последних десятилетий. Самым богатым человеком в мире уже является не производитель ПК (Билл Гейц), а владелец системы мобильной связи из Мексики.

**Телекоммуникационная система** – совокупность технических объектов, организационных мер и субъектов, реализующих процессы состоящих из: процессов соединения, процессов передачи и процессов доступа.

Для обмена информацией телекоммуникационные системы используют естественную или искусственную среду. Телекоммуникационные системы вместе со средой, которая используется для передачи информации образуют телекоммуникационные сети. Наиболее важными телекоммуникационными сетями являются:

- Почтовая связь;
- Телефонная сеть общего пользования (ТФОП);
- Мобильные телефонные сети;
- Телеграфная сеть;
- Интернет – глобальная сеть взаимодействия компьютерных сетей;
- Сеть проводного радиовещания;
- Сети кабельного телевидения;
- Сети телевизионного и радио вещания;
- Ведомственные сети связи, которые предоставляют услуги связи органам государственной службы, системы управления воздушным и морским движением, крупным производственным комплексам;
- Глобальные сети спасения и безопасности (Инмарсат, ГМССБ).

Перечисленные выше телекоммуникационные системы, как правило, тесно взаимодействуют друг с другом и используют общие ресурсы для реализации связи. Для организации такого взаимодейст-

вия в каждом государстве и в глобальном масштабе действуют специальные органы, которые

- Регулируют порядок использования общих ресурсов;
- Определяют общие правила взаимодействия (протоколы) телекоммуникационных систем;
- Разрабатывают перспективные телекоммуникационные технологии.

Предметом изучения раздела «Обработка аудио- и видеосигналов в системах телекоммуникаций» дисциплины "Системы телекоммуникаций" является ознакомление студентов с современными методами и алгоритмами обработки речевых, аудиосигналов и сигналов изображения для повышения качественных показателей современных систем связи.

Важнейшим видом услуг систем телекоммуникаций является обеспечение эффективной передачи аудио- и видеосигналов. В связи с этим повышенные требования предъявляются к быстродействию и качества и обработки речевых сигналов и изображений с целью улучшения характеристик современных систем связи.

После изучения данного раздела дисциплины студент должен знать принципы разработки, эксплуатации и обслуживания средств обработки аудио- и видеосигналов.

Под *сигналом* мы понимаем любую переменную, которая передает или содержит некий вид информации, и которую можно, например, переносить, выводить на экран или выполнять с ней какие-то действия. Особый интерес в системах телекоммуникаций представляют такие типы сигналов:

- речь - при разговоре по телефону, прослушивании радио, при беседе или слушании информационных сообщений в помещениях и на открытом пространстве;
- звуки и музыка – при записи и прослушивании музыкальных компакт-дисков, работе с оцифрованным и сжатым звуком (MP3-плееры, мультимедийные презентации и т.п.), при редактировании звука в студиях звукозаписи, радио и телестудиях (микширование и т.п.);
- биомедицинские сигналы - на выходе ультразвуковых доплеровских измерительных систем, систем диагностики заболеваний органов дыхания, слуха, речи;
- сигналы на выходе систем неразрушающего контроля;
- сигналы на выходе акустических измерительных систем;

- изображения – цифровая фотография, видео- и телеизображения;
- сигналы радаров – при обнаружении и распознавании отдаленных целей.

Задачи, решаемые при обработке сигналов в системах телекоммуникаций:

- идентификация и распознавание (идентификация диктора, конвертирование речи в текст, синтез речи по тексту...);
- обработка музыкальных и речевых сигналов (цифровые аудиосистемы, спектральный анализ, микширование, выравнивание, фильтрация, подавление шумовой и реверберационной помех...);
- измерение параметров (скорость и объем) кровотока, положения, размеров и плотности внутренних органов;
- изучение акустических свойств помещений, конструкций, материалов;
- изучение физических свойств материалов акустическими методами (измерение плотности, вязкости, ...);
- диагностика и контроль (дефекты в турбинах авиационных двигателей, шумы в легких, ...);
- определение геометрических свойств объектов (толщина стенок труб, уровень жидкости, ..);
- улучшение изображений, распознавание изображений, факсимиле, спутниковые карты погоды, анимация;
- визуализация акустических сигналов (УЗИ, локаторы бокового обзора,...).

## **1. АУДИОИНФОРМАЦИЯ В СИСТЕМАХ ТЕЛЕКОММУНИКАЦИЙ**

### **1.1. Речевой сигнал в системах связи и в каналах утечки информации**

Тракт передачи речи (ТПР), обобщенная схема которого приведена на рис.1.1, состоит из микрофона М, передающей аппаратуры (ПА), канала связи (КС), приемной аппаратуры (ПрА), телефона (Т) [1].

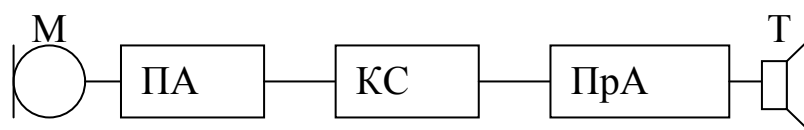


Рисунок 1.1 - Тракт передачи речи

Сопоставляя схему ТПР со схемой технического канала утечки информации (ТКУИ) [2,3], содержащей блоки «объект» (источник речевого сигнала), «среда» (воздух, элементы конструкции здания и т.п.) и «технические средства разведки» (ТСР – микрофоны, диктофоны и т.п.), приведенной на рис.1.2, нетрудно заметить очевидное их сходство:

- блоку ПА на рис.1.1 соответствует блок «объект» на рис.1.2;
- блоку «канал связи» на рис.1.1 соответствует блок «среда» на рис.1.2;
- блоку ПрА на рис.1.1. соответствует блок ТСР на рис.1.2.



Рисунок 1.2 - Технический канал утечки информации

Поэтому вполне понятно принципиальное сходство критериев качества, используемых при анализе обеих задач. Так, согласно [1], основными критериями качества тракта передачи речи служат:

- разборчивость;
- громкость;
- натуральность.

При этом громкость не является самостоятельным параметром – она используется совместно с разборчивостью и определяет желаемый (комфортный) уровень принимаемых сигналов. Даже из личной практики каждого человека известно, что чересчур низкий уровень громкости приводит к снижению разборчивости. Исследования показывают, что разборчивость понижается и при чересчур высоком уровне громкости речи.

Натуральность речи – это способность системы воспроизводить не только смысл передаваемой речи, но и ее тембр, индивидуальные особенности речи диктора. Для трактов передачи речи (телефонная

связь, радиосвязь) эта характеристика является второстепенной, за исключением тех случаев, когда стоит задача высококачественного воспроизведения речи диктора (или пения). В ТКУИ натуральность речи также является второстепенной, если только не стоит задача определения личности диктора. Таким образом, разборчивость речи является основным параметром, характеризующим как ТПР, так и ТКУИ.

## 1.2. Методы измерения разборчивости речи

Все методы измерения разборчивости речи условно делятся на *субъективные* и *объективные* [1]. «Условно» - поскольку существует по меньшей мере две трактовки «субъективности-объективности» метода.

**Первая трактовка.** Согласно первой трактовке, при субъективном методе разборчивость речи оценивают по результатам единственного опыта – отсюда неизбежное влияние на результаты измерений особенностей речи и слуха людей, участвующих в испытаниях. Чтобы *объективизировать* метод, нужно осуществить много экспериментов с различными дикторами и абонентами, а затем усреднить результаты измерений.

Разборчивость передаваемой речи оценивают по пятибалльной шкале Международного Консультативного Комитета по Радиосвязи (МККР) [1]:

- неразборчиво;
- разборчиво временами;
- разборчиво с трудом;
- разборчиво;
- совсем разборчиво.

Примером объективизации измерений разборчивости речи служат современные методы оценки качества передачи речи по трактам связи, изложенные в Государственном стандарте Российской Федерации [4], согласно которым в испытаниях должно участвовать не менее 3-х дикторов и 4-5-ти аудиторов, удовлетворяющих ряду условий (отсутствие выраженных дефектов речи и слуха, нетренированность). Например, при оценивании фразовой разборчивости диктор читает одну таблицу фраз в нормальном темпе произнесения (одна фраза за 2,4 с)



и вторую таблицу в ускоренном темпе (одна фраза за 1,5-1,6 с). Пауза между фразами должна быть 5-6с. Аудитор прослушивает сначала таблицу, прочитанную диктором в нормальном темпе, затем таблицу, прочитанную тем же диктором в ускоренном темпе. Правильность приема фраз определяют по квитанции, переданной по каналу телефонной связи. Цикл измерений состоит из передачи всеми дикторами по 10 таблиц каждому, и приема всеми аудиторами всех переданных таблиц. Фразу считают неправильно принятой, если хотя бы одно слово принято неправильно, пропущено или добавлено. Фразовую разборчивость определяют путем вычисления процента правильно принятых фраз для нормального и ускоренного темпов произнесения по формуле

$$J_{H(V)} = \frac{1}{N} \sum_{i=1}^N j_i$$

где  $j_i$  - результат единичного измерения фразовой разборчивости, %;  $N$  - число единичных измерений;  $J_H$  - фразовая разборчивость при нормальном темпе произнесения, %;  $J_V$  - фразовая разборчивость при ускоренном темпе произнесения, %.

Методы, подобные описанному выше, называют *артикуляционными*. При их использовании необходимо располагать специальными артикуляционными таблицами, составление которых – самостоятельная сложная задача. Другой недостаток артикуляционных методов – большое время испытаний, составляющее несколько недель.

Очевидное достоинство артикуляционных методов – простота, позволяющая участвовать в испытаниях операторам с относительно низким уровнем технической квалификации.

**Вторая трактовка.** При второй трактовке «субъективности-объективности» метода субъективными называют все методы, в которых человек является составной частью измерительного тракта, а объективными – такие методы, в которых весь измерительный процесс осуществляется приборами без участия органов чувств человека. С этой целью передающий и принимающий операторы должны быть заменены искусственными эквивалентами («искусственный голос», «искусственное ухо»).

**Тональный метод.** В [1] описывается два метода измерения разборчивости речи, базирующихся на такой трактовке объективности. В

так называемом «*тональном методе*» несколько дикторов заменены единственным искусственным голосом, который генерирует чистые тоны. Искусственный голос представляет собой обычный громкоговоритель без диффузора, возбуждаемый с помощью тонального генератора таким образом, чтобы *уровень звуковых давлений, создаваемых на различных частотах, соответствовал бы кривой спектра формант*. Помещение передачи и помещение приема речевых сигналов искусственно зашумляют – тем самым обеспечивают требуемое отношение сигнал-шум, при котором испытываемая система должна нормально функционировать. Прием информации по-прежнему производит бригада audитором. При этом *задача аудитором упрощается*: вместо того, чтобы осмыслить и записать услышанное звуко сочетание, от них требуется лишь определить, слышен ли сигнал на данной частоте или не слышен. Кроме того, операторы должны измерить уровень ощущения формант – делается это весьма просто, путем введения положительного или отрицательного затухания в тракт связи. Если сигнал слышен, затухание делают положительным, пока сигнал перестанет быть слышимым. Наоборот, если сигнал не слышен, затухание делают отрицательным, пока сигнал не станет слышимым. Дальнейшее определение величины разборчивости речи производят аналитически, с помощью графиков и несложных формул.

Таки образом, тональный метод, в отличие от артикуляционного, можно отнести к *косвенным методам* измерений разборчивости.

Достоинства тонального метода:

- 1) не применяются артикуляционные таблицы;
- 2) значительное сокращение времени измерений.

Недостатки тонального метода:

- 1) повышенные требования к технической грамотности персонала, организующего испытание;
- 2) человек еще не выведен из состава измерительной системы.

Как указано в [1], тональный метод регламентирован ГОСТ № 8031-56.

**Объективный метод.** В другом методе, именуемом «*объективным*», применяют как искусственный голос, так и искусственное ухо. В данном методе, как видим, *человек полностью выведен из состава измерительной системы*. Следует отметить, что искусственное ухо –

обычный в технике акустических измерений прибор, применяемый при испытании телефонов и позволяющий воспроизвести акустическую нагрузку, создаваемую на телефон естественным ухом. Тем самым удастся измерить звуковое давление, создаваемое звучащим телефоном в ухе.

Общий порядок измерений при этом такой:

1) С помощью генератора шума и громкоговорителя создают уровень шума, соответствующий условиям работы приемного конца испытываемого тракта. Измеряют уровень шума на выходе искусственного уха в критической полосе частот слуха, причем средняя частота этой полосы равна частоте измерительного тона.

2) Генератор шума выключают, а вместо него на вход тракта «искусственный голос - канал передачи - искусственное ухо» подают тональный сигнал. Уровень интенсивности звука на микрофоне берется таким, чтобы при условном нуле на регулировщике затуханий распределение звуковых давлений соответствовало кривой спектра формант.

3) С помощью регулирования затуханий добиваются, чтобы уровень сигнала на выходе искусственного уха был таким же, как уровень шума. Показания регулятора затуханий представляют собой *результат измерений уровня ощущений*.

4) Далее, как и в тональном методе, определение величины разборчивости речи производят аналитически, с помощью графиков и несложных формул (тех же).

Объективный метод точнее и быстрее тонального, для его проведения не нужны операторы (дикторы и аудиторы). Наконец, объективный метод принципиально позволяет полностью автоматизировать процедуру измерений на базе современных ЭВМ.

Как и тональный метод, объективный метод является косвенным, т.е. разборчивость речи оценивается не путем подсчета правильно распознанных речевых единиц, а путем проведения специального измерительного эксперимента со звуковыми сигналами в виде тона и полосового шума, в ходе которого измеряются уровни ощущений в нескольких полосах частот. Разборчивость речи вычисляют, базируясь на результатах экспериментальных измерений.

Ввиду неоспоримых достоинств объективного метода, а также в силу отмеченного выше принципиального сходства ТПР и ТКУИ, в настоящее время объективный метод в той или иной модификации применяют как при оценке качества каналов связи, так и при оценке эффективности защиты речевой информации [5,6,7]. Поэтому целесообразно подробно рассмотреть идею, лежащую в основе рассмотренных выше косвенных методов измерения разборчивости речи. При этом целесообразно также вспомнить некоторые положения психофизиологии речи и слуха, на использовании которых базируются косвенные методы.

### 1.3. Основные свойства речевых сигналов

**Спектральные свойства звуков речи.** Спектры *гласных звуков* представляют собой (в первом приближении) периодическую последовательность спектральных пиков. Период следования этих пиков называют частотой основного тона. Выраженные всплески уровня «оггибающей» спектральных пиков именуют «формантами» (рис.1.3).

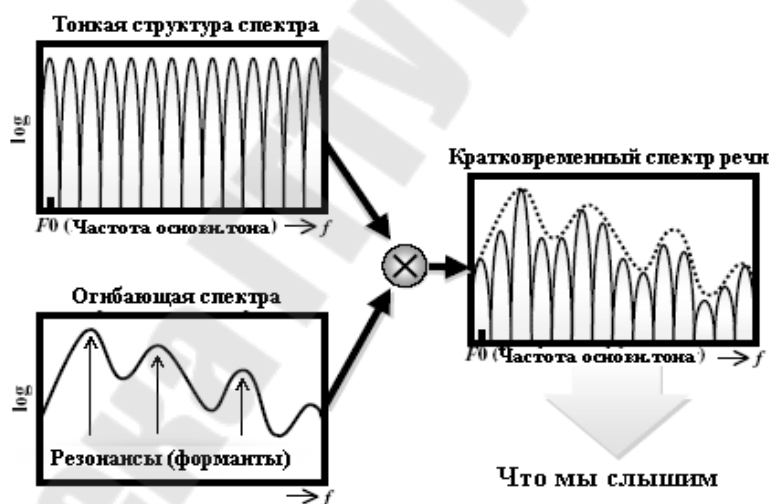


Рисунок 1.3 - Вид спектра гласного звука

Полезная информация о гласном звуке речи содержится в описании соответствующих формант. Принято каждую форманту описывать ее граничными частотами. В русском языке достаточно ограничиться одной-двумя формантами, чтобы достигнуть приемлемой разборчивости речи.

Спектры согласных звуков либо полностью сплошные, т.е. совсем не содержат дискретных компонентов, либо сплошные в отдельных

полосах частот. Эти спектры также содержат локальные всплески. Некоторые из них являются формантами, некоторые - нет.

Чтобы решить, какие всплески уровня спектра являются формантами, следует помнить, что физическая природа формант – явление резонанса в полостях глотки и носоглотки (рис.1.4).

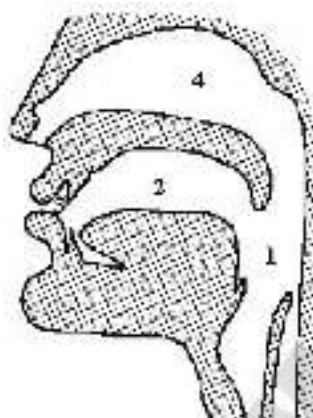


Рисунок 1.4 - Полости глотки (1) и носоглотки (2-4)

В отдельных звуках можно заметить до 6 спектральных подъемов. К формантам относятся только те, которые обусловлены явлением резонанса в речевом аппарате человека. Часть формант (как указывалось – одна-две в русском языке) обеспечивают разборчивость речи, другая часть обеспечивает индивидуальность голоса диктора, что может быть использовано в задачах распознавания голоса (идентификации) диктора.

Форманты звуков речи расположены в области частот от 200 до 8600 Гц. Однако подавляющая часть формант звуков речи находится в пределах от 300 до 3000 Гц, поэтому такую полосу обычно считают достаточной для хорошей разборчивости речи.

Спектральные различия между звуками речи являются главными, хотя и не единственными. Например, при распознавании согласных звуков важны и временные характеристики.

**Интегральные спектральные характеристики речи.** Как следует из названия, интегральные спектральные характеристики речи характеризуют свойства речевого процесса в целом [1]. Сюда относят:

- спектр речи  $B_p(f)$ ;
- спектр формант  $B'_p(f)$ ;
- относительную встречаемость формант по спектру  $\Delta A\%(f)$ .

*Спектром речи* называют оценку спектральной плотности мощности речевого сигнала  $B_p(f)$ , вычисленную по отрезку речевого сигнала значительной протяженности (более минуты). Спектр речи характеризует распределение мощности речевого сигнала по частоте.

*Спектром формант*  $B'_p(f)$  называют зависимость наиболее вероятного уровня формант от частоты. Чтобы оценить спектр формант, необходимо также располагать отрезком речи большой протяженности. Спектр формант на всех частотах меньше спектра речи (рис.1.5):

$$B'_p(f) < B_p(f).$$

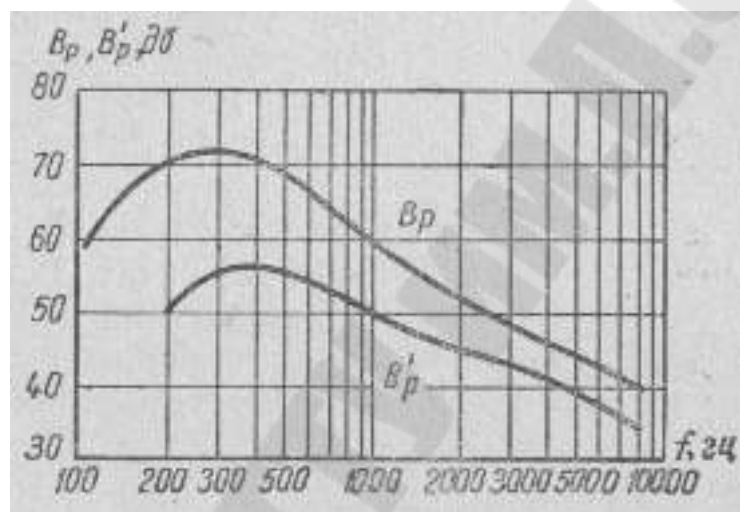


Рисунок 1.5 - Соотношение спектра речи  $B_p(f)$  и спектра формант  $B'_p(f)$

*Относительная встречаемость формант*  $\Delta A\%(f)$  по спектру может быть оценена так. Разобьем весь диапазон частот на полоски, например, по 100 Гц, и подсчитаем относительное число формант (в %) каждой полоске. Результат такого подсчета даст нам кривую  $\Delta A\%(f)$  (рис.1.6).

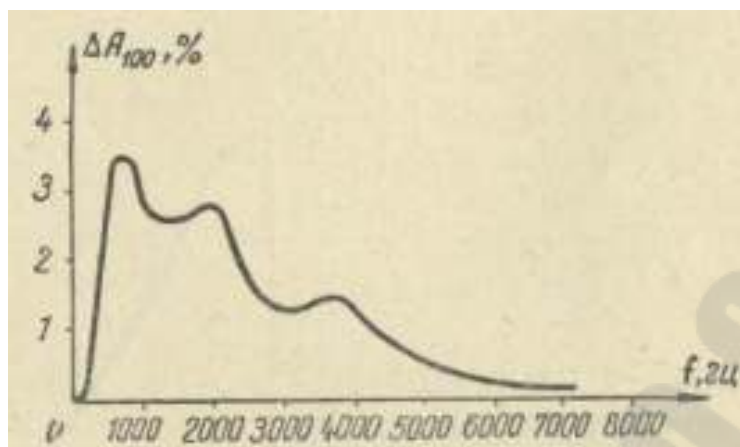


Рисунок 1.6 - Относительная встречаемость формант  $\Delta A\%(f)$

#### 1.4. Основные свойства слуха

Для оценивания разборчивости речи наибольшее значение имеют следующие характеристики слуховой системы человека, именуемые «постоянными слуха» [1]:

- порог слышимости  $\beta_0(f)$ ;
- логарифмическая ширина критической полосы слуха  $k(f)$ ;
- маскировка слуха  $M(f)$ .

Порог слышимости  $\beta_0(f) = 20 \lg \frac{p_{\min}(f)}{p_0}$  – это минимальное звуко-

вое давление, ниже которого ухо не воспринимает звук (рис.1.7). Выражается в децибелах, по отношению к давлению  $p_0 = 2 \cdot 10^{-5} \text{ Pa}$ , соответствующему пороговой величине давления звука на частоте 1000 Гц.

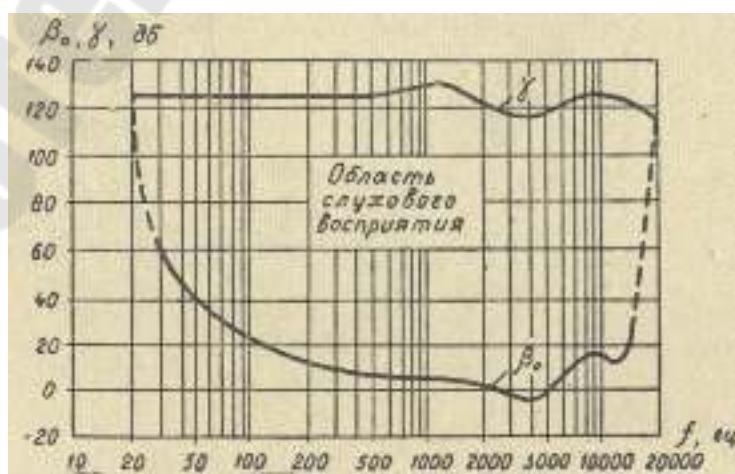


Рисунок 1.7- Порог слышимости  $\beta_0(f)$  и болевой порог  $\gamma(f)$

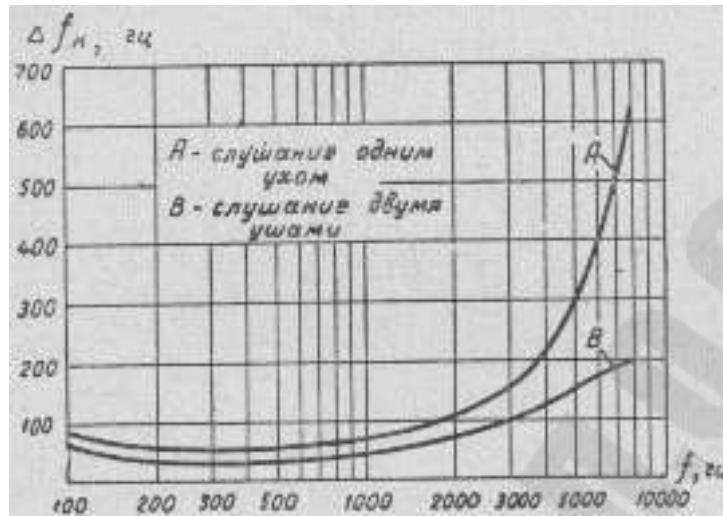


Рисунок 1.8 - Критическая полоса слуха  $\Delta f_k(f)$

Ширина критической полосы слуха  $\Delta f_k(f)$  - это разрешающая способность слухового аппарата человека, который можно уподобить гребенке фильтров. Например, на частоте 100 Гц критическая полоса слуха близка 100 Гц, а на частоте 8000 Гц – близка 600 Гц (рис. 1.8). Для удобства расчетов вводят понятие *логарифмической критической полосы слуха* (рис. 1.9):

$$k(f) = 10 \lg \Delta f_k(f).$$

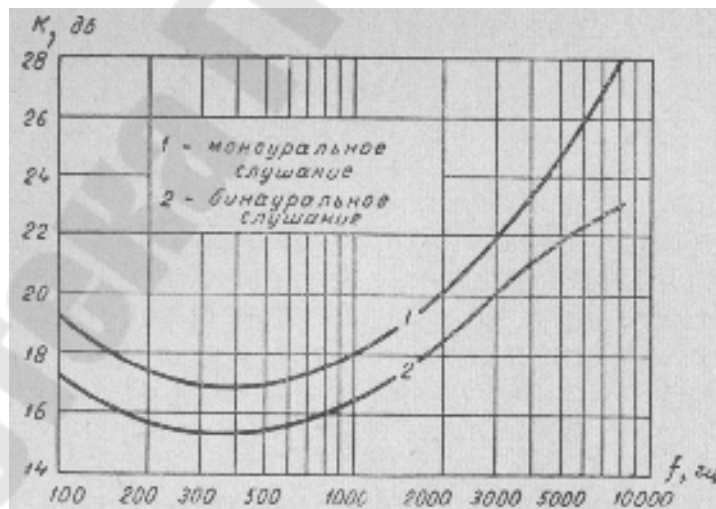


Рисунок 1.9 - Логарифмическая критическая полоса слуха [1]

Маскировка слуха – это явление ослабления слышимости или полного пропадания полезного звука на фоне мешающего звука. Количественно выражается как разница:

$$M(f) = \beta_m(f) - \beta_0(f),$$



где  $\beta_m(f)$  - порог слышимости при наличии мешающего звука. На рис.10 приведено семейство индивидуальных кривых маскировки для различных уровней маскирующего сигнала  $N$ . Здесь  $\Delta z = z(i) - z(j)$  - разность высот тона маскирующей  $z(i)$  и маскируемой  $z(j)$  компонент, причем высота тона  $z$  измеряется в Барках:

$$z = 13 \cdot \arctg(0.0076 \cdot f) + 3.5 \cdot \arctg[(f / 7500)]^2 .$$

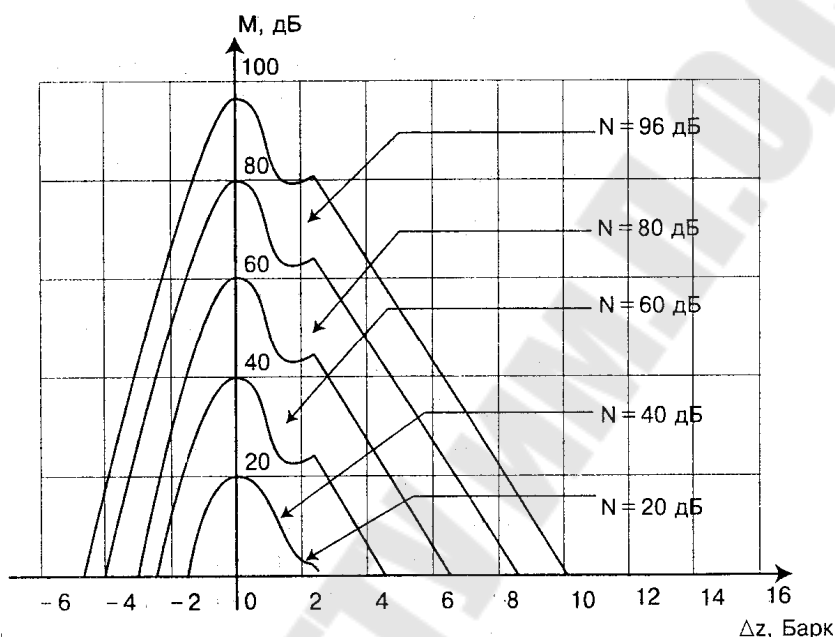


Рисунок 1.10 - Семейство индивидуальных кривых маскировки [8]

### 1.5. Измерение разборчивости

Различают [1] следующие виды (меры) разборчивости речи:

- разборчивость формант  $A$ ;
- разборчивость звуков  $D$ ;
- разборчивость слогов  $S$ ;
- разборчивость слов  $W$ ;
- разборчивость фраз  $J$ .

При расчете разборчивости приходится иметь дело с частотно-зависимыми функциями. Поэтому результаты количественного расчета для различных участков спектра различны. Ввиду этого задача расчета решается разделением диапазона частот, используемого для передачи речи, на узкие полосы, внутри которых можно не считаться с указанной частотной зависимостью и относить полученные результа-

ты к средней частоте полосы. Далее вычисляется разборчивость для каждой полосы частот, а общая разборчивость находится суммированием «полосовых» разборчивостей.

Поскольку *свойством аддитивности обладает только формантная разборчивость*:

$$A = \sum_{i=1}^N \Delta A_i,$$

где  $\Delta A_i$  - формантная разборчивость в  $i$ -той полосе частот, идея расчета сводится к предварительному вычислению величины  $A$ , с последующим пересчетом ее в величины  $D$ ,  $S$ ,  $W$ ,  $J$ , на основании имеющейся информации о зависимости между разными мерами разборчивости.

Разборчивость в каждой полосе можно представить в виде:

$$\Delta A_i = P_i \cdot \Delta A_{im},$$

где  $\Delta A_{im}$  - формантная разборчивость в отсутствие мешающих факторов (шум, влияние тракта передачи);  $P_i$  - коэффициент восприятия, учитывающий потери разборчивости из-за наличия мешающих факторов.

Разделение диапазона частот речевого сигнала на полосы можно производить по-разному. В [1] называется два способа:

- деление на полосы одинаковой ширины;
- деление на равноартикуляционные полосы,

причем предпочтение отдается второму способу, позволяющему упростить выкладки. Количество полос при этом предлагается выбрать равным  $N = 20$ . В работе [9] также выбран способ деления на равноартикуляционные полосы.

В [9-12] указывается иной способ – деление на октавные или третьоктавные полосы. Количество полос при этом предлагается выбрать равным  $N = 5$ . Зачастую, выбор способа деления на полосы частот – вопрос не столько принципиальный, сколько зависящий от «вкуса» исследователя.

Рассмотрим далее идею расчета разборчивости речи, исходя из принципа деления на равноартикуляционные полосы [1]. При этом

$$\Delta A_{im} = \frac{1}{20},$$

поскольку, в силу вероятностного характера формантной разборчивости, справедливо соотношение

$$\sum_{i=1}^N \Delta A_{im} = 1.$$

Таким образом,

$$A = 0.05 \sum_{i=1}^{20} P_i.$$

Величины  $P_i$  определяют, исходя из эмпирической функциональной зависимости  $P(E')$  (функцию  $P(E'_i)$  называют «постоянной артикуляционной характеристикой речи» [1]), где уровень ощущения формант  $E'$  вычисляют по формуле:

$$E' = B'_p + k - (b + M) - \beta_0,$$

где  $B'_p$  - значение спектра формант на входе тракта;  $\beta_0$  - порог слышимости;  $M$  - маскировка от шумов всех видов;  $b = 20 \lg \frac{P_{ex}}{P_{вых}}$  - затухание в тракте;  $k$  - логарифмическая ширина критической полосы слуха.

Для достаточно высоких уровней шума:

$$Z = B_{ш} + k - \beta_0 > 10 \text{ дБ}$$

выражение для уровень ощущения формант  $E'$  можно вычислять по упрощенной формуле:

$$E' = B'_p - b - B_{ш}.$$

Вид зависимости  $P(E')$  приведен на рис.11.

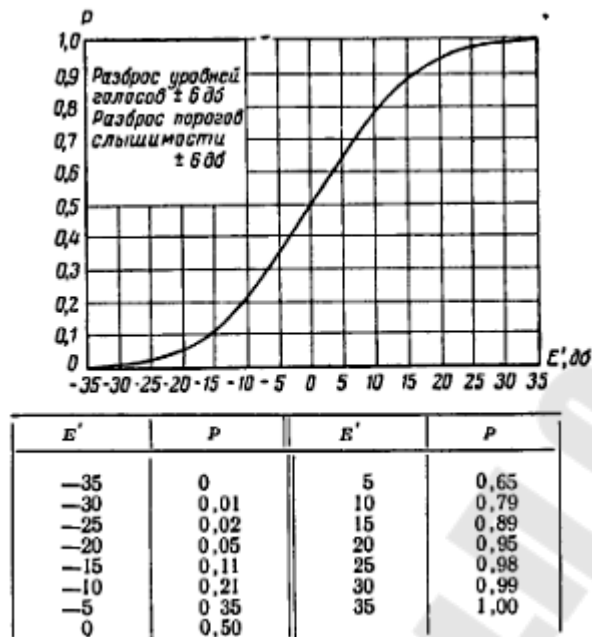


Рисунок 1.11 - Вид зависимости  $P(E')$

Отметим два важных обстоятельства. Во-первых, функция  $P(E')$  не зависит от полосы частот. Во-вторых, в литературе можно встретить весьма различающиеся кривые  $P(E')$ . Например, даже в работе [1] встречаем две такие кривые: для «идеализированной артикуляционной бригады» и для «типовой артикуляционной бригады». А в работе [6] приводится аналогичная зависимость, существенно отличающаяся от соответствующих кривых в работе [1]. Более внимательный анализ работы [6] показывает, что здесь вместо спектра формант  $B'_p$  используют спектр речи  $B_p > B'_p$ , и, как следствие, вместо уровня ощущения формант  $E'$  используют уровень ощущения речи:

$$E = B_p + k - (b + M) - \beta_0.$$

Таким образом, в [6] вместо зависимости  $P(E')$  предлагают использовать сходную, но количественно отличающуюся зависимость  $P(E)$ .

Таким образом, хотя аналитическая методика расчета разборчивости речи на сегодняшний день проработана теоретически и экспериментально весьма глубоко, при практическом ее использовании следует помнить о существовании множества модификаций такой методики. Непродуманное «перекрестное» использование элементов этих методик может привести к неверным результатам расчета разборчивости речи.

## 1.6. Шумовая и реверберационная помехи, защита пассивная и активная

Из приведенных выше соотношений следует, что в системах защиты речевой информации для снижения разборчивости речи необходимо стремиться уменьшить отношение "уровень речевого сигнала/уровень шума" (сигнал/шум) в местах возможного размещения датчиков аппаратуры акустической разведки. Уменьшение отношения сигнал/шум возможно путем или уменьшения (ослабления) уровня речевого сигнала (*пассивные методы защиты*), или увеличения уровня шума (создания акустических и вибрационных помех) (*активные методы защиты*).

## 1.7. Повышение разборчивости речи путем ее обработки

Для последнего времени характерен повышенный интерес к проблеме создания систем автоматического распознавания речи, идентификации человека по голосу. Очевидна тесная связь такой проблемы с проблемой защиты речевой информации. Действительно, как в задачах автоматического распознавания речи, так и в задачах защиты речевой информации, «результатирующий» речевой сигнал  $z(t)$  можно представить в виде:

$$z(t) = s(t) \otimes h(t) + v(t),$$

где  $s(t)$  - исходный речевой сигнал;  $h(t)$  - импульсная переходная характеристика канала передачи;  $v(t)$  - шумовая помеха;  $\otimes$  - символ свертки.

Очевидно, разборчивость речи можно повысить, если поставить задачу такой коррекции (фильтрации) сигнала  $z(t)$ , результат которой  $\hat{s}(t)$  максимально возможно приближался бы к исходному сигналу  $s(t)$ .

Постановка задачи коррекции речевых сигналов изображена на рис. 1.12.

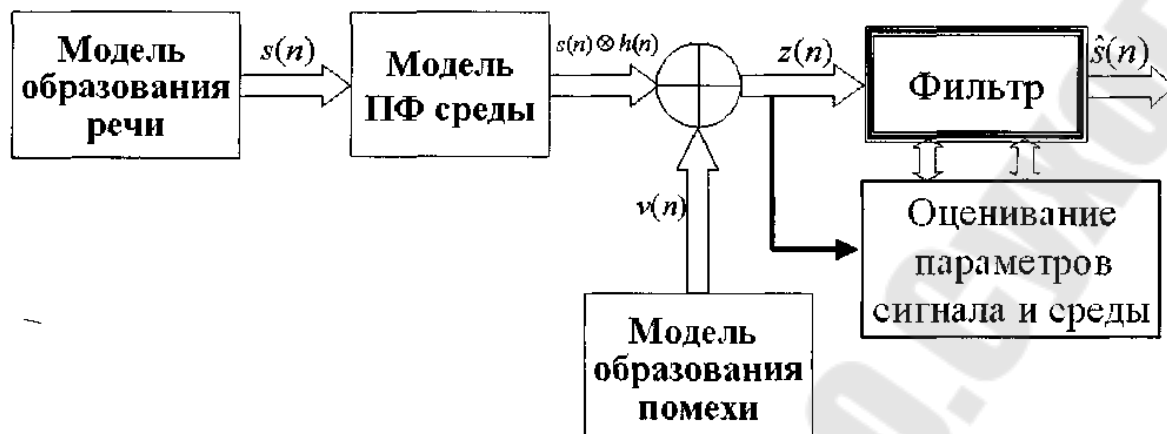


Рис.1.12 – Постановка задачи коррекции речевых сигналов

Существует большое количество подходов к коррекции речевых сигналов, среди которых можно выделить две основные группы:

- 1) общие фильтрационные подходы, не учитывающие специфику речевых сигналов;
- 2) методы, основанные на специальных моделях речеобразования.

Вторая группа методов является более эффективной. Задача восстановления речевого сигнала при этом разбивается на три этапа:

- 1) оценивание параметров авторегрессионной модели речеобразования;
- 2) оценивание параметров передаточной функции среды и фоновых помех;
- 3) применение фильтрационной процедуры, на основе полученных оценок параметров, с целью восстановления искаженного сигнала.

Успешное развитие современных речевых технологий стало возможным во многом благодаря внедрению авторегрессионной (АР) модели речеобразования (рис.1.13).

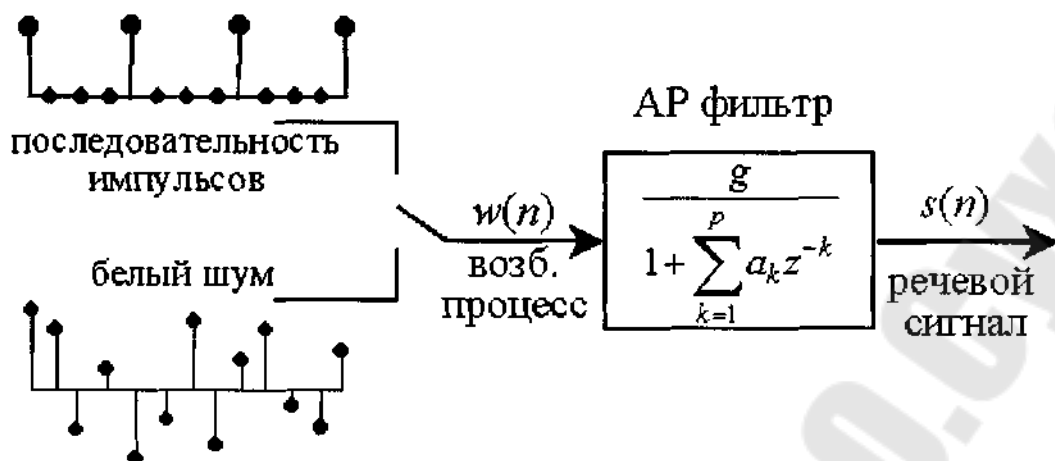


Рисунок 1.13 – Модель речеобразования

Данная модель учитывает основные механизмы генерирования речевых сигналов и в то же время удобна при практическом применении. Однако искажения, вводимые в речевой сигнал вследствие его прохождения через акустическую среду или канал связи, приводят к отклонениям от данной модели, что влечет за собой существенное ухудшение характеристик систем обработки речи. Поэтому возникает необходимость в использовании систем предварительной коррекции речевых сигналов, обеспечивающих устранение или сведение к приемлемому уровню описанных искажений.

В [13] произведен обзор работ, посвященных системам цифровой коррекции речевых сигналов, а также предложены собственные методы такой коррекции. В качестве одного из наиболее эффективных методов фильтрации речевых сигналов называется калмановская фильтрация. Традиционные методы калмановской фильтрации речевых сигналов имеют итерационную структуру, что удобно при проведении обработки в реальном масштабе времени, однако требуют пере-вычисления всех внутренних переменных алгоритма при поступлении каждого нового отсчета речевого сигнала. Как следствие, существующие алгоритмы являются низкоэффективными в вычислительном плане. Вместе с тем, попытки модифицировать алгоритм калмановской фильтрации зачастую приводят к неприемлемому снижению точности вычислений. В [13] называется еще один недостаток существующих методов калмановской фильтрации – неспособность обрабатывать сигналы «блоками» - и предлагается эффективный алгоритм блочной калмановской фильтрации. Этот недостаток является принципиальным в задачах передачи речи по коммуникационным каналам.

Заметим, однако, что он вряд ли важен в задаче защиты речи от утечки в акустическом и вибрационном каналах.

Интерес представляет рассмотренный в [13] метод «слепой деконволюции» речевых сигналов, направленный на устранение влияния передаточной функции среды  $h(t)$ . Спецификой предложенного в [13] подхода является использование единственного канала приема речевой информации, тогда как большинство существующих работ рассматривает ситуации многоканального приема информации.

### 1.9. Каналы утечки речевой информации

Выше приведены два определения *технического канала утечки информации* (ТКУИ): 1) как совокупности объект+среда+ТСР (техническое средство разведки); 2) как способа получения информации с помощью ТСР.

В зависимости от физической природы возникновения информационных сигналов, среды распространения акустических колебаний и способов их перехвата технические каналы утечки акустической (речевой) информации подразделяют на **воздушный, вибрационный, электроакустический, оптико-электронный и параметрический**.

В **воздушных технических каналах утечки информации** средой распространения акустических сигналов является воздух, и для их перехвата используются миниатюрные высокочувствительные микрофоны и специальные направленные микрофоны. Миниатюрные микрофоны объединяются (или соединяются) с портативными звукозаписывающими устройствами (диктофонами) или специальными миниатюрными передатчиками. Автономные устройства, конструктивно объединяющие миниатюрные микрофоны и передатчики, называют закладными устройствами перехвата речевой информации, или просто **акустическими закладками**. Перехваченная закладными устройствами речевая информация может передаваться по радиоканалу, оптическому каналу, телефонной линии, сети переменного тока, вспомогательным техническим средствам и системам (ВТСС) - водопроводным и канализационным трубам и т.п. Использование портативных диктофонов и акустических закладок требует проникновения на контролируемый объект (в помещение). В том случае, когда это не



удается, для перехвата речевой информации используются направленные микрофоны.

**Вибрационный канал** – средой распространения являются конструкции помещения или здания (стены, пол, потолок, трубы и т.п.), в качестве ТСР обычно применяют так называемые «электронные стетоскопы» (контактный микрофон+усилитель). Реализованные в виде комбинации с радиоустройством, такие стетоскопы именуют «радиостетоскопами».

**Электроакустический канал** – любой элемент/устройство, обладающее «микрофонным эффектом», т.е. превращающее механическое колебание в электрический сигнал. Примеры: трансформаторы, дроссели, некоторые датчики пожарной сигнализации, громкоговорители радиотрансляционной сети, телефонные аппараты с электромеханическими вызывными звонками. Особая разновидность электроакустического канала – высокочастотное навязывание (контактное введение токов высокой частоты в ВТСС, с последующим съемом модуляционного сигнала).

**Параметрический канал** – используется изменение параметров электронных схем, обусловленное воздействием акустических волн. В качестве «носителей» таких электронных схем используют, например, гетеродинные приемники, телевизоры и т.п. Иной вариант реализации – высокочастотное облучение полуактивных закладок.

**Оптико-электронный канал** – используются инфракрасные лазеры, совместно с приемной аппаратурой, фиксирующей отражение лазерного луча от оконных стекол, зеркал, стекол в картинах («лазерные микрофоны»).

## 2. МЕТОДЫ СЖАТИЯ РЕЧЕВЫХ СИГНАЛОВ

На приведенных ниже рисунках изображены фрагменты речевых сигналов, содержащих гласные (рис. 2.1) и согласные (рис. 2.2) звуки, а также спектры этих сигналов (рис. 2.3 и 2.4). Хорошо видна разница в характере соответствующих сигналов, а также то, что как в первом, так и во втором случаях ширина спектра сигнала не превышает 3,5 кГц. Кроме этого, можно отметить, что уровень низкочастотных (то есть медленных по времени) составляющих в спектре речевого сигнала значительно выше уровня высокочастотных (быстрых)

составляющих. Эта существенная неравномерность спектра, кстати, является одним из факторов сжимаемости таких сигналов.

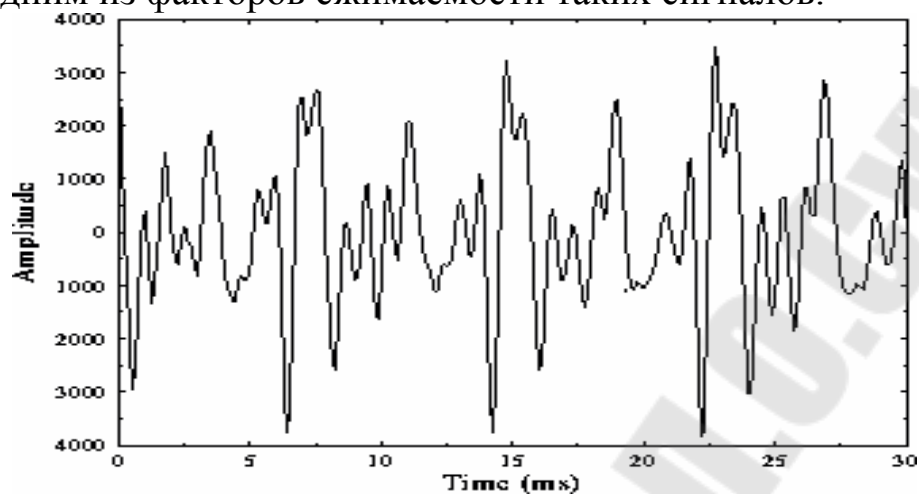


Рисунок 2.1 - Фрагмент речевого сигнала, содержащий гласные звуки

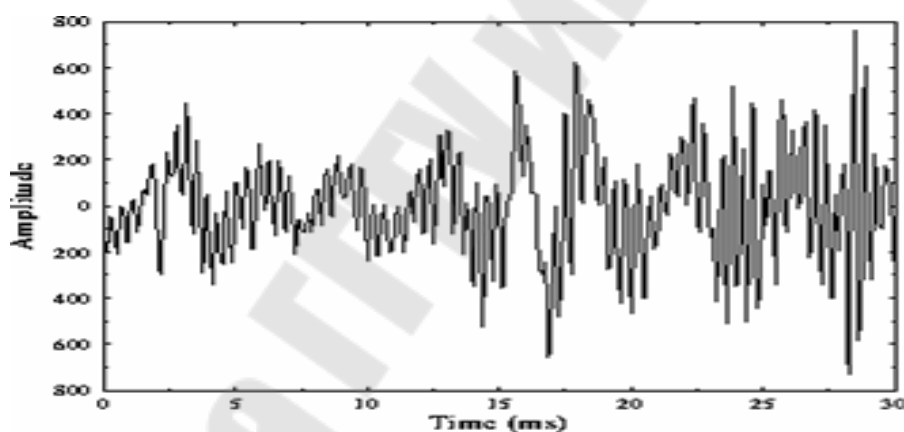


Рисунок 2.2 - Фрагмент речевого сигнала, содержащий согласные звуки

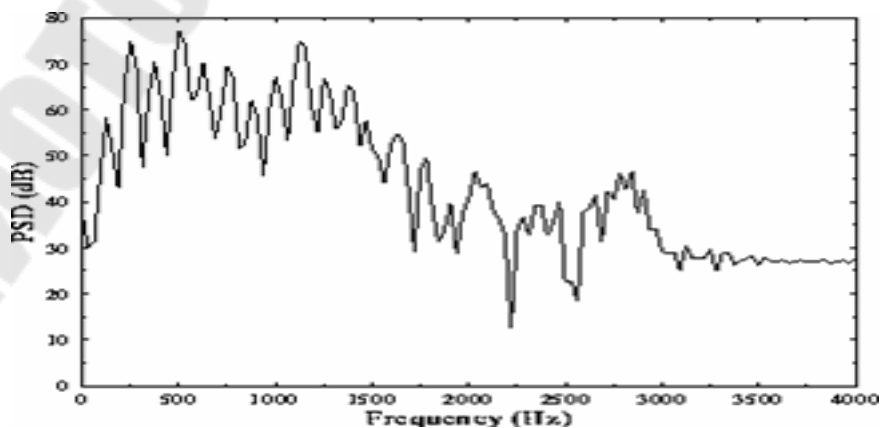


Рисунок 2.3 - Спектр речевого сигнала, содержащего гласные звуки

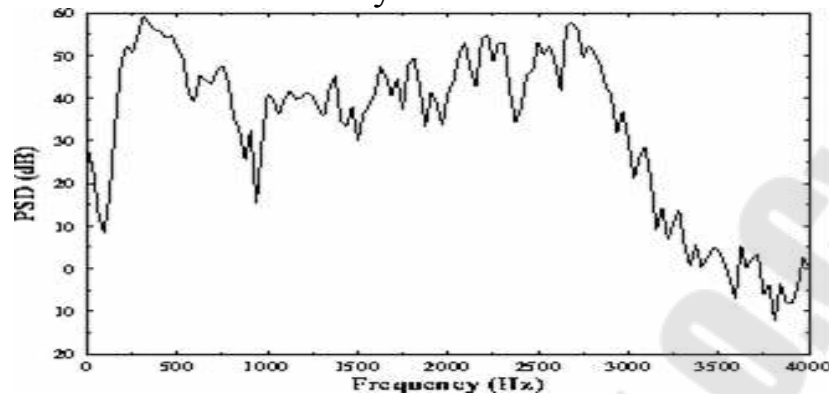


Рисунок 2.4 - Спектр речевого сигнала, содержащего согласные звуки

Второй особенностью речевых сигналов, как это можно отметить из приведенных примеров, является неравномерность распределения вероятностей (плотности вероятности) мгновенных значений сигнала. Малые уровни сигнала значительно более вероятны, чем большие. Особенно это заметно на фрагментах большой длительности с невысокой активностью речи. Этот фактор также обеспечивает возможность экономного кодирования – более вероятные значения могут кодироваться короткими кодами, менее вероятные – длинными.

Еще одна особенность речевых сигналов – их существенная нестационарность во времени: свойства и параметры сигнала на различных участках значительно различаются. При этом размер интервала стационарности составляет порядка нескольких десятков миллисекунд. Это свойство сигнала значительно затрудняет его экономное кодирование и заставляет делать системы сжатия адаптивными, то есть подстраивающимися под значения параметров сигнала на каждом из участков. Наконец, исключительно важным для организации сжатия речевых сигналов является понимание физики механизма речеобразования.

Речь формируется при прохождении выталкиваемого легкими потока воздуха через голосовые связки и голосовой тракт. Голосовой тракт начинается от голосовых связок и заканчивается губами и в среднем имеет длину порядка 15 – 17 сантиметров. Голосовой тракт в силу своих резонансных свойств вносит в формируемый сигнал набор характерных для каждого человека частотных составляющих, называемых формантами. Частоты и полосы этих формант могут управ-

ляться изменением формы голосового тракта, например, изменением положения языка. Важной частью многих голосовых кодеров/декодеров является моделирование голосового тракта как кратковременного фильтра с изменяемыми параметрами. Поскольку форма голосового тракта может изменяться сравнительно медленно (трудно предположить, что можно изменять положение языка чаще, чем 20 – 30 раз в секунду), то параметры такого фильтра должны обновляться (или изменяться) также сравнительно редко (обычно – через каждые 20 миллисекунд или даже реже).

Таким образом, голосовой тракт возбуждается потоком воздуха, направляемым в него через голосовые связки. В зависимости от способа возбуждения возникающие при этом звуки можно разделить на три класса:

1. Гласные звуки, возникающие, когда голосовые связки вибрируют, открываясь и закрываясь, прерывая тем самым поток воздуха от легких к голосовому тракту. Возбуждение голосового тракта при этом производится квазипериодическими импульсами. Скорость (частота) открывания и закрывания связок определяют высоту возникающего звука (тона). Она может управляться изменением формы и напряжения голосовых связок, а также изменением давления подводящего воздушного потока. Гласные звуки имеют высокую степень периодичности основного тона с периодом 2 - 20 мс. Эта долговременная периодичность хорошо видна на рис. 2.1, где приведен фрагмент речевого сигнала с гласным звуком.

2. Согласные звуки, возникающие при возбуждении голосового тракта шумоподобным турбулентным потоком, формируемым проходящим с высокой скоростью через открытые голосовые связки потоком воздуха. В таких звуках, как это видно из рис.2.2, практически отсутствует долговременная периодичность, обусловленная вибрацией голосовых связок, однако кратковременная корреляция, обусловленная влиянием голосового тракта, имеет место.

3. Звуки взрывного характера, возникающие, когда закрытый голосовой тракт с избыточным давлением воздуха внезапно открывается.

Некоторые звуки в чистом виде не подходят ни под один из описанных выше классов, но могут рассматриваться как их смесь.

Из рис. 2.1 и 2.2 хорошо видно, что речевой сигнал обладает высокой степенью кратковременной и долговременной предсказуемости из-за периодичности вибраций голосовых связок и резонансных

свойств голосового тракта. Большинство кодеров/декодеров речи и используют эту предсказуемость, а также медленность изменения параметров модели системы речеобразования для уменьшения скорости кода. При этом все известные способы экономного кодирования речевых сигналов можно условно разделить на три класса, описанные ниже.

## 2.1. Кодирование формы сигнала

Простейшими кодерами/декодерами речи, вообще не использующими информацию о том, как был сформирован кодируемый сигнал, а просто старающимися максимально приблизить восстанавливаемый сигнал по форме к оригиналу, являются кодеры/декодеры формы сигнала. Теоретически они инвариантны к характеру сигнала, подаваемого на их вход, и могут использоваться для кодирования любых, в том числе и неречевых, сигналов. Эти кодеры – самые простые по принципу действия и устройству, но больших степеней сжатия (низких скоростей кода) обеспечить не могут.

Простейшим способом кодирования формы сигнала является импульсно-кодовая модуляция – ИКМ или РСМ – Pulse Code Modulation, при использовании которой производятся просто дискретизация и равномерное квантование входного сигнала, а также преобразование полученного результата в равномерный двоичный код.

Для речевых сигналов со стандартной для передачи речи полосой 0,3 – 3,4 кГц обычно используют частоту дискретизации  $F_d \geq 2F_{\max} = 8 \text{ кГц}$ . Экспериментально показано, что при равномерном квантовании для получения практически идеального качества речи нужно квантовать сигнал не менее чем на  $\pm 2000$  уровней, т.е. для представления каждого отсчета понадобится 12 бит, а результирующая скорость кода будет составлять  $R = 8000 \times 12 = 96000 \text{ (бит / с)} = 96 \text{ (кбит / с)}$ .

Используя неравномерное квантование (более точное для малых уровней сигнала и более грубое для больших его уровней, таким образом, чтобы относительная ошибка квантования была постоянной для всех уровней сигнала), можно достичь того же самого субъективного качества восстановления речевого сигнала, но при гораздо меньшем числе уровней квантования – порядка  $\pm 128$ . В этом случае для двоичного представления отсчетов сигнала понадобится 8 бит и результирующая скорость кода составит 64 кбит/с.

С учетом статистических свойств речевого сигнала (вида распределения вероятностей мгновенных значений), а также нелинейных свойств слуха, гораздо лучше различающего слабые звуки, оптимальной является логарифмическая шкала квантования, которая и была принята в качестве стандарта еще в середине 60-х годов и сегодня повсеместно используется. Правда, в США и Европе стандарты нелинейного квантования несколько различаются ( $\mu$ -law companding и A-law compression), что приводит к необходимости перекодирования сигналов.

Таким образом, исходной для любого сравнения эффективности и качества кодирования речевых сигналов может служить скорость кода, равная 64 кбит/с.

Следующим приемом, позволяющим уменьшить результирующую скорость кода, может быть попытка предсказать значение текущего отсчета сигнала по нескольким предыдущим его значениям, и далее, кодирование уже не самого отсчета, а ошибки его предсказания – разницы между истинным значением текущего отсчета и его предсказанным значением. Если точность предсказания достаточно высока, то ошибка предсказания очередного отсчета будет значительно меньше величины самого отсчета и для ее кодирования понадобится гораздо меньшее число бит. Таким образом, чем более предсказуемым будет поведение кодируемого сигнала, тем более эффективным будет его сжатие.

Описанная идея лежит в основе дифференциальной импульсно-кодовой модуляции – ДИКМ (DPCM) – способа кодирования, при котором кодируются не сами значения сигнала, а их отличия от некоторым образом предсказанных значений. Простейшим способом предсказания является использование предыдущего отсчета сигнала в качестве предсказания его текущего значения. Это так называемое предсказание нулевого порядка, самое простое, но и наименее точное. Более точным, очевидно, будет предсказание текущего отсчета на основе линейной комбинации двух предшествующих и т.д.

К сожалению, точность предсказания не всегда растет с ростом порядка предсказания, поскольку свойства сигнала между отсчетами начинают уже изменяться, поэтому обычно ограничиваются предсказанием не выше 2 – 3-го порядка.

На рис. 2.5 и 2.6 приведены схемы ДИКМ кодера и декодера.

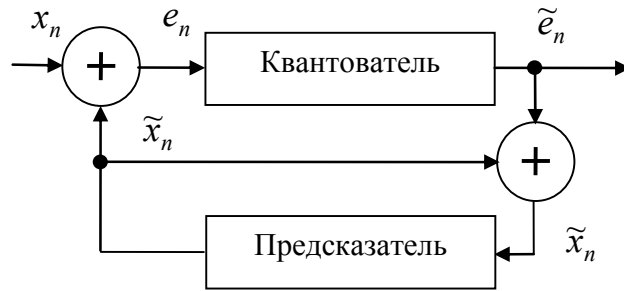


Рисунок 2.5 - ДИКМ кодер

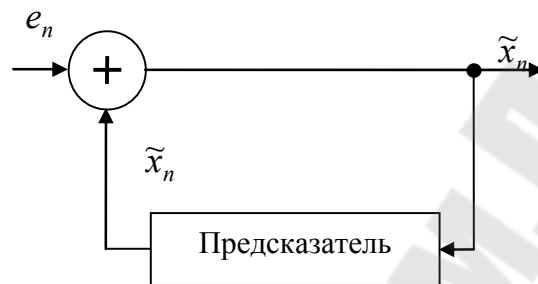


Рисунок 2.6 - ДИКМ декодер

При кодировании речевых сигналов с учетом степени их кратковременной (на несколько очередных отсчетов) предсказуемости результирующая скорость кода для ДИКМ (DPCM) обычно составляет 5 – 6 бит на отсчет или 40 – 48 кбит/с.

Эффективность ДИКМ может быть несколько повышена, если предсказание и квантование сигнала будет выполняться не на основе некоторых усредненных его характеристик, а с учетом их текущего значения и изменения во времени, то есть адаптивно. Так, если скорость изменения сигнала стала большей, можно увеличить шаг квантования, и, наоборот, если сигнал стал изменяться гораздо медленнее, величину шага квантования можно уменьшить. При этом ошибка предсказания уменьшится и, следовательно, будет кодироваться меньшим числом бит на отсчет. Такой способ кодирования называется адаптивной ДИКМ, или АДИКМ (ADPCM). Сегодня такой способ кодирования стандартизован и широко используется при сжатии речи в междугородных цифровых системах связи, в системе микросотовой связи DECT, в цифровых бесшнуровых телефонах и т.д. Использование АДИКМ со скоростью кода 4 бита/отсчет или 32 кбит/с обеспечивает такое же субъективное качество речи, что и 64 кбит/с - ИКМ, но при вдвое меньшей скорости кода.

На сегодня стандартизованы также АДИКМ – кодеки для скоростей 40, 24 и 16 кбит/с (в последнем случае с несколько худшим, чем

для 32 кбит/с – АДИКМ, качеством сигнала). Таким образом, видно, что сжатие речевых сигналов на основе кодирования их формы обеспечивает в лучшем случае двух - трехкратное уменьшение скорости кода. Дальнейшее снижение скорости ведет к резкому ухудшению качества кодируемого сигнала.

## 2.1. Субполосные кодеры

Описанные выше кодеры формы сигнала использовали чисто временной подход к описанию этого сигнала. Однако возможны и другие подходы. Примером может служить так называемое кодирование поддиапазонов (Sub-Band Coding - SBC), при котором входной сигнал разбивается (или расфильтровывается) на несколько частотных диапазонов (поддиапазонов - sub-bands) и сигнал в каждом из этих поддиапазонов кодируется по отдельности, например, с использованием техники АДИКМ.

Поскольку каждый из частотных поддиапазонов имеет более узкую полосу (все поддиапазоны в сумме дают полосу исходного сигнала), то и частота дискретизации в каждом поддиапазоне также будет меньше. В результате суммарная скорость всех кодов будет по крайней мере не больше, чем скорость кода для исходного сигнала. Однако у такой техники есть определенные преимущества. Дело в том, что субъективная чувствительность слуха к сигналам и их искажениям различна на разных частотах. Она максимальна на частотах 1 - 1,5 кГц и уменьшается на более низких и более высоких частотах. Таким образом, если в диапазоне более высокой чувствительности слуха квантовать сигнал более точно, а в диапазонах низкой чувствительности более грубо, то можно получить выигрыш в результирующей скорости кода. Действительно, при использовании технологии кодирования поддиапазонов получено хорошее качество кодируемой речи при скорости кода 16 – 32 кбит/с. Кодер получается несколько более сложным, чем при простой АДИКМ, однако гораздо проще, нежели для других эффективных способов сжатия речи.

Упрощенная схема подобного кодера (с разбиением на 2 поддиапазона) приведена на рис. 2.7.



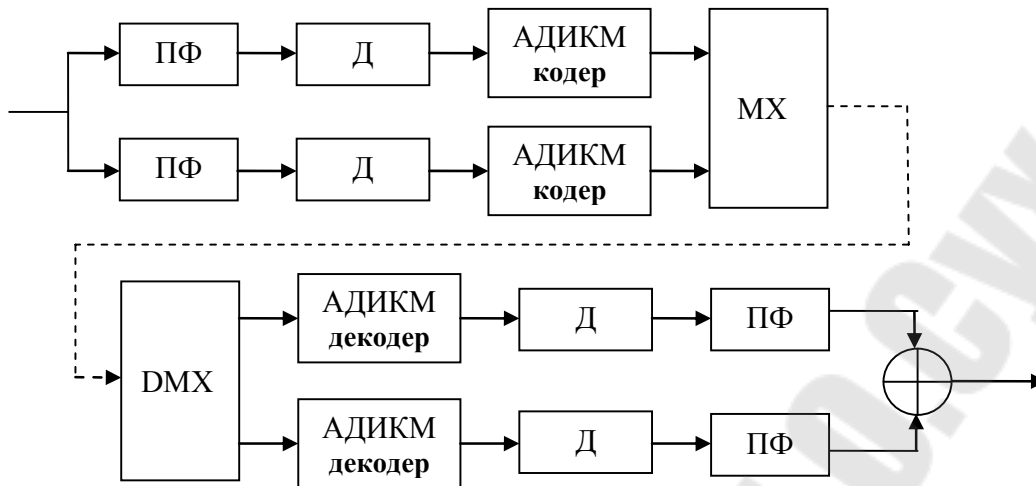


Рисунок 2.7 - Упрощенная схема субполосного кодера с разбиением на 2 поддиапазона

## 2.2. Кодеры основанные на спектральных преобразованиях

Близким к кодированию поддиапазонов является метод сжатия, основанный на применении к сигналу линейных преобразований, к примеру, дискретного косинусного или синусного преобразования. Для кодирования речи используется так называемая технология АТС (Adaptive Transform Coding), при которой сигнал разбивается на блоки, к каждому блоку применяется дискретное косинусное преобразование и полученные коэффициенты адаптивно, в соответствии с характером спектра сигнала, квантуются. Чем более значимыми являются коэффициенты преобразования, тем большим числом бит они кодируются. Техника очень похожа на JPEG, но применяется к речевым сигналам. Достигаемые при таком кодировании скорости кодов составляют 12 – 16 кбит/с при вполне удовлетворительном качестве сигнала. Широкого распространения для сжатия речи этот метод не получил, поскольку известны гораздо более эффективные и простые в исполнении методы кодирования.

## 2.3. Кодирование источника. Параметрические кодеры

В отличие от кодеров формы сигнала, вообще не использующих информацию о том, как был сформирован кодируемый сигнал, кодеры источника основываются именно на модели источника и из кодируемого сигнала извлекают информацию о параметрах этой модели.

При этом результатом кодирования являются не коды сигналов, а коды параметров источника этих сигналов.

Кодеры источника для кодирования речи называются вокодерами (VOICE CODERS) и работают примерно следующим образом.

Голосообразующий тракт представляется как линейный фильтр с переменными во времени параметрами, возбуждаемый либо источником белого шума (при формировании согласных звуков), либо последовательностями импульсов с периодом основного тона (при формировании гласных звуков) рис. 2.8.

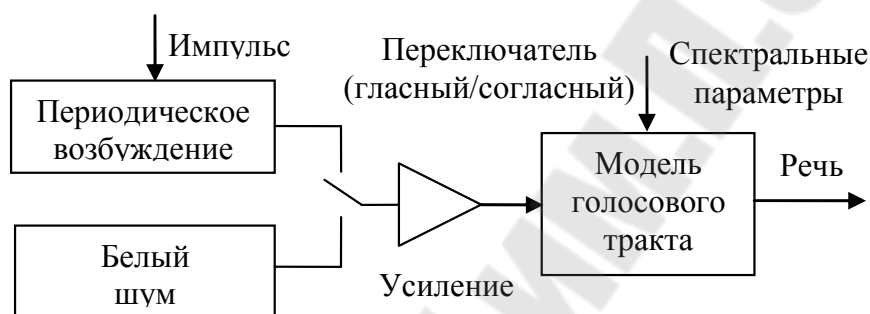


Рисунок 2.8 - Модель речеобразования

Информация, которую получает вокодер в результате анализа речевого сигнала и передает декодеру, это параметры речеобразующего фильтра, указатель гласный/негласный звук, мощность сигнала возбуждения и период основного тона для гласных звуков. Эти параметры должны обновляться каждые 10 – 20 мс, чтобы отслеживать нестационарность речевого сигнала.

Вокодер, в отличие от кодера формы сигнала, пытается сформировать сигнал, звучащий как оригинальная речь, и не обращает внимания на отличие формы этого сигнала от исходного. При этом результирующая скорость кода на его выходе обычно составляет не более 2,4 кбит/с, то есть в пятнадцать раз меньше, чем при АДИКМ.

К сожалению, качество речи, обеспечиваемой вокодерами, очень далеко от идеального, ее звучание хотя и достаточно разборчиво, но абсолютно ненатурально. При этом даже существенное увеличение скорости кода практически не улучшает качества речи, поскольку для кодирования была выбрана слишком простая модель системы речеобразования. Особенно грубым является предположение о том, что речь

состоит лишь из гласных и согласных звуков, не допускающее каких либо промежуточных состояний.

Основное применение вокодеры нашли в военной области, где главное – это не натуральность речи, а большая степень ее сжатия и очень низкая скорость кода, позволяющая эффективно защищать от перехвата и засекречивать передаваемую речь. Кратко рассмотрим основные из известных типов вокодеров.

## 2.4. Канальные вокодеры

Это наиболее древний тип вокодера, предложенный еще в 1939 году. Этот вокодер использует слабую чувствительность слуха человека к незначительным фазовым (временным) сдвигам сигнала. Для сегментов речи длиной примерно в 20 - 30 мс с помощью набора узкополосных фильтров определяется амплитудный спектр. Чем больше фильтров, тем лучше оценивается спектр, но тем больше нужно бит для его кодирования и тем больше результирующая скорость кода. Сигналы с выходов фильтров детектируются, пропускаются через ФНЧ, дискретизируются и подвергаются двоичному кодированию (рис. 2.9).

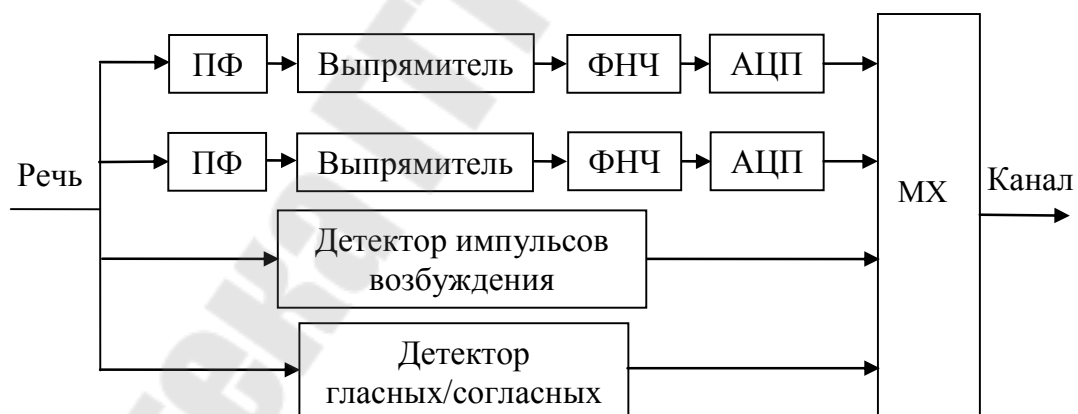


Рисунок 2.9 - Структурная схема канального вокодера – блок кодера

Таким образом, определяются медленно изменяющиеся параметры голосообразующего тракта и, кроме того, с помощью детекторов основного тона и гласных звуков, период основного тона возбуждения и признак гласный/негласный звук.

Канальный вокодер может быть реализован как в цифровой, так и в аналоговой форме и обеспечивает достаточно разборчивую речь при скорости кода на его выходе порядка 2,4 кбит/с.

Декодер (рис. 2.10), получив информацию, вырабатываемую кодером, обрабатывает ее в обратном порядке, синтезируя на своем выходе речевой сигнал, в какой-то мере похожий на исходный. Канальные вокодеры используются в основном только там, где главным образом необходимы разборчивость и высокая степень сжатия: в военной связи, авиации, космической связи и т.д.

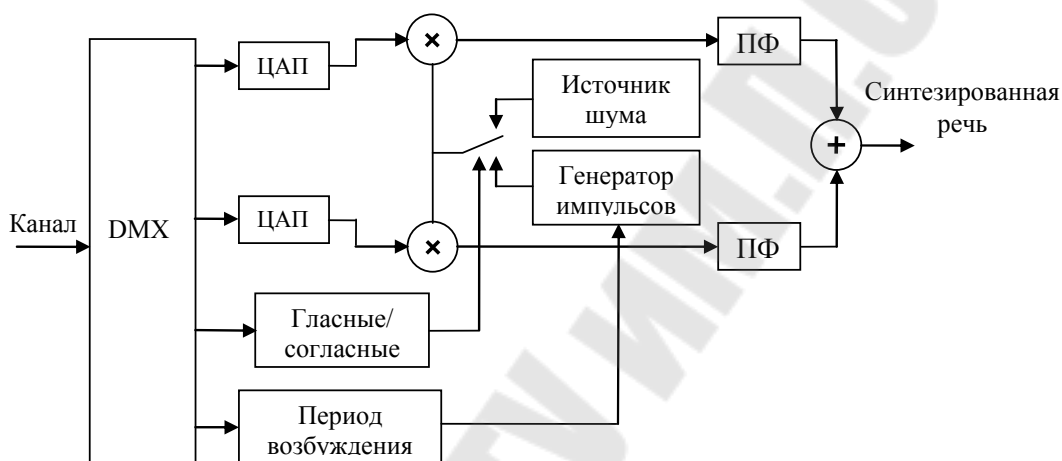


Рисунок 2.10 - Структурная схема канального вокодера – блок декодера

## 2.5. Гомоморфный вокодер

Гомоморфная обработка сигналов представляет собой один из нелинейных методов обработки, который может эффективно применяться к сложным сигналам, например к речевым. С учетом используемой в вокодерах модели системы голосообразования речевой сигнал можно представить как временную свертку импульсной переходной характеристики голосового тракта с сигналом возбуждения. В частотной области это соответствует произведению частотной характеристики голосового тракта и спектра сигнала возбуждения. Наконец, если взять логарифм от этого произведения, то получим сумму логарифмов спектра сигнала возбуждения и частотной характеристики голосового тракта. Поскольку человеческое ухо практически не чувствительно к фазе сигнала, можно оперировать с амплитудными спектрами:

$$\log(|S(e^{j\omega})|) = \log(|P(e^{j\omega})|) + \log(|V(e^{j\omega})|)$$

где  $S(e^{j\omega})$  – спектр речи,  $P(e^{j\omega})$  – спектр сигнала возбуждения и  $V(e^{j\omega})$  – частотная характеристика голосового тракта.

Если теперь выполнить над  $\log(|S(e^{j\omega})|)$  обратное преобразование Фурье (ОПФ), то получим так называемый кепстр сигнала. Параметры голосового тракта изменяются во времени сравнительно медленно (их спектр находится в области низких частот – НЧ), тогда как сигнал возбуждения – быстроосциллирующая функция (ее спектр сосредоточен в области высоких частот - ВЧ). Поэтому в кепстре речевого сигнала эти составляющие разделяются (рис. 2.11) и могут быть закодированы по отдельности.

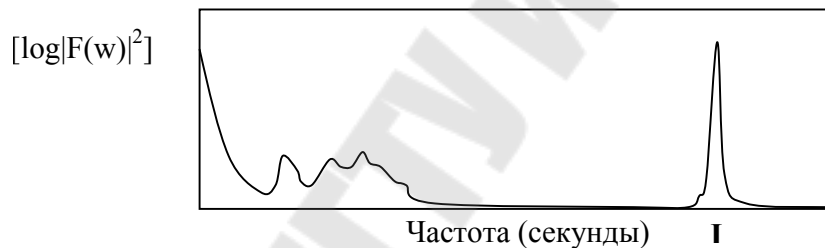


Рисунок 2.11 - Спектр речевого сигнала

Схема гомоморфного кодера/декодера речи приведена на рис. 2.12, с его использованием можно получить скорость кода порядка 4 кбит/с.

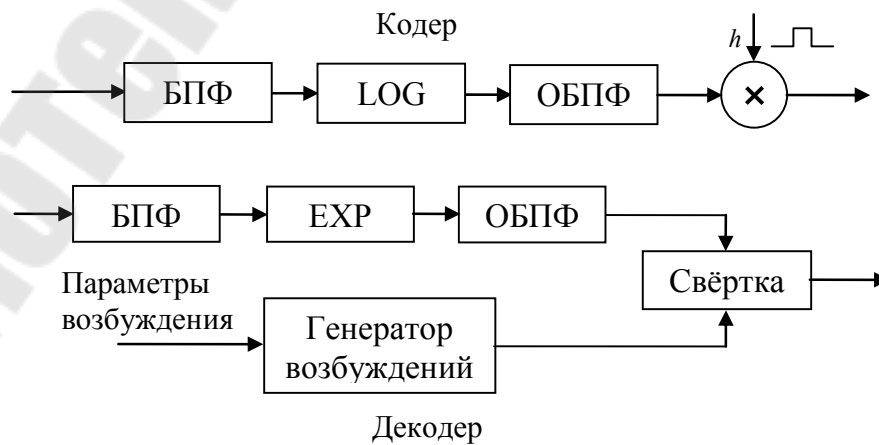


Рисунок 2.12 - Схема гомоморфного кодера/декодера речи

## 2.6. Формантные вокодеры

Как уже отмечалось ранее, основная информация о речевом сигнале содержится в положении и ширине составляющих его формант. Если с высокой точностью определять и кодировать параметры этих формант, можно получить очень низкую результирующую скорость кода – менее 1 кбит/с. К сожалению, сделать это очень трудно, поэтому формантные кодеры речи пока не нашли широкого распространения.

## 2.7. Вокодеры с линейным предсказанием

Вокодеры на основе линейного предсказания используют такую же модель речеобразования, что и остальные из рассмотренных выше. Что их отличает – это метод определения параметров тракта. Линейные предсказывающие кодеры, или ЛПК, полагают голосовой тракт линейным фильтром с непрерывной импульсной переходной характеристикой, в котором каждое очередное значение сигнала может быть получено как линейная комбинация некоторого числа его предыдущих значений.

В ЛПК-вокодере речевой сигнал делится на блоки длиной около 20 мс, для каждого из которых определяются коэффициенты предсказывающего фильтра. Эти коэффициенты квантуются и передаются декодеру. Затем речевой сигнал пропускается через фильтр, частотная характеристика которого обратна частотной характеристике голосового тракта. На выходе фильтра получается ошибка предсказания. Назначение предсказателя – устранить корреляцию между соседними отсчетами сигнала. В результате гораздо отчетливее проявляется долговременная корреляция в сигнале, что позволяет точнее определить частоту основного тона и выделить признак гласный/согласный звук.

Вокодеры на основе линейного предсказания сейчас наиболее популярны, поскольку все используемые ими фильтровые модели речевого тракта работают очень хорошо. Получаемые с их помощью скорости кодов при неплохом качестве речи составляют до 2,4 кбит/с.

## 2.8. Гибридные методы кодирования речи

Гибридные, или комбинированные, методы кодирования речи заполняют разрыв между кодерами формы сигнала, совершенно не учитывающими его природы, и кодерами источника, кодирующими, по сути, не сигнал, а параметры модели порождающего его источника. Как отмечалось ранее, кодеры формы сигнала обеспечивают очень хорошее качество речи при скоростях кодирования выше 16 кбит/с, но вообще не работают при более низких скоростях, тогда как вокодеры обеспечивают разборчивую речь при скоростях кодирования 2,4 кбит/с и ниже, но не могут дать хорошего качества при любой скорости кода.

Наиболее распространенными в настоящее время являются гибридные методы кодирования, работающие во временной области (то есть с сигналом, а не его спектром или другими линейными преобразованиями), основанные на анализе сигнала через его синтез (так называемые ABS-кодеки). Эти кодеры так же, как и вокодеры, используют модель голосового тракта, но несколько иным образом – для подбора сигнала возбуждения, обеспечивающего наилучшее совпадение синтезированного на ее основе речевого сигнала с исходным.

ABS-кодеры были впервые предложены сравнительно недавно – в 1982 году - и в своем первоначальном виде получили название MPE-кодеров (Multi-Pulse Excited - кодеры с многоимпульсным возбуждением). Позднее были предложены более совершенные RPE-кодеры (Regular-Pulse Excited – кодеры с регулярным импульсным возбуждением) и CELP-кодеры (Codebook-Excited Linear Predictive – с возбуждением на основе кодовых книг). Сегодня существуют и другие их разновидности, но все они используют общую идею.

Чтобы понять, на чем основаны эффективность и качество ABS-кодера, сначала рассмотрим работу так называемого RELP-кодера (Residual Excited Linear Prediction - RELP).

Если речевой сигнал (имеющий спектр рис. 2.13, а) пропустить через линейный предсказатель (с частотной характеристикой вида рис.2.13, б), то корреляция между отсчетами выходного сигнала (ошибки предсказания) значительно уменьшится. Если предсказание выполнялось достаточно хорошо, то выходом предсказателя будет практически белый шум с равномерным спектром (рис. 2.13, в).

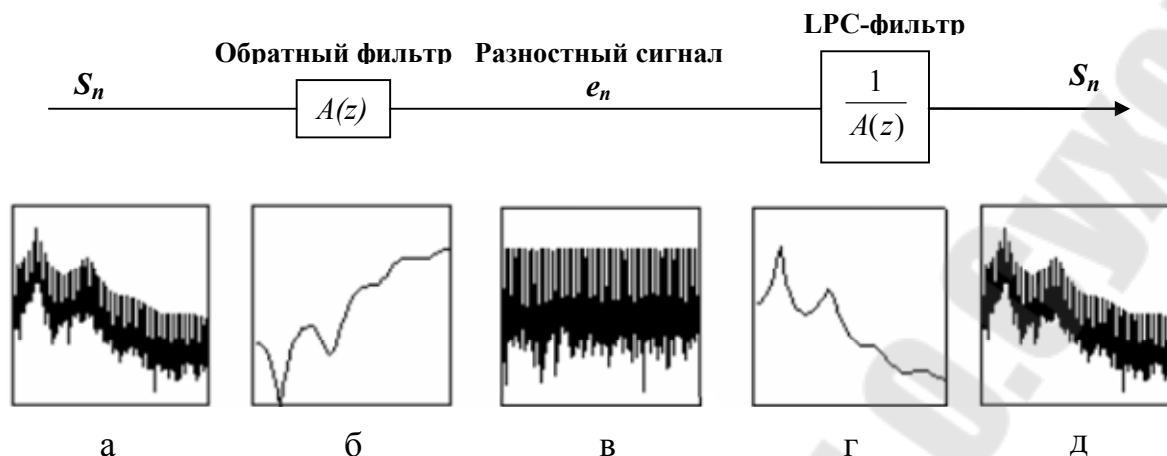


Рисунок 2.13 - Анализ речевого сигнала с помощью коэффициентов линейного предсказания

Вместе с тем этот белый шум (ошибка предсказания) несет всю информацию о кодируемом речевом сигнале, и если его пропустить снова через LPC-фильтр (с частотной характеристикой - рис. 5.13,г), то мы абсолютно точно восстановим исходный речевой сигнал. Поскольку эта информация распределена по спектру ошибки предсказания более или менее равномерно, то возникла идея кодировать и передавать только небольшую часть спектра ошибки предсказания, а остальное восстанавливать в декодере.

В RELP-кодере сигнал ошибки предсказания пропускается через низкочастотный фильтр с частотой среза около 1 кГц. Сигнал с выхода фильтра кодируется по форме, например ДИКМ-кодером. В декодере ошибка предсказания восстанавливается путем ее переноса в область удаленных низкочастотным фильтром кодера частот. RELP-кодер работал бы идеально, если бы в процессе линейного предсказания мы получали белый шум. Однако из-за наличия в речевом сигнале квазипериодических формантных составляющих линейный предсказатель не может устранить долговременной корреляции с периодом основного тона формант и они будут явно присутствовать в спектре ошибки предсказания. Если теперь пропустить ошибку предсказания через ФНЧ, то высокочастотные формантные составляющие будут утеряны и в дальнейшем не смогут быть восстановлены.

RELP-кодеры позволяют получить неплохое качество сигнала при скорости кода порядка 9.6 кбит/с, однако им в некоторой степени присущ недостаток вокодеров – синтетический характер восстановленной речи. В связи с этим на смену им практически повсеместно



пришли похожие по принципу работы ABS-кодеры в их разновидностях.

ABS-кодер работает следующим образом. Кодированный входной сигнал (уже в цифровой форме, в виде потока отсчетов) разбивается на фрагменты длиной порядка 20 мс, в пределах которых свойства сигнала изменяются незначительно. Для каждого из этих фрагментов определяются текущие параметры синтезирующего фильтра (аналога голосового тракта) и далее подбирается сигнал возбуждения, который, будучи пропущенным через синтезирующий фильтр, минимизирует ошибку между входным и синтезированным сигналами.

Таким образом, название метода Analysis-by-Synthesis состоит в том, что кодер анализирует входную речь посредством синтеза множества приближений к ней. В конечном итоге кодер передает декодеру информацию, представляющую собой комбинацию текущих параметров синтезирующего фильтра и сигнала возбуждения. Желательно, чтобы этих данных было поменьше. Декодер по этим параметрам восстанавливает закодированную речь, причем делает это так же, как это делал кодер в процессе анализа через синтез. Различие между ABS-кодерами разного типа состоит в том, как в каждом из них подбирается сигнал возбуждения синтезирующего фильтра  $u(n)$ . Теоретически на вход синтезирующего фильтра нужно подать бесконечно большое число различных сигналов возбуждения, чтобы посмотреть, какой сигнал получится на его выходе, и сравнить его с кодируемым. Сигнал возбуждения, который даст минимум взвешенной ошибки между оригиналом и синтезированной речью, выбирается в качестве результата кодирования. Именно эта замкнутая схема определения сигнала возбуждения (рис. 2.14) и обеспечивает ABS-кодерам высокое качество кодируемой речи при низких скоростях кода.

Проблема состоит в большом количестве вычислительных операций, необходимых для подбора наилучшего сигнала возбуждения. Но для сегодняшних возможностей вычислительной и микропроцессорной техники это вполне разрешимая задача.

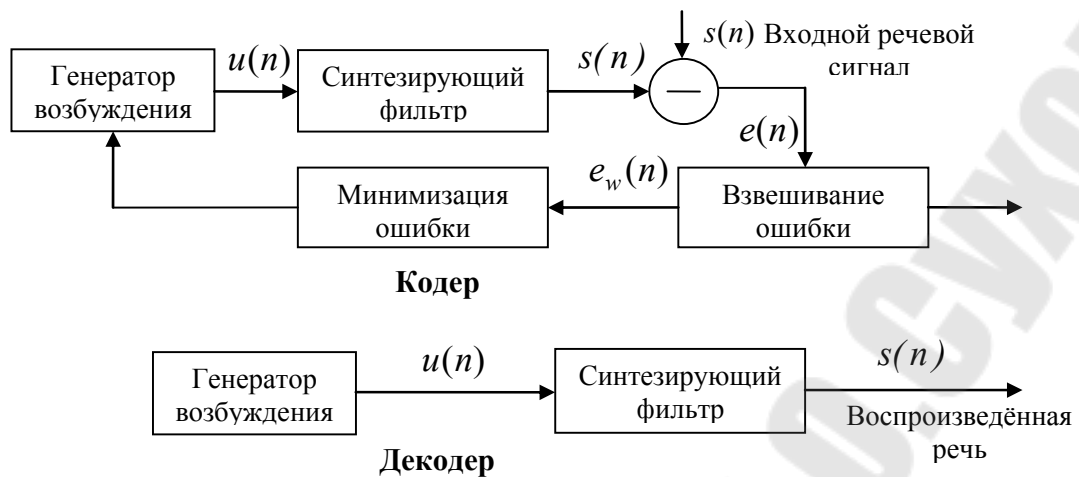


Рисунок 2.14 - Структурная схема ABS-кодера/декодера

## 2.9. Многоимпульсные кодеры ( МРЕ-кодеры)

Как уже говорилось, при прохождении речевого сигнала через предсказывающий фильтр корреляция между его соседними отсчетами значительно уменьшается. Однако для гласных звуков наличие формантных составляющих приводит к появлению в речевом сигнале квазипериодичности и высокой долговременной корреляции. Эта периодичность не устраняется линейным предсказанием и приводит к появлению в сигнале ошибки предсказания высокоамплитудных скачков.

Чтобы устранить долговременную корреляцию, можно пропустить сигнал ошибки предсказания через второй линейный предсказатель. Этот линейный предсказатель должен устранить корреляцию уже не между соседними отсчетами речевого сигнала, а между соседними периодами ошибки предсказания. Это достигается введением в предсказатель временной задержки на величину периода основного тона речевого сигнала:

$$P(z) = 1 - \sum_i \beta_i z^{-M-i},$$

где  $M$  – период основного тона.

На приведенном ниже рис.2.15 изображены: а – исходный речевой сигнал; б – сигнал ошибки кратковременного линейного предсказания (увеличенный в 3 раза); в – сигнал на выходе двухкаскадного

(кратковременного + долговременного) предсказателя (увеличенный в 10 раз).

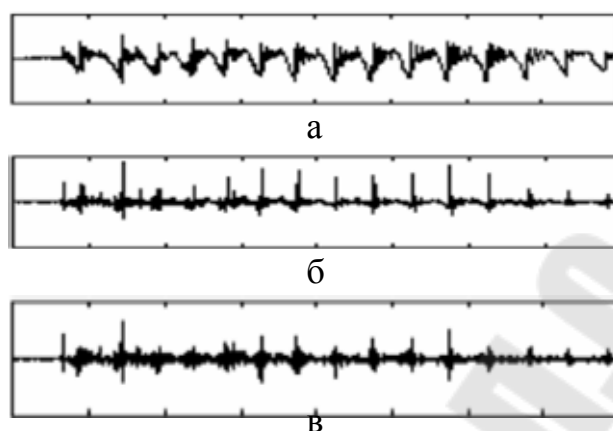


Рисунок 2.15 - Представление речевого сигнала

Если теперь подать результирующий сигнал ошибки предсказания в качестве возбуждения на последовательно соединенные кратковременный и долговременный фильтры-предсказатели, то на выходе получим исходный неискаженный речевой сигнал. Можно было бы кодировать и передавать по каналу связи полученный сигнал ошибки предсказания, и уже это обеспечивало бы определенную экономию из-за существенно меньшей его амплитуды по сравнению с исходным речевым сигналом. Однако форма сигнала все же остается довольно сложной, что требует для его кодирования достаточно много бит. В многоимпульсных кодерах (MPE) в качестве сигнала возбуждения  $u(n)$  берут не ошибку предсказания, а просто последовательность из четырех – шести коротких импульсов. Временное положение каждого из этих импульсов и их амплитуды определяются в процессе процедуры анализа через синтез (ABS) до достижения минимальных различий между исходным и синтезированным речевыми сигналами. Параметры импульсов возбуждения, минимизирующие ошибку, подбирают последовательно, сначала для первого импульса, затем для второго и т.д. На практике достаточно задавать положение импульсов с шагом около 1 мс и точностью амплитуд до 5 %, и это обеспечивает хорошее качество синтезируемого звука при скорости кода около 10 кбит/с.

## **2.10. Кодеры с регулярным импульсным возбуждением (RPE-кодеры)**

Так же как и MPE-кодек, Regular Pulse Excited, или RPE-кодек, использует в качестве сигнала возбуждения  $u(n)$  фиксированный набор коротких импульсов. Однако в этом кодеке импульсы расположены регулярно на одинаковых расстояниях друг от друга, и кодеру необходимо определить лишь положение первого импульса и амплитуды всех импульсов. Таким образом, декодеру нужно передавать меньше информации о положении импульсов, следовательно, в сигнал возбуждения можно включить их большее количество и тем самым улучшить приближение синтезированного сигнала к оригиналу. К примеру, если при скорости кода 10 кбит/с в MPE-кодеке используется четырехимпульсный сигнал возбуждения, то в RPE-кодеке можно использовать уже десятиимпульсный сигнал. При этом существенно повышается качество речи.

Метод регулярного импульсного возбуждения RPE сегодня широко применяется, в том числе в системе сотовой связи GSM.

## **2.11. Кодеры с возбуждением на основе кодовых книг (CELP-кодеры)**

Методы кодирования MPE и RPE обеспечивают хорошее качество кодируемой речи при скоростях кода порядка 10 кбит/с и выше, но начинают сильно искажать сигнал при более низких скоростях. Дело в том, что для описания необходимых параметров сигнала возбуждения – временного положения и амплитуд импульсов с требуемой точностью просто не хватает бит.

В связи с этим был предложен метод, использующий в качестве сигнала возбуждения не импульсные последовательности, задаваемые набором своих параметров, а библиотеки (кодовые книги) специальным образом подготовленных и записанных в запоминающее устройство сигналов возбуждения различной формы – Codebook Excited Linear Prediction ( CELP ).

Схема формирования сигнала возбуждения CELP-кодера приведена на рис. 2.16.

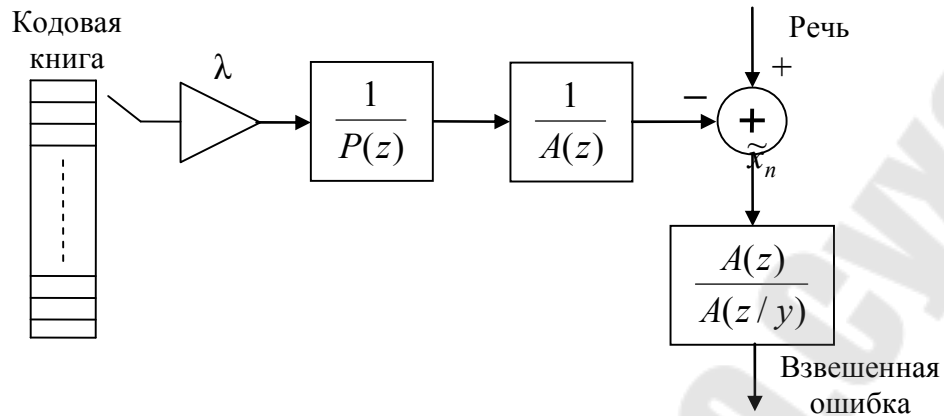


Рисунок 2.16 - Схема формирования сигнала возбуждения CELP-кодера

Результатом кодирования при этом являются не параметры импульсов сигнала возбуждения, а индекс кодовой книги (номер хранящегося в ней образца сигнала возбуждения), а также его амплитуда. Если кодовая книга содержит, к примеру, 1024 сигнала, а амплитуда сигнала кодируется с точностью 2 – 3 %, то необходимое число бит составит 10 (для индекса) + 5 (для амплитуды) = 15 бит на фрагмент сигнала длительностью в 20 мс (в сравнении с 47 битами, используемыми в GSM RPE-кодеке). Правда, процедура кодирования требует очень больших вычислительных затрат, поэтому реализация CELP-кодеров стала возможной только в последнее время с использованием специализированных сигнальных процессоров с производительностью порядка 300 млн. операций в секунду и более.

Кодирование на основе алгоритма CELP с успехом используется в современных системах связи при скоростях кода от 16 до 4,8 кбит/с. При этом для скорости кода 16 кбит/с CELP обеспечивается такое же качество речи, как и для 64 кбит/с ИКМ, а при скорости кода 4,8 кбит/с – как для 13 кбит/с GSM RPE.

## 2.12. Психоакустические принципы сжатия аудиосигналов

Человек в состоянии принимать огромные потоки информации. Но сознательно он способен обрабатывать лишь около 100 бит/с информации. Поэтому можно говорить о присущей аудиосигналам существенной избыточности. Важной проблемой при цифровом представлении звуковых сигналов является сокращение имеющейся в них статистической и психоакустической (перцептивной) избыточности.

Сокращение статистической избыточности основывается на учете свойств самих звуковых сигналов. Она обусловлена наличием корреляционной связи между соседними отсчетами звукового сигнала при его дискретизации. Устранение статистической избыточности не приводит к значительному уменьшению скорости цифрового потока. Наиболее перспективными с этой точки зрения оказались методы сокращения психоакустической избыточности звуковых сигналов, учитывающие такие свойства слухового восприятия. Степень учета психоакустических закономерностей слухового восприятия определяет качество систем кодирования со сжатием цифровых данных. Методами устранения психофизической избыточности можно обеспечить сжатие цифровых аудиоданных в 10 – 12 раз без существенных потерь в качестве.

Описание характеристик любой слуховой системы осуществляется по частотно-пороговым кривым, которые отражают предельные значения интенсивности (или мощности звукового давления) в зависимости от частоты звуковых колебаний. Частотно-пороговые кривые, полученные для слуховой системы в целом, получили название кривых слышимости. Чувствительность слуха оценивается минимальной интенсивностью звука, при которой человек может отличить звук от постоянно существующего фона собственных шумов. Интенсивность, при которой звук обнаруживается с вероятностью 0,5 (в 50% случаев), называется порогом слышимости или абсолютным порогом для данного сигнала.

Кривая порога слышимости (рис.2.17), характеризующая наименьшую интенсивность звука определенной частоты, который может быть услышан человеком в тишине, хорошо аппроксимируется с помощью нелинейной функции

$$T(f) = 3,64(f/1000)^{-0,8} - 6,5e^{-0,6(f/1000-3,3)^2} + 10^{-3}(f/1000)^4,$$

где  $T(f)$  – порог слышимости, дБ;  $f$  – частота звукового сигнала, Гц. Данная кривая представляет собой геометрическое место точек, которые соответствуют тональным компонентам разных частот, имеющих одинаковую громкость.

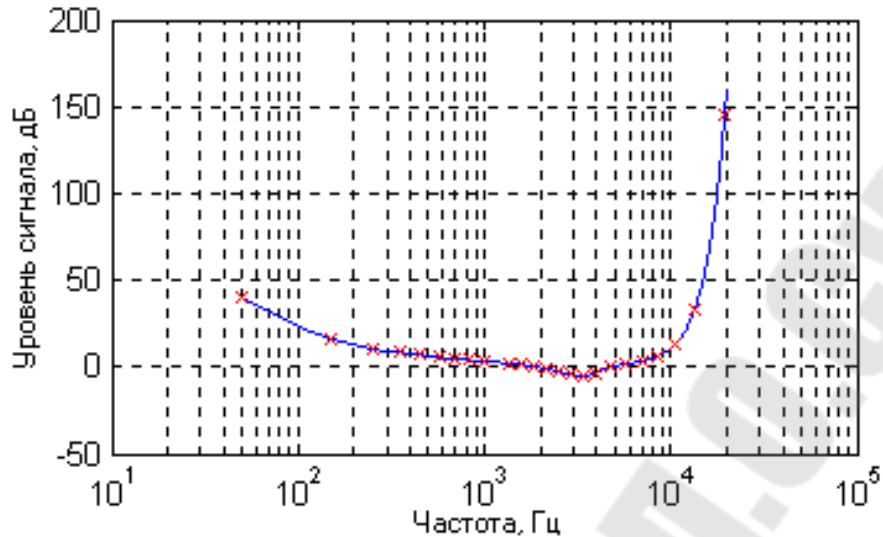


Рисунок 2.17 - Кривая минимального порога слышимости слухового аппарата человека в тишине

Анализ усредненного графика порога слышимости показывает, что слуховая система человека чувствительна к частотам в диапазоне 0,01–20 кГц. Наибольшая чувствительность у человека наблюдается на средних частот 2–5 кГц. Порог слышимости сигнала на частоте около 3 кГц составляет приблизительно 0 дБ. Порог слышимости значительно повышается в области низких (менее 200 Гц) и высоких (более 10 кГц) частот и может достигать значения громкости звука 40 дБ и выше. Следовательно, для слабых сигналов амплитудно-частотная характеристика слуха является нелинейной, и, как следствие, слабые низкочастотные и высокочастотные составляющие сигнала могут быть неслышны. Кривые слышимости позволяют оценить работоспособность слуховой системы по ряду признаков: диапазону воспринимаемых частот, полосе частот, в пределах которой чувствительность максимальна, абсолютным значениям звукового давления и их изменениям в зависимости от частоты.

На рис. 2.18 схематически показан метод определения ширины критической полосы слуховой системы.

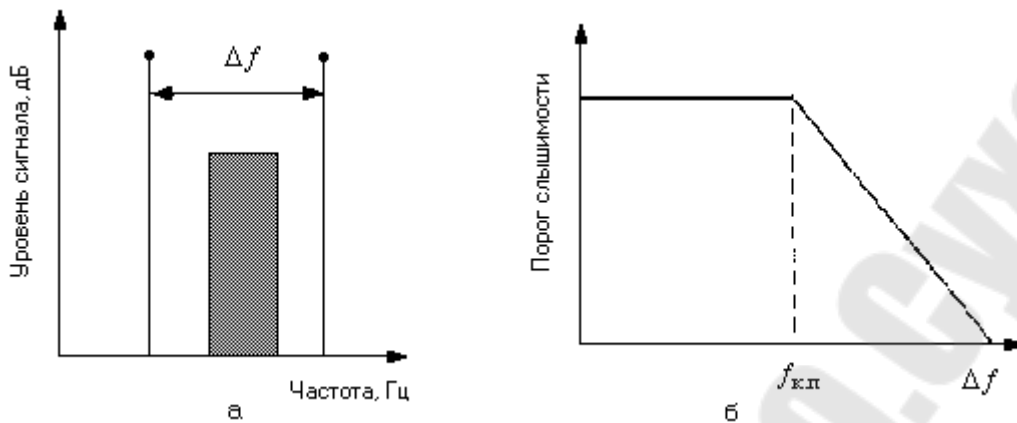


Рисунок 2.18 - Определение ширины критической полосы при маскировании узкополосного шумового компонента двумя тональными компонентами

Узкополосный шумовой сигнал маскируется двумя близко расположенными тональными компонентами, расстояние  $\Delta f$  между которыми постепенно увеличивается. На рис. 2.18,б представлена зависимость поведения порога обнаружения шумовой компоненты от изменения расстояния между двумя тональными компонентами  $\Delta f$ . Данный порог остается постоянным до тех пор, пока частотное расстояние между тональными компонентами остается в пределах критической полосы  $f_{к.п}$ . Вне полосы порог быстро уменьшается.

На рис. 2.19 приведен случай, когда в качестве маскируемого сигнала выступает тональная компонента, а в качестве маскирующих сигналов – две узкополосные шумовые компоненты (рис. 2.19,а). Поведение порога обнаружения для тональной компоненты (рис. 2.19,б) аналогично рис. 2.18,б.

В таблице 2.1 представлено разбиение слышимого диапазона частот на критические полосы с соответствующими центральной частотой и шириной полос идеализированного банка фильтров с прямоугольными амплитудно-частотными характеристиками, который позволяет качественно воспроизвести звуковые сигналы. Из таблицы видно, что критическая полоса не соответствует диапазону с фиксированными нижней и верхней границами.



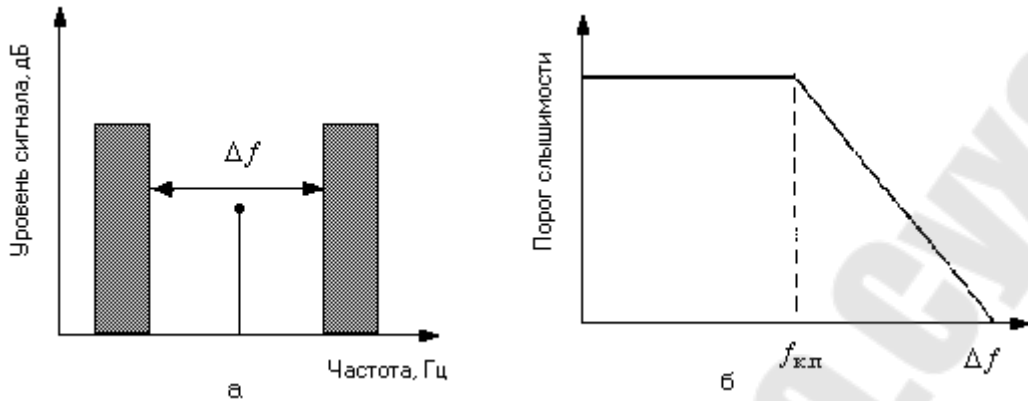


Рисунок 2.19 - Определение ширины критической полосы при маскировании тонального компонента двумя узкополосными шумовыми компонентами

Ширина критической полосы остается приблизительно постоянной (около 100 Гц) вплоть до 500 Гц и увеличивается приблизительно на 20% от центральной частоты при 500 Гц и выше:

$$BW(f) = \begin{cases} 100 \text{ Гц}, & f < 500 \text{ Гц}, \\ 0,2 f \text{ Гц}, & f \geq 500 \text{ Гц}. \end{cases}$$

Из выражения видно, что в частотной области до 500 Гц органы слуха воспринимают интенсивность звука, разделяя ее на участки постоянной абсолютной ширины, а свыше 500 Гц – на участки постоянной относительной ширины.

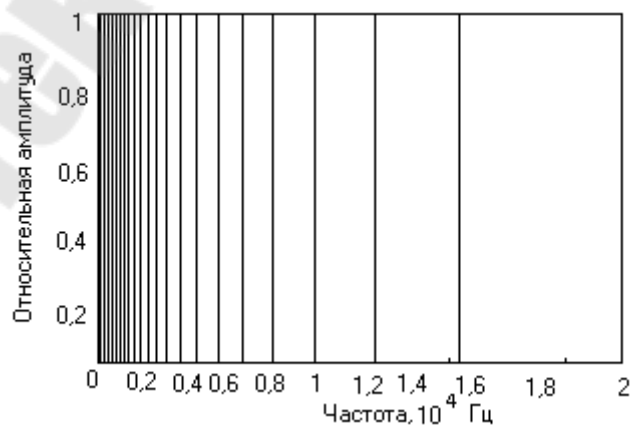


Рисунок 2.20 - Идеализированный банк фильтров как упрощенная модель слуховой системы

Таблица 2.1

Разбиение слышимого частотного диапазона на критические  
полосы в психоакустической модели слуховой системы

Номер критической полосы	Центральная частота, Гц	Нижняя и верхняя частоты, ширина критической полосы, Гц	
1	50	20–100	100
2	150	100–200	100
3	250	200–300	100
4	350	300–400	100
5	450	400–510	110
6	570	510–630	120
7	700	630–770	140
8	840	770–920	150
9	1000	920–1080	160
10	1170	1080–1270	190
11	1370	1270–1480	210
12	1600	1480–1720	240
13	1850	1720–2000	280
14	2150	2000–2320	320
15	2500	2320–2700	380
16	2900	2700–3150	450
17	3400	3150–3700	550
18	4000	3700–4400	700
19	4800	4400–5300	900
20	5800	5300–6400	1100
21	7000	6400–7700	1300
22	8500	7700–9500	1800
23	10500	9500–12000	2500
24	13500	12000–15500	3500
25	19500	15500–22500	7000

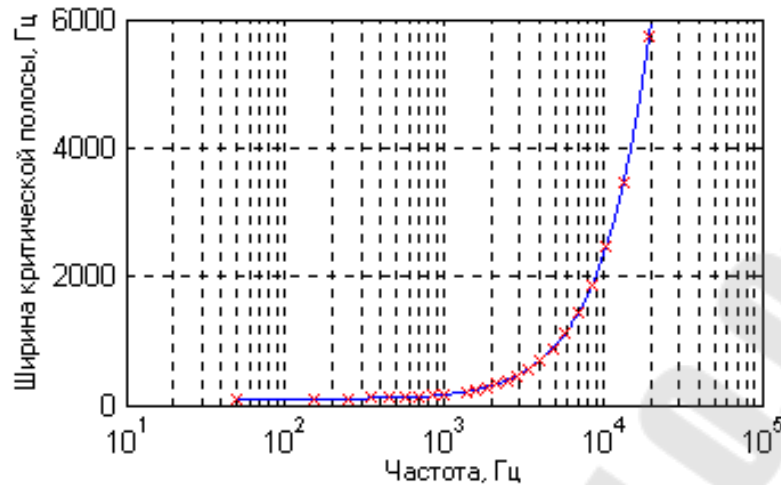


Рисунок 2.21 - Ширина критических полос как функция центральной частоты

Для среднего слушателя ширина критической полосы слуховых фильтров может быть определена из выражения

$$BW(f) = 25 + 75 \left[ 1 + 1,4 \left( f / 1000 \right)^2 \right]^{0,69} .$$

Из рис. 2.21 видно, что по мере увеличения центральной частоты критическая полоса расширяется. Поскольку имеет место однозначная зависимость между частотой синусоидального колебания и высотой тона, то по аналогии с частотной группой слуха было введено понятие тональных групп слуха. Ширина тональной группы не зависит от высоты тона и составляет 100 мел (один мел равен ощущаемой высоте звука частотой 1000 Гц при уровне интенсивности 40 дБ), поэтому оказалось удобным ввести единицу высоты тона в одну тональную группу слуха. В честь немецкого физика Г. Баркгаузена она носит название барк (1 барк = 100 мел), равная ширине критической полосы или расстоянию между центральными частотами соседних критических полос. Переход от одной полосы к другой соответствует изменению высоты в 1 барк.

Функция для перехода от частоты в Гц к частоте в барках (рис. 2.22) определяется соотношением

$$z(f) = 13 \operatorname{arctg}(0,00076 f) + 3,5 \operatorname{arctg} \left[ \left( \frac{f}{7500} \right) \right]^2 ,$$

где  $z(f)$  представляет собой частоту  $f$  в барках.

В основе способов выделения в звуковом сигнале слышимой части информации и удаления из сигнала всех остальных неслышимых деталей лежит эффект маскирования. Он связан с процессом взаимодействия сигналов, приводящим к изменению слуховой чувствительности к маскируемому сигналу в присутствии маскирующего.

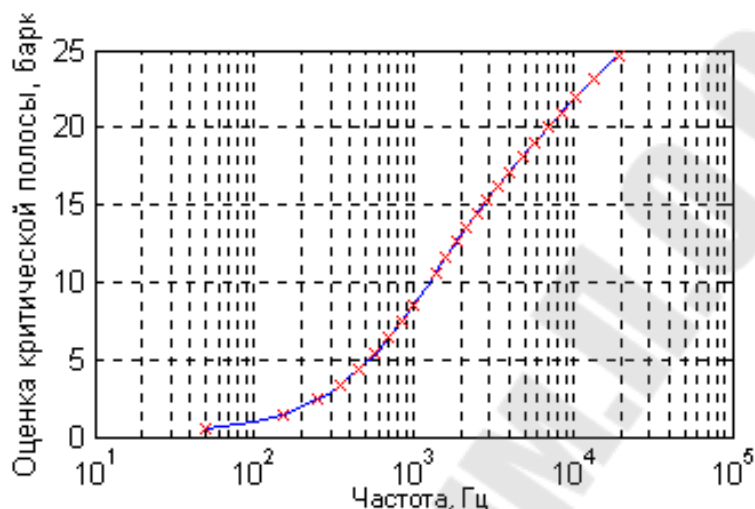


Рисунок 2.22 - Расстояние между центральными частотами критических полос

Эффекты слухового маскирования зависят от спектральных и временных характеристик маскируемого сигнала и сигнала маскирования и могут быть разделены на две основные группы: частотное (одновременное) и временное (неодновременное) маскирования. Маскирование по частоте заключается в следующем: если два сигнала одновременно находятся в ограниченной частотной области, то более слабый сигнал становится неслышимым на фоне более сильного. Маскирование по времени определяет следующий эффект: более слабый сигнал становится неслышимым за 5 – 20 мс до включения сигнала маскирования и становится слышимым через 50 – 200 мс после его включения.

Маскирование непосредственно связано с нелинейностью слуха. Она проявляется в том, что при воздействии на барабанную перепонку достаточно громкого синусоидального звука с частотой  $f$  в слуховом аппарате формируются гармоники этого звука с частотами  $2f$ ,  $3f$  и т.д. Поскольку в первичном воздействующем тоне этих гармоник нет, они получили название субъективных гармоник.

Анализ кривых маскировки (кривых порога слышимости при наличии маскировки) показывает, что маскировка сигналов, имеющих

частоту, лежащую ниже частоты мешающего тона, проявляется значительно слабее чем вышележащих частот. Это дает основание предполагать, что маскирующее действие обусловлено возникновением субъективных гармоник мешающего тона.

Если в качестве мешающего звука использовать узкополосный белый шум, пороги слышимости определяются более четко, а сами кривые получаются более симметричными. Это связано с отсутствием гармонических биений. Большой практический интерес представляют кривые порога слышимости, полученные при маскировке широкополосными шумами.

На рис. 2.23 приведены кривые порогов слышимости синусоидальных звуков, полученные при маскировке белым шумом с различной спектральной плотностью. Из рис. 2.23 видно, что на низких частотах кривые практически не зависят от частоты (примерно до 500 Гц). При дальнейшем увеличении частоты уровень порога слышимости повышается на 3 дБ при каждом удвоении частоты, что примерно соответствует зависимости расширения ширины критических полос с увеличением центральной частоты.

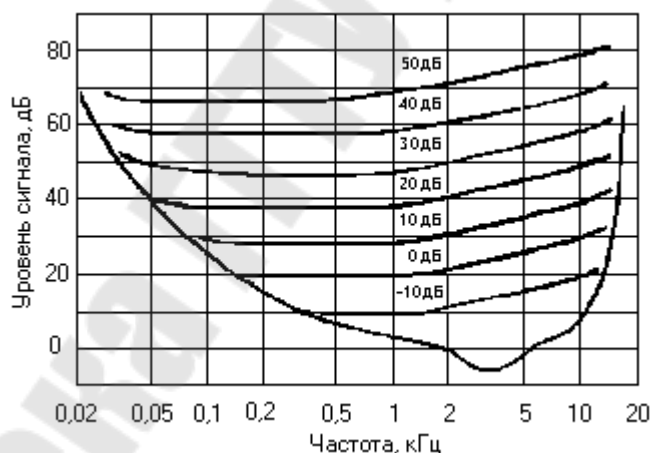


Рисунок 2.23 - Кривые порога слышимости при маскировке белым шумом

Из этого следует, что белый шум неодинаково эффективен для маскировки разных частот. Свойство широкополосных сигналов оказывать максимальное влияние на маскировку сигнала в пределах критических полос положено в основу современных психоакустических алгоритмов сжатия сигналов.

Эффект маскирования в частотной области связан с тем, что в присутствии больших звуковых амплитуд человеческое ухо нечувствительно к малым амплитудам близких частот (рис. 2.24).

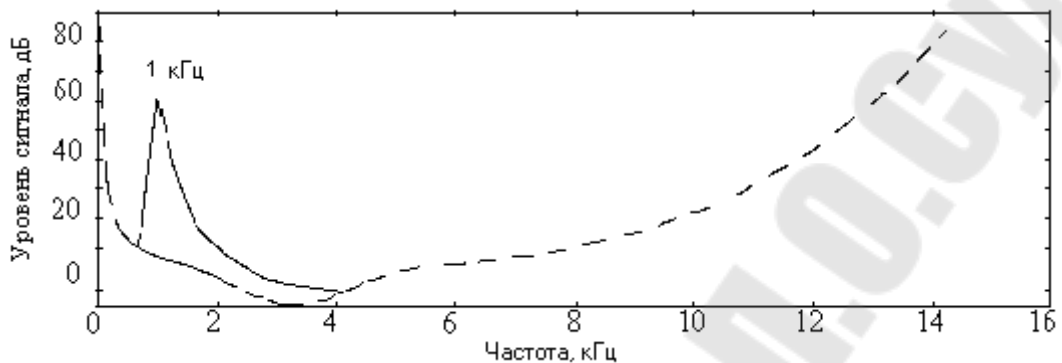


Рисунок 2.24 - Изменение порога чувствительности слуховой системы при маскировании тональным сигналом определенной частоты

Таким образом, для маскирования важнее не абсолютная громкость звуков, а отношение мощности громкого сигнала к мощности тихого. Кроме того, чем ближе частота к частоте маскирующего сигнала, тем сильнее эффект маскирования (рис. 2.25). Степень маскировки — это разность в децибелах между уровнем порога слышимости маскируемого тона в присутствии маскера и его уровнем порога слышимости в тишине. На рис. 2.25 показаны зависимость абсолютного порога чувствительности уха и характер его изменения в присутствии тона с частотой 1 кГц и амплитудой в 60 дБ. Из рис. 2.25 видно, что порог слышимости имеет адаптивный характер: он повышается при появлении каких-либо звуков и понижается в тишине. Для того, чтобы услышать более слабые звуки на фоне более сильных, их частоты должны значительно различаться. Кривая порога чувствительности во многом подобна изменяющейся во времени функции кратковременной спектральной плотности мощности и имеет локальные максимумы в соответствии с тональными компонентами высокого уровня.

Из рис. 2.25 видно, что с увеличением средней частоты маскирующего сигнала диапазон частот, где проявляется маскировка, становится шире. Кривые маскировки являются несимметричными, они имеют крутой спад в сторону низких частот и пологий спад в сторону высоких частот.

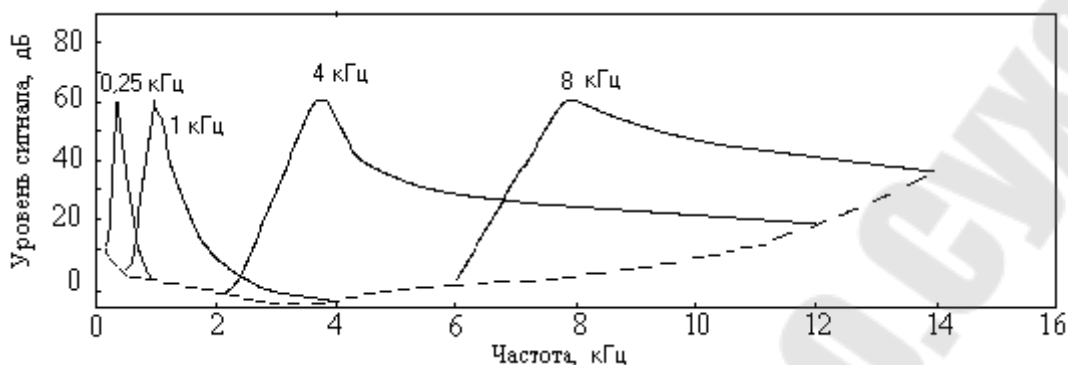


Рисунок 2.25 - Изменение порога чувствительности слуховой системы при маскировании сигналами со средними частотами полос 0,25, 1, 4 и 8 кГц и уровнем интенсивности 60 дБ

Если при построении кривых маскировки по оси абсцисс откладывать не частоты  $f$ , а значения высоты тона  $z$  в барках, то кривые маскировки при разных значениях  $z$  маскирующего сигнала будут одинаковыми при одном и том же значении уровня маскирующего сигнала (рис.2.26). Их форма при таком представлении будет зависеть только от маскирующего сигнала, и не будет зависеть от величины  $z$ .

Маскирование имеет место не только внутри критической полосы, оно распространяется и на соседние полосы. Маскирующий сигнал, центрированный в пределах критической полосы, оказывает определенное воздействие на порог обнаружения в соседних критических полосах. Функция протяженности маскирования, которая учитывает маскирование как внутри, так и между полосами, задается аналитическим выражением

$$SF(x) = 15,81 + 7,5(x + 0,474) - 17,5\sqrt{1 + (x + 0,474)^2},$$

где  $x$  и  $SF(x)$  представлены соответственно в барках и дБ.

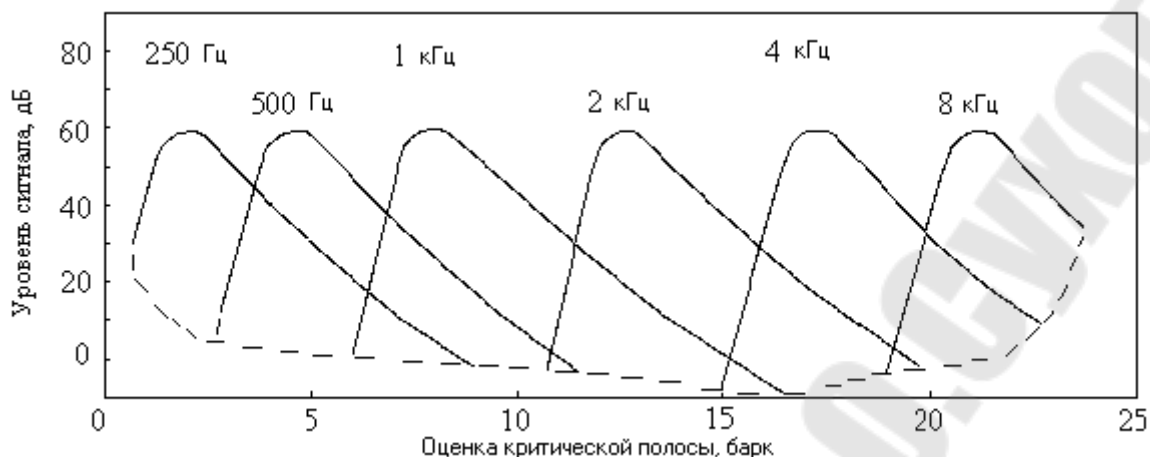


Рисунок 2.26 - Частотное маскирование с частотной шкалой в барках

Вследствие разделения спектра звукового сигнала на частотные полосы слух реагирует не на общую мощность сигнала, а на мощность, сосредоточенную в отдельных частотных группах. При этом более интенсивные частотные группы при определенных условиях могут маскировать менее интенсивные группы. Применительно к сжатию это означает, что в группах меньшей интенсивности допустим больший уровень шума квантования, т.е. допустимо уменьшение числа разрядов при кодировании соответствующих спектральных составляющих, что в свою очередь снижает скорость передаваемого цифрового потока.

Маскирование во временной области (рис. 2.27) характеризует динамические свойства слуха, показывая изменение во времени относительного порога слышимости (порог слышимости одного сигнала в присутствии другого), когда маскирующий и маскируемый сигналы звучат не одновременно.



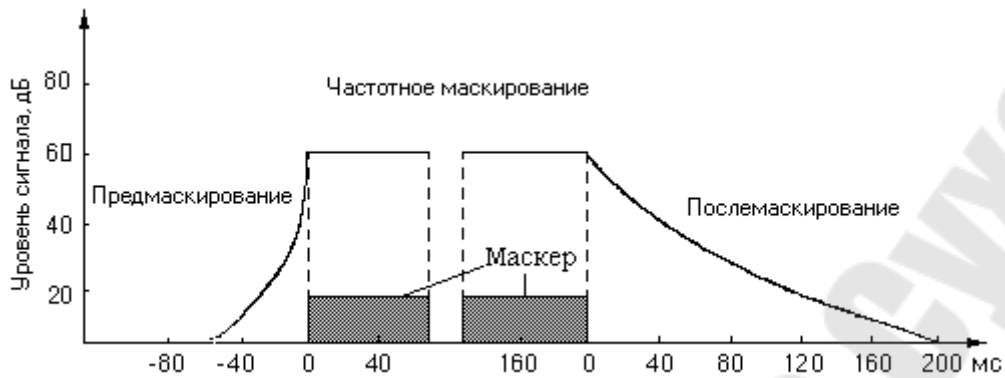


Рисунок 2.27 - Зависимость изменения порога слышимости маскируемого сигнала от величины временного интервала между сигналом и маскером

Различают предмаскировку (маскируемый сигнал перед маскером) и послемаскировку (маскируемый сигнал после маскера). Оба вида маскировки наступают, если маскируемый сигнал и маскер одновременно находятся во временном окне шириной 100–200 мс. Считается, что за это время завершается полная обработка сигнала с выделением всех его признаков. В этом случае слуховую систему можно рассматривать как процессор для обработки звуковых сигналов во временном окне шириной примерно 200 мс.

Из рис. 2.27 видно, что длительность типичных промежутков времени, в пределах которых действуют предмаскирование и послемаскирование, составляет соответственно 5 – 50 мс и 50 – 200 мс. Предмаскировка проявляется на значительно более коротком временном интервале, причем ее длительность в очень сильной степени зависит от опыта слушателя. Степень маскирования звукового сигнала зависит от интервала между маскируемым сигналом и маскером, уровня интенсивности и длительности воздействия маскера. Сближение во времени подачи сигнала и маскера увеличивает маскировку. Временное маскирование зависит от частотного взаимоотношения сигнала и маскера точно так же, как и при частотном маскировании. Оно проявится в большей степени, если сигнал и маскер близки по частоте.

### 2.13. Сжатие аудиосигнала на основе стандарта MP3

Международная организация по стандартизации (International Organization for Standardization - ISO) и экспертная группа по вопросам движущего изображения (Motion Picture Experts Group - MPEG) разработали стандарт аудиосжатия для сигнала, синхронизированного с сжатым видеосигналом, известный как MPEG. В стандарте кодирования аудиосигналов MPEG-1 использованы три уровня (Layer 1, 2 и 3) увеличивающей сложности и улучшающей субъективной производительности, частоты дискретизации 32, 44,1 и 48 кГц, а биты на выход подаются со скоростью от 32 до 192 Кбит/с (монофонический канал) или со скоростью от 64 до 384 Кбит/с (стереофонический канал). При данных высоких значениях скорости потока двоичных данных каждый из уровней стандарта MPEG-1 обеспечивает приблизительно одинаковое качество звучания, близкое к качеству компакт-диска. Система сжатия звука MPEG-1 Layer 3 (MP3) позволяет регулировать коэффициент сжатия и искать оптимальное соотношение качество звучания/коэффициент сжатия.

Перцептивный кодер, используемый в стандарте MPEG, является примером эффективного сжатия аудиоинформации с учетом особенностей слуховой системы человека. На рисунке представлена структурная схема перцептивного кодера на основе MP3. Из рис. 2.28 видно, что цифровой 16-битовый аудиосигнал разделяется на  $M = 32$  частотные полосы с помощью группы равномерно расположенных полосовых фильтров. Все фильтры в группе фильтров получают путем модулирования фильтра прототипа, предназначенного для увеличения частотного разрешения аудиосигнала и лучшей аппроксимации критических полос слуховой системы. Импульсный отклик фильтра  $i$ -й полосы на единичный импульс  $h_i(k)$  можно представить в виде произведения коэффициентов  $h(k)$  фильтра прототипа, являющихся квантованными значениями импульсной характеристики этого фильтра, и модулирующей функции, которая сдвигает низкочастотный отклик в  $i$ -ю частотную полосу,

$$h_i(k) = h(k) \cos\left(\frac{2i-1}{2M} + \phi(i)\right),$$

где  $M$  – число полос, на которые разделяется звуковой сигнал  $x(t)$ ;  $i=1,2,\dots, 32$ ;  $k$  – номер коэффициента импульсной характеристики  $h(k)$  фильтра прототипа ( $k=1,2,\dots,512$ ),  $\phi(i)$  - фазовый сдвиг для  $i$ -й полосы.

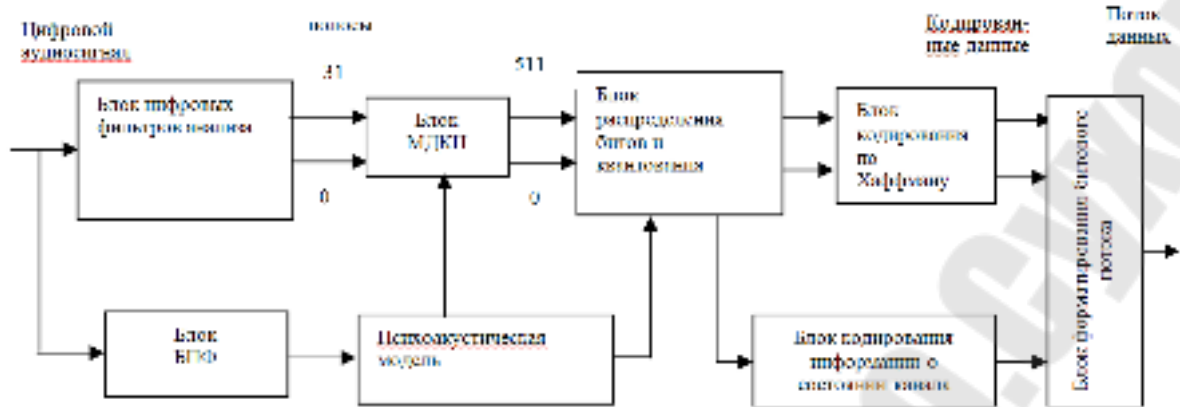


Рисунок 2.28 - Структурная схема перцептивного кодера MP3

Сигнал на выходе фильтра  $i$ -й полосы в дискретный момент времени  $n$  представляется в виде дискретной свертки

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

где  $*$  – символ свертки;  $h_i(k)$  – массив коэффициентов  $i$ -го полосового фильтра порядка  $N=512$ ;  $x(n-k)$  – входной массив цифровых данных фильтра. Число  $N$  определяет эффективность фильтра, реализующего  $N$  операций умножения с накоплением для каждого выходного отсчета  $y_i(n)$ . Подход к кодированию, основанный на использовании набора цифровых фильтров, дает высокое временное разрешение и таким образом, допускает более точное моделирование эффекта маскирования. Полосы нормализуются с помощью масштабирующего фактора так, что максимальное значение амплитуды отсчета в каждой полосе равно единице. Повышение точности распознавания маскирующих сигналов достигается посредством дальнейшей обработки 32 полосовых сигналов с помощью перекрывающегося модифицированного дискретно косинусного преобразования (МДКП), предназначенного для анализа соседних областей частотных полос. Это преобразование задается выражением

$$Y(n) = \sum_{k=0}^{N-1} h(k) y(k) \cos\left(\frac{\pi}{2N} (2n+1)(2k+1 + \frac{N}{2})\right),$$

где  $n$  и  $k$  – индексы спектральной компоненты  $Y(k)$  и дискретного отсчета  $y(k)$  аудиосигнала соответственно;  $n=0,1, \dots, \frac{N}{2}-1$ ,  $h(k)$  – импульсный отклик окна анализа сигнала  $x(k)$ , обладающий свойствами фильтра нижних частот.

МДКП основано на идеи устранения наложения во временной области путем использования пересекающихся временных окон анализа полосовых сигналов. Наложение во временной области, вводимое с помощью 50% перекрытия окон, устраняется при обратном преобразовании МДКП. Для более точного восстановления аудиосигнала весовые коэффициенты  $h(k)$  функции анализирующего окна должны удовлетворять соотношениям

$$\begin{aligned} h^2\left(\frac{N}{2}-1-k\right)+h^2(k) &= 2, & 0 \leq k \leq N/2, \\ h^2\left(\frac{N}{2}+k\right)+h^2(N-1-k) &= 2, & 0 \leq k \leq N/2. \end{aligned}$$

С практической точки зрения ни сигналы, ни реализуемые узкополосные фильтры не имеют строго ограниченной полосы. Следовательно, всегда имеет место некоторое перекрытие соседних частотных полос, что приведет к неизбежным искажениям при восстановлении непрерывного сигнала. Для устранения нежелательного временного эффекта наложения, называемого *aliasing* (явление, являющееся результатом недостаточной частоты выборки и сопровождаемое появлением ложных частот), окна анализа задаются выражением

$$h(k) = \pm\sqrt{2} \sin\left[\left(k + \frac{1}{2}\right)\frac{\pi}{N}\right].$$

В перцептивном кодере часто используется синусоидальное окно  $h(k) = \sin\left(\pi\left(\frac{k}{N}\right)\right)$ , также удовлетворяющее требованиям совершенного восстановления непрерывного аудиосигнала.

Кодер MP3 реализует метод адаптивной обработки значений МДКП, позволяющий устранить некоторые артефакты, вызванные перекрывающимися полосами группы фильтров анализа и более эффективно использовать психоакустические свойства слуховой системы.

Кодер использует два блока МДКП различной длины: длинный блок с 18 выборками и короткий блок с 6 выборками. Поскольку имеет место 50% перекрытие между последовательными окнами МДКП, то ширины окон преобразования достигают длительностей 36 и 12 выборок для длинного и короткого блоков соответственно

На рис. 2.29 приведена структурная схема блока МДКП перцептивного кодера МР3. Для изменения длины МДКП используется набор оконных функций: синусное окно типа short для длинного блока, синусное окно short для короткого блока и два окна перехода start и stop, необходимые для уменьшения искажений, возникающих при переходе от длинных к коротким окнам и наоборот. Решение о выборе длины МДКП принимает психоакустическая модель кодера. Из рис. 2.29 видно, что выходные полосовые сигналы обрабатываются с помощью МДКП в двух различных по длине блоках: коротком (16 - точечное МДКП) и длинном (36- точечное МДКП). Короткий блок улучшает временное разрешение кодера, что позволяет эффективно использовать предмаскирование для скрывания различных артефактов, возникающих при кодировании. Длинный блок позволяет

реализовать более высокое частотное разрешение для квазистационарных сигналов, что сокращает дополнительную информацию о состоянии канала и уменьшает скорость аудио потока.



Рисунок 2.29 - Структурная схема блока МДКП перцептивного кодера MP3

Для заданного кадра аудио выборок МДКП может иметь все блоки одинаковой длины (длинные или короткие) или иметь режим смешанных блоков. Применение 18-точечного МДКП к выходным отсчетам одной из 32 полос анализирующего блока фильтров позволяет улучшить частотное разрешение, приблизив его к критическим полосам. Максимальное число компонент в кодере равно  $32 \cdot 18 = 576$ , а частотное разрешение составляет  $24000/576$  или 41,67 Гц.

Для спектральной декомпозиции аудиосигнала и достижения достаточно высокого спектрального разрешения при вычислении маскирующих порогов параллельно с разбиением аудиосигнала на полосы вычисляется 1024 - точечное БПФ. Психоакустическая модель использует результаты БПФ для разделения полосовых данных на длинные и короткие блоки в зависимости от того, какое требуется частотное или временное разрешение, и обнаружения тональных и нетональных компонентов спектра. Значение тональных компонентов

вычисляется путем нахождения локальных максимумов спектра. Маскирующие эффекты, обусловленные тональными и нетональными компонентами и порогом слышимости, используются при вычислении минимального маскирующего порога для каждой полосы. Далее следует итеративная процедура распределения разрядов, которая использует вычисленные маскирующие пороги для выбора оптимального квантователя для каждой частотной полосы. Квантователи выбираются таким образом, чтобы одновременно удовлетворить требованиям маскирования и заданного битрейта. Сущность разрядного (битового) распределения состоит в том, что часто определенные фрагменты аудиосигнала не могут быть закодированы в рамках данного разрядного диапазона без потерь качества. В таком случае MP3 использует небольшой запас разрядов, имеющихся в буфере, для кодирования менее сложных фрагментов в меньший разрядовый диапазон. Данный режим работы предназначен для кодирования блоков отсчетов входного сигнала, содержащих легко сжимаемые сигналы, минимально необходимым числом битов, а оставшие разряды добавляются в блоки со сложным, трудно компрессируемым сигналом.

Классический алгоритм Хаффмана в MP3 используется на последнем этапе сжатия. Это так называемое энтропийное кодирование, учитывающее статистические особенности звукового сигнала. Кодирование Хаффмана хорошо дополняет перцептивные методы сжатия. Перцептивное кодирование особенно эффективно при кодировании сложного полифонического звучания в силу того, что многие звуки маскируются и их влияние на общую картину снижается. При кодировании же чистых звуков оказывается эффективным алгоритм Хаффмана из-за того, что оцифрованный звук содержит много избыточных байтов, которые при кодировании заменяются более короткими кодами. Квантованные данные (значения коэффициентов МДКП) вместе с информацией о распределении битов, квантованных масштабирующих коэффициентах и выбранном квантователе передаются в блок временного уплотнения (мультиплексор) для передачи потока данных по каналу связи.

На рис 2.30 приведена упрощенная структурная схема перцептивного декодера, восстанавливающего аудиоданные с блочной структурой из потока данных на выходе кодера.



Рисунок 2.30 - Структурная схема перцептивного декодера MP3

Обратное модифицированное дискретно-косинусное преобразование (ОМДКП) задается выражением

$$y(m) = h(m) \frac{4}{\pi} \sum_{n=0}^{\frac{N}{2}-1} Y(n) \cos\left(\frac{\pi}{2N} (2n+1) \left(2m+1 + \frac{N}{2}\right)\right),$$

где  $m=0, 1, \dots, N-1$ .

Обозначение  $y$  в последнем преобразовании подчеркивает тот факт, что последовательность данных, восстановленная с помощью МДКП, не соответствует исходной последовательности данных. Блок цифровых фильтров синтеза осуществляет обратное преобразование полос аудиосигнала, первоначально формируемых блоком цифровых фильтров анализа, в аудиосигнал с блочной структурой.

### 3. ОБРАБОТКА ВИДЕОИНФОРМАЦИИ В СИСТЕМАХ ТЕЛЕКОММУНИКАЦИЙ

Многие отрасли техники, имеющие отношение к получению, обработке, хранению и передаче информации, в значительной степени



ориентируются в настоящее время на развитие систем, в которых информация имеет характер изображений.

Цифровое изображение представляет собой прямоугольную таблицу точек, или элементов изображения, расположенных в  $m$  строках и  $n$  столбцах. Выражение  $m \times n$  называется *разрешением* изображения (хотя иногда этот термин используется для обозначения числа пикселей, приходящихся на единицу длины изображения). Точки изображения называются *пикселями* (за исключением случаев, когда изображение передается факсом или видео; в этих случаях точка называется *пелом*). Для целей сжатия графических образов удобно выделить следующие типы изображений:

1. *Двухуровневое* (или монохроматическое) изображение. В этом случае все пиксели могут иметь только два значения, которые обычно называют черным (двоичная единица, или основной цвет) и белым (двоичный нуль или цвет фона). Каждый пиксел такого изображения представлен одним битом, поэтому это самый простой тип изображения.

2. *Полутоновое* изображение. Каждый пиксел такого изображения может иметь  $2^n$  значений от 0 до  $2^n - 1$ , обозначающих одну из  $2^n$  градаций серого (или иного) цвета. Число  $n$  обычно сравнимо с размером байта, то есть, оно равно 4, 8, 12, 16, 24 или другое кратное 4 или 8. Множество самых значимых битов всех пикселов образуют самую значимую битовую плоскость или слой изображения. Итак, полутоновое изображение со шкалой из  $2^n$  уровней составлено из  $n$  битовых слоев.

3. *Цветное изображение*. Существует несколько методов задания цвета, но в каждом из них участвуют три параметра. Следовательно, цветной пиксел состоит из трех частей. Обычно, цветной пиксел состоит из трех байтов. Типичными цветовыми моделями являются RGB, HLS и CMYK.

4. *Изображение с непрерывным тоном*. Этот тип изображений может иметь много похожих цветов (или полутонов). Когда соседние пиксели отличаются всего на единицу, глазу практически невозможно различить их цвета. В результате такие изображения могут содержать области, в которых цвет кажется глазу непрерывно меняющимся. В этом случае пиксел представляется или большим числом (в полутоновом случае) или тремя компонентами (в случае цветного образа). Изображения с непрерывным тоном являются природными или естественными (в отличие от рукотворных, искусственных); обычно

они получаются при съемке на цифровую фотокамеру или при сканировании фотографий или рисунков.

5. *Дискретно-тоновое* изображение (оно еще называется синтетическим). Обычно, это изображение получается искусственным путем. В нем может быть всего несколько цветов или много цветов, но в нем нет шумов и пятен естественного изображения. Примерами таких изображений могут служить фотографии искусственных объектов, машин или механизмов, страницы текста, карты, рисунки или изображения на дисплее компьютера. (Не каждое искусственное изображение будет обязательно дискретно-тоновым. Сгенерированное компьютером изображение, которое должно выглядеть натуральным, будет иметь непрерывные тона, несмотря на свое искусственное происхождение.) Искусственные объекты, тексты, нарисованные линии имеют форму, хорошо определяемые границы. Они сильно контрастируют на фоне остальной части изображения (фона). Прилегающие пикселы дискретно-тонового образа часто бывают одиночными или сильно меняют свои значения. Такие изображения плохо сжимаются методами с потерей данных, поскольку искажение всего нескольких пикселей буквы делает ее неразборчивой, преобразует привычное начертание в совершенно неразличимое. Методы сжатия изображений с непрерывными тонами плохо обращаются с четкими краями дискретно-тоновых образов, для которых следует разрабатывать особые методы компрессии. Отметим, что дискретно-тоновые изображения, обычно, несут в себе большую избыточность. Многие ее фрагменты повторяются много раз в разных местах изображения.

6. Изображения, *подобные мультфильмам*. Это цветные изображения, в которых присутствуют большие области одного цвета. При этом соприкасающиеся области могут весьма различаться по своему цвету. Это свойство можно использовать для достижения лучшей компрессии.

Интуитивно становится ясно, что каждому типу изображений присуща определенная избыточность, но все они избыточны по-разному. Поэтому трудно создать один метод, который одинаково хорошо сжимает любые типы изображений. Существуют отдельные методы для сжатия двухуровневых образов, непрерывно-тоновых и дискретно-тоновых изображений. Существуют также методы, которые пытаются разделить изображение на непрерывно-тоновую и дискретно-тоновую части и сжимать их по отдельности.

### 3.1. Дискретизация непрерывных изображений

Очень редко изображения, получаемые в информационных системах, имеют цифровую форму. Поэтому их преобразование к этому виду является обязательной операцией, если предполагается использовать цифровую обработку, передачу, хранение. Как и при одномерных сигналах, данное преобразование включает в себя две процедуры. Первая состоит в замене непрерывного кадра дискретным и обычно называется *дискретизацией*, а вторая выполняет замену непрерывного множества значений яркости множеством квантованных значений и носит название *квантования*. При цифровом представлении каждому из квантованных значений яркости ставится в соответствие двоичное число, чем и достигается возможность ввода изображения в ЭВМ.

Двумерный характер изображения по сравнению с обычными сигналами содержит дополнительные возможности оптимизации цифрового представления с целью сокращения объема получаемых цифровых данных. В связи с этим изучался вопрос о наилучшем размещении уровней квантования, а также об использовании различных растров, другие аспекты данной задачи. Следует, однако, сказать, что в подавляющем большинстве случаев на практике применяют дискретизацию, основанную на использовании прямоугольного раstra, и равномерное квантование яркости. Это связано с простотой выполнения соответствующих операций и относительно небольшими преимуществами от использования оптимальных преобразований. При использовании прямоугольного раstra в окончательном виде цифровое изображение обычно представляет собой матрицу, строки и столбцы которой соответствуют строкам и столбцам изображения.

Замену непрерывного изображения дискретным можно выполнить различными способами. Можно, например, выбрать какую-либо систему ортогональных функций и, вычислив коэффициенты представления изображения по этой системе (по этому базису), заменить ими изображение. Многообразие базисов дает возможность образования различных дискретных представлений непрерывного изображения. Однако наиболее употребительной является периодическая дискретизация, в частности, как упоминалось выше, дискретизация с прямоугольным растром. Такой способ дискретизации может рассматриваться как один из вариантов применения ортогонального базиса, использующего в качестве своих элементов сдвинутые  $\delta$ -

функции. Далее, следуя, в основном, подробно рассмотрим основные особенности прямоугольной дискретизации.

Пусть  $x_H(t_1, t_2)$  - непрерывное изображение, а  $x(i_1, i_2)$  - соответствующее ему дискретное, полученное из непрерывного путем прямоугольной дискретизации. Это означает, что связь между ними определяется выражением:

$$x(i_1, i_2) = x(i_1 \Delta t_1, i_2 \Delta t_2), \quad (3.1)$$

где  $\Delta t_1, \Delta t_2$  - соответственно вертикальный и горизонтальный шаги или интервалы дискретизации. Рис. 3.1 иллюстрирует расположение отсчетов на плоскости  $(t_1, t_2)$  при прямоугольной дискретизации.

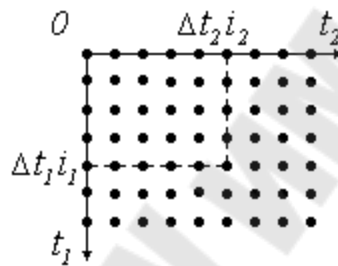


Рисунок 3.1 - Расположение отсчетов при прямоугольной дискретизации

Основной вопрос, который возникает при замене непрерывного изображения дискретным, состоит в определении условий, при которых такая замена является полноценной, т.е. не сопровождается потерей информации, содержащейся в непрерывном сигнале. Потери отсутствуют, если, располагая дискретным сигналом, можно восстановить непрерывный. С математической точки зрения вопрос, таким образом, заключается в восстановлении непрерывного сигнала в двумерных промежутках между узлами, в которых его значения известны или, иными словами, в осуществлении двумерной интерполяции. Ответить на этот вопрос можно, анализируя спектральные свойства непрерывного и дискретного изображений.

Двумерный непрерывный частотный спектр  $X_H(\Omega_1, \Omega_2)$  непрерывного сигнала  $x_H(t_1, t_2)$  определяется двумерным прямым преобразованием Фурье:

$$X_H(\Omega_1, \Omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t_1, t_2) \exp(-j\Omega_1 t_1 - j\Omega_2 t_2) dt_1 dt_2, \quad (3.2)$$

которому отвечает двумерное обратное непрерывное преобразование Фурье:

$$x(t_1, t_2) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(\Omega_1, \Omega_2) \exp(j\Omega_1 t_1 + j\Omega_2 t_2) d\Omega_1 d\Omega_2. \quad (3.3)$$

Последнее соотношение верно при любых значениях  $t_1, t_2$ , в том числе и в узлах прямоугольной решетки  $t_1 = i_1 \Delta t_1, t_2 = i_2 \Delta t_2$ . Поэтому для значений сигнала в узлах, учитывая (3.1), соотношение (3.3) можно записать в виде:

$$x(i_1, i_2) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(\Omega_1, \Omega_2) \exp(j\Omega_1 i_1 \Delta t_1 + j\Omega_2 i_2 \Delta t_2) d\Omega_1 d\Omega_2. \quad (3.4)$$

Обозначим для краткости через  $S(k_1, k_2)$  прямоугольный участок в двумерной частотной области

$$-\frac{\pi + 2\pi k_1}{\Delta t_1} \leq \Omega_1 \leq \frac{\pi + 2\pi k_1}{\Delta t_1}, \quad -\frac{\pi + 2\pi k_2}{\Delta t_2} \leq \Omega_2 \leq \frac{\pi + 2\pi k_2}{\Delta t_2}.$$

Вычисление интеграла в (3.4) по всей частотной области можно заменить интегрированием по отдельным участкам  $S(k_1, k_2)$  и суммированием результатов:

$$x(i_1, i_2) = \frac{1}{4\pi^2} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \iint_{S(k_1, k_2)} X(\Omega_1, \Omega_2) \times \\ \times \exp(j\Omega_1 \Delta t_1 i_1 + j\Omega_2 \Delta t_2 i_2) d\Omega_1 d\Omega_2.$$

Выполняя замену переменных по правилу  $\Omega_2 \rightarrow \Omega_2 - 2\pi k_2 / \Delta t_2$ , добиваемся независимости области интегрирования от номеров  $k_1$  и  $k_2$ :

$$x(i_1, i_2) = \frac{1}{4\pi^2} \int_{-\frac{\pi}{\Delta t_1}}^{\frac{\pi}{\Delta t_1}} \int_{-\frac{\pi}{\Delta t_2}}^{\frac{\pi}{\Delta t_2}} \sum_{k_1} \sum_{k_2} X(\Omega_1 + \frac{2\pi k_1}{\Delta t_1}, \Omega_2 + \frac{2\pi k_2}{\Delta t_2}) \exp(j\Omega_1 \Delta t_1 i_1 + j\Omega_2 \Delta t_2 i_2) d\Omega_1 d\Omega_2.$$

Здесь учтено, что  $\exp(-j2\pi ki) = 1$  при любых целых значениях  $k$  и  $i$ . Данное выражение по своей форме очень близко к обратному преобразованию Фурье. Отличие состоит лишь в неправильном виде экспоненциального множителя. Для придания ему необходимого вида введем нормированные частоты  $\omega_1 = \Omega_1 \Delta t_1, \omega_2 = \Omega_2 \Delta t_2$  и выполним в соответствии с этим замену переменных. В результате получим:

$$x(i_1, i_2) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \frac{1}{\Delta t_1 \Delta t_2} \sum_{k_1} \sum_{k_2} X\left(\frac{\omega_1 + 2\pi k_1}{\Delta t_1}, \frac{\omega_2 + 2\pi k_2}{\Delta t_2}\right) \times \exp(j\omega_1 i_1 + j\omega_2 i_2) d\omega_1 d\omega_2. \quad (3.5)$$

Теперь выражение (3.5) имеет форму обратного преобразования Фурье, следовательно стоящая под знаком интеграла функция

$$X(\omega_1, \omega_2) = \frac{1}{\Delta t_1 \Delta t_2} \sum_{k_1} \sum_{k_2} X\left(\frac{\omega_1 + 2\pi k_1}{\Delta t_1}, \frac{\omega_2 + 2\pi k_2}{\Delta t_2}\right) \quad (3.6)$$

является двумерным спектром дискретного изображения. В плоскости ненормированных частот выражение (9.6) имеет вид:

$$X(\Omega_1 \Delta t_1, \Omega_2 \Delta t_2) = \frac{1}{\Delta t_1 \Delta t_2} \sum_{k_1} \sum_{k_2} X\left(\Omega_1 + \frac{2\pi k_1}{\Delta t_1}, \Omega_2 + \frac{2\pi k_2}{\Delta t_2}\right) \quad (3.7)$$

Из (3.7) следует, что двумерный спектр дискретного изображения является прямоугольно периодическим с периодами  $2\pi/\Delta t_1$  и  $2\pi/\Delta t_2$  по осям частот  $\Omega_1$  и  $\Omega_2$  соответственно. Спектр дискретного изображения  $X(\Omega_1, \Omega_2)$  образуется в результате суммирования бесконечного количества спектров  $X_H(\Omega_1, \Omega_2)$  непрерывного изображения, отличающихся друг от друга частотными сдвигами  $2\pi/\Delta t_1$  и  $2\pi/\Delta t_2$ . Рис.3.2 качественно показывает соотношение между двумерными спектрами непрерывного (рис.3.2.а) и дискретного (рис.3.2.б) изображений.

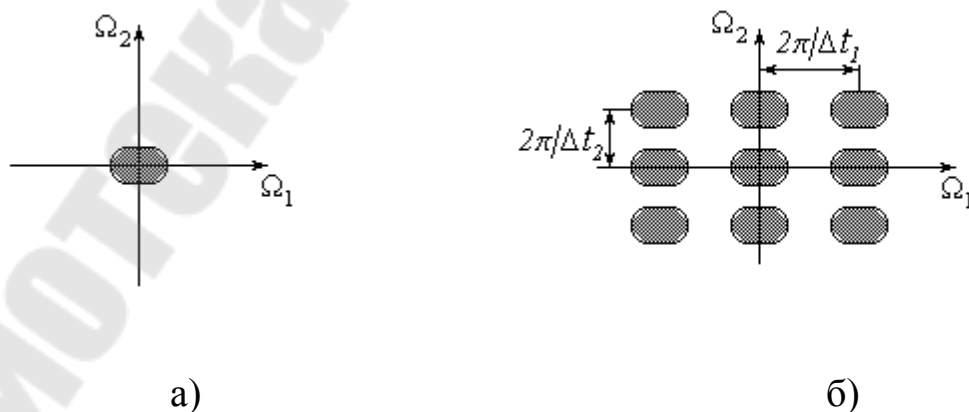


Рисунок 3.2 - Частотные спектры непрерывного и дискретного изображений

Сам результат суммирования существенно зависит от значений этих частотных сдвигов, или, иными словами, от выбора интервалов

дискретизации  $\Delta t_1, \Delta t_2$ . Допустим, что спектр непрерывного изображения  $X_H(\Omega_1, \Omega_2)$  отличен от нуля в некоторой двумерной области в окрестности нулевой частоты, т. е. описывается двумерной финитной функцией. Если при этом интервалы дискретизации выбраны так, что  $X_H(\Omega_1, \Omega_2) = 0$  при  $|\Omega_1| \geq \pi/\Delta t_1$ ,  $|\Omega_2| \geq \pi/\Delta t_2$ , то наложения отдельных ветвей при формировании суммы (9.7) происходить не будет. Следовательно, в пределах каждого прямоугольного участка  $S(k_1, k_2)$  от нуля будет отличаться лишь одно слагаемое. В частности, при  $k_1 = 0, k_2 = 0$  имеем:

$$X(\Omega_1, \Omega_2) = \frac{1}{\Delta t_1 \Delta t_2} X_H(\Omega_1, \Omega_2) \text{ при } (|\Omega_1| < \pi/\Delta t_1), |\Omega_2| < \pi/\Delta t_2. \quad (3.8)$$

Таким образом, в пределах частотной области  $S(0,0)$  спектры непрерывного и дискретного изображений с точностью до постоянного множителя совпадают. При этом спектр дискретного изображения в этой частотной области содержит полную информацию о спектре непрерывного изображения. Подчеркнем, что данное совпадение имеет место лишь при оговоренных условиях, определяемых удачным выбором интервалов дискретизации. Отметим, что выполнение этих условий, согласно (9.8), достигается при достаточно малых значениях интервалов дискретизации  $\Delta t_1, \Delta t_2$ , которые должны удовлетворять требованиям:

$$\Delta t_1 \leq \pi/\Omega_{1\text{гр}}, \quad \Delta t_2 \leq \pi/\Omega_{2\text{гр}}, \quad (3.9)$$

в которых  $\Omega_{1\text{гр}}, \Omega_{2\text{гр}}$  - граничные частоты двумерного спектра.

Соотношение (3.8) определяет способ получения непрерывного изображения  $x_H(t_1, t_2)$  из дискретного  $x(i_1, i_2)$ . Для этого достаточно выполнить двумерную фильтрацию дискретного изображения низкочастотным фильтром с частотной характеристикой

$$K(j\Omega_1, j\Omega_2) = \begin{cases} \Delta t_1 \Delta t_2 & \text{при } |\Omega_1| \leq \pi/\Delta t_1, |\Omega_2| \leq \pi/\Delta t_2, \\ 0 & \text{при других } \Omega_1, \Omega_2. \end{cases} \quad (3.10)$$

Спектр изображения на его выходе содержит ненулевые компоненты лишь в частотной области  $S(0,0)$  и равняется, согласно (3.8), спектру непрерывного изображения  $X_H(\Omega_1, \Omega_2)$ . Это означает, что изображение на выходе идеального фильтра низких частот совпадает с  $x_H(t_1, t_2)$ .

Таким образом, идеальное интерполяционное восстановление непрерывного изображения выполняется при помощи двумерного

фильтра с прямоугольной частотной характеристикой (3.10). Нетрудно записать в явном виде алгоритм восстановления непрерывного изображения. Двумерная импульсная характеристика восстанавливающего фильтра, которую легко получить при помощи обратного преобразования Фурье от (3.10), имеет вид:

$$h(t_1, t_2) = \frac{\sin(\pi t_1 / \Delta t_1)}{\pi t_1 / \Delta t_1} \frac{\sin(\pi t_2 / \Delta t_2)}{\pi t_2 / \Delta t_2}.$$

Продукт фильтрации может быть определен при помощи двумерной свертки входного изображения и данной импульсной характеристики. Представив входное изображение  $x_{\text{вх}}(t_1, t_2)$  в виде двумерной последовательности  $\delta$ -функций

$$x_{\text{вх}}(t_1, t_2) = \sum_{i_1} \sum_{i_2} x(i_1, i_2) \cdot \delta(t_1 - i_1 \Delta t_1) \delta(t_2 - i_2 \Delta t_2),$$

после выполнения свертки находим:

$$\begin{aligned} x(t_1, t_2) &= \sum_{i_1} \sum_{i_2} x(i_1, i_2) \cdot \\ &\frac{\sin[\pi(t_1 - i_1 \Delta t_1) / \Delta t_1]}{\pi(t_1 - i_1 \Delta t_1) / \Delta t_1} \frac{\sin[\pi(t_2 - i_2 \Delta t_2) / \Delta t_2]}{\pi(t_2 - i_2 \Delta t_2) / \Delta t_2} \end{aligned} \quad (3.11)$$

Полученное соотношение указывает способ точного интерполяционного восстановления непрерывного изображения по известной последовательности его двумерных отсчетов. Согласно этому выражению для точного восстановления в роли интерполирующих функций должны использоваться двумерные функции вида  $\sin x/x$ . Соотношение (3.11) представляет собой двумерный вариант теоремы Котельникова-Найквиста.

Подчеркнем еще раз, что эти результаты справедливы, если двумерный спектр сигнала является финитным, а интервалы дискретизации достаточно малы. Справедливость сделанных выводов нарушается, если хотя бы одно из этих условий не выполняется. Реальные изображения редко имеют спектры с ярко выраженными граничными частотами. Одной из причин, приводящих к неограниченности спектра, является ограниченность размеров изображения. Из-за этого при суммировании в (3.7) в каждой из зон  $S(k_1, k_2)$  проявляется действие слагаемых из соседних спектральных зон. При этом точное восстановление непрерывного изображения становится вообще невозможным. В частности, не приводит к точному восстановлению и использование фильтра с прямоугольной частотной характеристикой.

Особенностью оптимального восстановления изображения в промежутках между отсчетами является использование всех отсчетов дис-



кретного изображения, как это предписывается процедурой (3.11). Это не всегда удобно, часто требуется восстанавливать сигнал в локальной области, опираясь на некоторое небольшое количество имеющихся дискретных значений. В этих случаях целесообразно применять квазиоптимальное восстановление при помощи различных интерполирующих функций. Такого рода задача возникает, например, при решении проблемы привязки двух изображений, когда из-за геометрических расстроек этих изображений имеющиеся отсчеты одного из них могут соответствовать некоторым точкам, находящимся в промежутках между узлами другого. Решение этой задачи более подробно обсуждается в последующих разделах данного пособия.

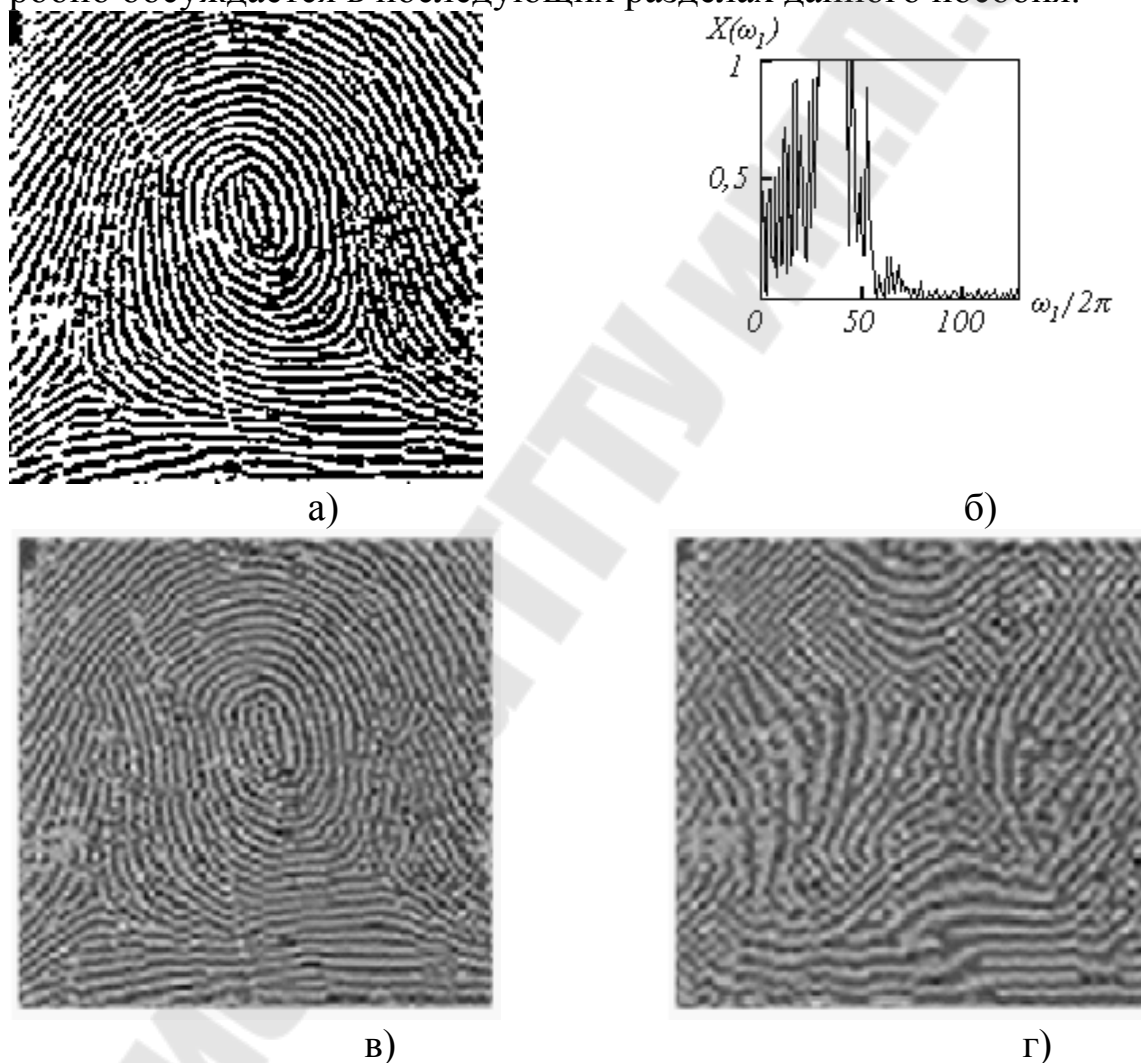


Рисунок 3.3 - Влияние интервала дискретизации на восстановление изображения «Отпечаток пальца»

Рис. 3.3 иллюстрирует влияние интервалов дискретизации на восстановление изображений. Исходное изображение, представляю-

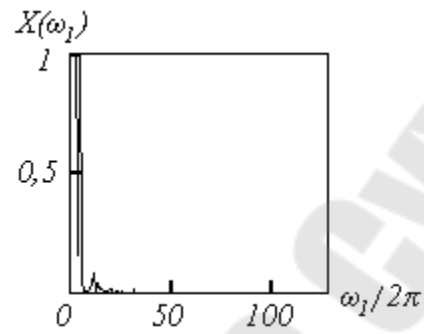
щее собой отпечаток пальца, приведено на рис. 3.3.а, а одно из сечений его нормированного спектра - на рис. 3.3.б. Данное изображение является дискретным, а в качестве граничной частоты использовано значение  $\omega_{1гр} = 2\pi \cdot 128$ . Как следует из рис. 3.3.б, значение спектра на этой частоте пренебрежимо мало, что гарантирует качественное восстановление. По сути дела, наблюдаемая на рис. 3.3.а картина и является результатом восстановления непрерывного изображения, а роль восстанавливающего фильтра выполняет устройство визуализации - монитор или принтер. В этом смысле изображение рис. 3.3.а может рассматриваться как непрерывное.

Рис. 3.3.в,г показывают последствия от неправильного выбора интервалов дискретизации. При их получении осуществлялась “дискретизация непрерывного” изображения рис. 3.3.а путем прореживания его отсчетов. Рис. 3.3.в соответствует увеличению шага дискретизации по каждой координате в три, а рис. 3.3.г - в четыре раза. Это было бы допустимо, если бы значения граничных частот были ниже в такое же число раз. В действительности, как видно из рис.3.3.б, происходит нарушение требований (3.9), особенно грубое при четырехкратном прореживании отсчетов. Поэтому восстановленные при помощи алгоритма (3.11) изображения оказываются не только расфокусированными, но и сильно искажают текстуру отпечатка.

На рис. 3.4 приведена аналогичная серия результатов, полученных для изображения типа “портрет”. Последствия более сильного прореживания ( в четыре раза на рис. 3.4.в и в шесть раз на рис. 3.4.г) проявляются в основном в потере четкости. Субъективно потери качества представляются менее значительными, чем на рис. 3.3. Это находит свое объяснение в значительно меньшей ширине спектра, чем у изображения отпечатка пальца. Дискретизация исходного изображения соответствует граничной частоте  $\omega_{1гр} = 2\pi \cdot 86$ . Как видно из рис. 3.4.б, это значение намного превышает истинное значение  $\omega_{1гр}$ . Поэтому увеличение интервала дискретизации, иллюстрируемое рис.3.3.в,г, хотя и ухудшает картину, все же не приводит к таким разрушительным последствиям, как в предыдущем примере.



а)



б)



в)



г)

Рисунок 3.4 - Влияние интервала дискретизации на восстановление изображения «Портрет»

### 3.2. Квантование изображений

При цифровой обработке изображений непрерывный динамический диапазон значений яркости делится на ряд дискретных уровней. Эта процедура называется квантованием. Квантователь преобразует непрерывную переменную  $x$  в дискретную переменную  $x_{кв}$ , принимающую конечное множество значений  $\{r_1, \dots, r_L\}$ . Эти значения называются уровнями квантования. В общем случае преобразование выражается ступенчатой функцией (рис. 3.5). Если яркость  $x$  отсчета изображения принадлежит интервалу  $(d_j, d_{j+1}]$  (т.е., когда  $d_j < x \leq d_{j+1}$ ), то исходный отсчет заменяется на уровень квантования

$r_j$ , где  $d_j, j = \overline{1, L+1}$  - пороги квантования. При этом полагается, что динамический диапазон значений яркости ограничен и равен  $[d_1, d_{L+1}]$ .

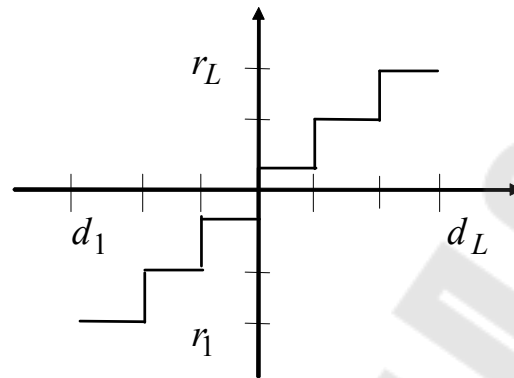


Рисунок 3.5 - Функция, описывающая квантование

Задача построения квантователя состоит в определении значений порогов  $d_j$  и уровней  $r_j$ . Простейший способ решения этой задачи состоит в разбиении динамического диапазона на одинаковые интервалы. Однако такое решение не является наилучшим. Если значения яркости большинства отсчетов изображения сгруппированы, например, в «темной» области и число уровней  $L$  ограничено, то целесообразно квантовать неравномерно. В «темной» области следует квантовать чаще, а в «светлой» реже. Это позволит уменьшить ошибку квантования  $\varepsilon = x - x_{кв}$ .

Таким образом, задачу построения квантователя можно сформулировать как задачу нахождения оптимальных значений  $d_j$  и  $r_j$ , удовлетворяющих некоторому критерию оптимизации. Обычно при фиксированном числе уровней квантователь оптимизируется по критерию минимальной среднеквадратической ошибки

$$E\{\varepsilon^2\} = E\{(x - x_{кв})^2\} = \min_{\{d_j, r_j\}}, \quad (3.12)$$

в предположении, что яркость  $x$  - случайная величина с известной плотностью вероятности  $w(x)$ .

Среднеквадратическая ошибка квантования (3.12) равна

$$E\{\varepsilon^2\} = \int_{d_1}^{d_{L+1}} (x - x_{кв})^2 w(x) dx = \sum_{j=1}^L \int_{d_j}^{d_{j+1}} (x - r_j)^2 w(x) dx. \quad (3.13)$$

Дифференцируя (3.13) по переменным  $d_j$ ,  $r_j$  и приравнявая производные нулю, получаем  $2L - 1$  нелинейных уравнений

$$(d_j - r_j)^2 w(d_j) - (d_j - r_{j-1})^2 w(d_j) = 0, \quad j = \overline{2, L}, \quad (3.14)$$

$$\int_{d_j}^{d_{j+1}} (x - r_j) w(x) dx = 0, \quad j = \overline{1, L}.$$

Следует отметить, что крайние пороги  $d_1$  и  $d_{L+1}$  определяются динамическим диапазоном яркости. Уравнения (3.14) нетрудно привести к виду

$$d_j = (r_j + r_{j-1}) / 2, \quad j = \overline{2, L}, \quad (3.15)$$

$$r_j = \frac{\int_{d_j}^{d_{j+1}} x w(x) dx}{\int_{d_j}^{d_{j+1}} w(x) dx}, \quad j = \overline{1, L}.$$

Из (3.15) следует, что пороги  $d_j$  должны располагаться по середине между двумя соседними уровнями  $r_{j-1}$  и  $r_j$ . Решение этих уравнений можно найти итеративным способом. Оптимальный квантователь, удовлетворяющий критерию (3.12), называется квантователем Ллойда-Макса, а среднеквадратическая ошибка для такого квантователя равна

$$E\{\varepsilon^2\} = E\{x^2\} - \sum_{j=1}^L r_j^2 \int_{d_j}^{d_{j+1}} w(x) dx \quad (3.16)$$

При равномерном распределении яркости нелинейные уравнения (3.15) можно представить в виде

$$d_j = d_1 + (j-1)\Delta, \quad j = \overline{2, L},$$

$$r_j = d_j + \Delta/2, \quad j = \overline{1, L},$$

$$\Delta = (d_{L+1} - d_1) / L,$$

а среднеквадратическая ошибка равна  $E\{\varepsilon^2\} = \frac{\Delta^2}{12}$ .

В системах цифровой обработки изображений стремятся уменьшить число уровней и порогов квантования, т.к. от их количества зависит длина двоичного кодового слова, которым представляются проквантованные отсчеты в ЭВМ. Однако при относительно небольшом числе уровней  $L$  на проквантованном изображении появляются ложные контуры. Они возникают вследствие скачкообразного изме-

нения яркости проквантованного изображения (рис.3.6) и особенно заметны на пологих участках ее изменения.

Ложные контуры значительно ухудшают визуальное качество изображения, т.к. зрение человека особенно чувствительно именно к контурам. При равномерном квантовании типичных изображений требуется не менее 64 уровней. На рис. 3.7 .а и 3.7.б приведены результаты равномерного квантования изображения «Портрет» соответственно на 256 и 14 уровней квантования.

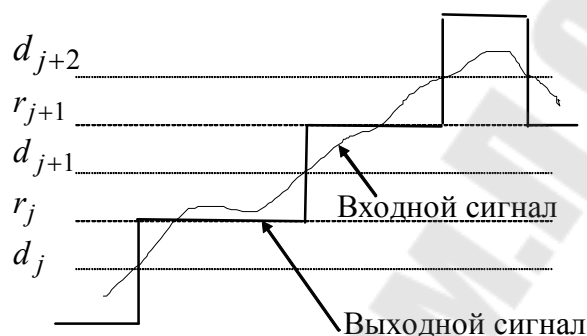
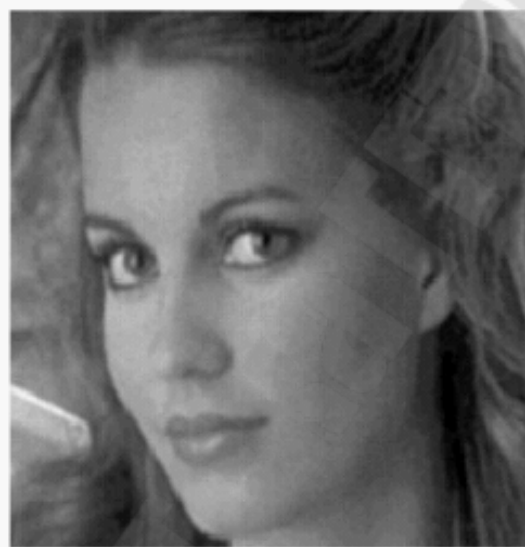


Рисунок 3.6 - К механизму возникновения ложных контуров



а)



б)

Рисунок 3.7 - Результаты равномерного квантования

В темных частях изображения заметны ложные контуры. Использование квантователя Ллойда-Макса позволяет существенно снизить их уровень

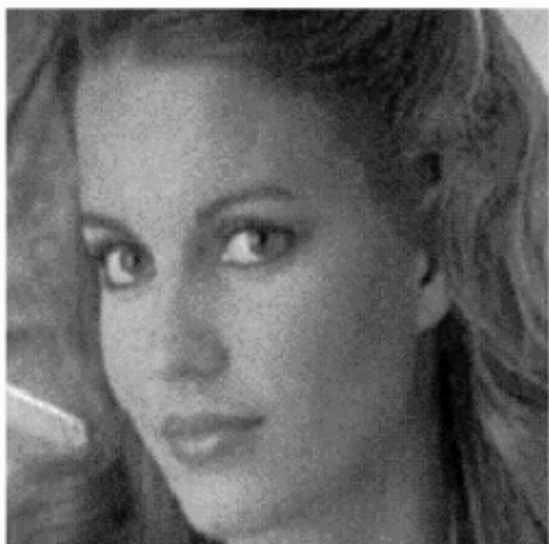
Чтобы избежать неравномерного квантования, которое не может быть выполнено с помощью стандартного АЦП, используют не-

линейные преобразования (рис. 3.8). Отсчет  $x$  исходного изображения подвергается нелинейному преобразованию, чтобы плотность распределения вероятностей преобразованных отсчетов  $y$  была равномерной, т.е. выполняется процедура эквализации. Затем отсчеты  $y$  квантуются с равномерным шагом и подвергаются обратному нелинейному преобразованию.



Рисунок 3.8 - Квантование с предварительным нелинейным преобразованием

Для разрушения ложных контуров Робертс предложил перед равномерным квантованием к отсчетам яркости добавлять шум с равномерной плотностью распределения вероятностей. Добавленный шум переводит одни отсчеты изображения на уровень выше, а другие на уровень ниже. Тем самым разрушаются ложные контуры. Дисперсия добавляемого шума должна быть небольшой, чтобы не привести к искажениям, воспринимаемым как «снег» на изображении, и в то же время достаточной для разрушения ложных контуров. Обычно используют равномерно распределенный шум на интервале  $[-\Delta/2, \Delta/2]$ . Результаты равномерного квантования на 14 и 8 уровней изображения «Портрет» с предварительным добавлением шума приведены на рис.3.9.а и 3.9.б. При 8-ми уровнях квантования добавляемый шум становится слишком заметным, однако ложные контуры разрушены практически полностью.



а)



б)

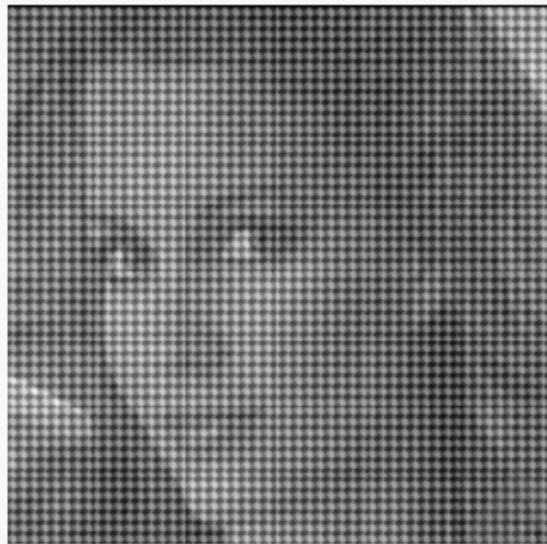
Рисунок 3.9 - Результаты равномерного квантования с предварительным добавлением шума

Еще один метод квантования используется в полиграфии. Это метод формирования растровых бинарных (2-х уровневых) изображений из полутоновых. При печати (например, газет или журналов) изображение формируется из белых и черных точек. Для этого все исходное изображение разбивается по пространственным координатам на одинаковые квадратные блоки. Обычно блок содержит  $5 \times 5$  элементов. К каждому отсчету блока добавляется число с соответствующими координатами из матрицы возмущающего сигнала, размеры которой равны размерам блока. Например, в качестве матрицы возмущающего сигнала используют числа:

$$\begin{bmatrix} 40 & 60 & 150 & 90 & 10 \\ 80 & 170 & 240 & 200 & 110 \\ 140 & 210 & 250 & 220 & 130 \\ 120 & 190 & 230 & 180 & 70 \\ 20 & 100 & 160 & 50 & 30 \end{bmatrix}.$$

Эта операция повторяется для всех блоков. Получаемое при этом изображение квантуется на два уровня. На рис. 3.10.а приведено полутоновое изображение «Портрет» с добавленным возмущающим сигналом. На рис. 3.10.б,в приведены результаты бинарного квантования изображения «Портрет» с добавленным возмущающим сигналом (рис.3.10.б) и без него (рис.3.10.в).





а)



б)



в)

Рисунок 3.10 - Растрирование изображений

Бинарное растровое изображение обеспечивает значительно лучшее зрительное впечатление, чем обычное бинарное изображение. Передача шкалы яркости при растрировании достигается благодаря изменению геометрических размеров белого пятна, наблюдаемого на черном фоне. Если в блоке сгруппировались «светлые» отсчеты, то геометрические размеры белого пятна максимальны и равны размеру блока. При уменьшении яркости его геометрические размеры также уменьшаются. Глаз человека выполняет локальное усреднение, создавая иллюзию наблюдения полутонного изображения. Процедура растрирования особенно эффективна при печати изображений с высоким разрешением, когда одиночное пятно едва различимо глазом.

### 3.3. Препарирование изображения

Препарирование представляет собой целый класс поэлементных преобразований изображений. Характеристики применяемых на практике процедур препарирования приведены на рис.3.11. Остановимся на описании некоторых из них.

Преобразование с пороговой характеристикой (рис. 3.11.а) превращает полутоновое изображение, содержащее все уровни яркости, в бинарное, точки которого имеют яркости  $y = 0$  или  $y = y_{\max}$ . Такая операция, называемая иногда бинаризацией или бинарным квантованием, может быть полезной, когда для наблюдателя важны очертания объектов, присутствующих на изображении, а детали, содержащиеся внутри объектов или внутри фона, не представляют интереса. Основной проблемой при проведении такой обработки является определение порога  $x_0$ , сравнение с которым яркости исходного изображения позволяет определить значение выходного изображения в каждой его точке. Наиболее оправданным для математического описания изображения является применение теории вероятностей, случайных процессов и случайных полей. При этом определение оптимального порога бинарного квантования представляет собой статистическую задачу. Статистическому подходу к обработке изображений в последующих разделах уделяется значительное внимание, в том числе и при решении задачи разделения точек изображения на два класса так называемой бинарной сегментации. Здесь же ограничимся обсуждением частного, но практически важного случая. Иногда при обработке приходится иметь дело с изображениями, хранимыми как полутоновые, но по своему содержанию мало отличающимися от бинарных.

К ним относятся текст, штриховые рисунки, чертежи, изображение отпечатка пальца, пример которого приведен на рис.3.13. Плотность вероятности  $w(x)$ , описывающая распределение яркости такого изображения, может содержать два хорошо разделяющихся пика. Интуитивно понятно, что порог бинарного квантования следует выбирать посередине провала между этими пиками, как это показано на рис.3.12.

Замена исходного полутонового изображения *бинарным препаратом* решает две основные задачи. Во-первых, достигается бóльшая наглядность при визуальном восприятии, чем у исходного изображения. Во-вторых, ощутимо сокращается объем запоминающего устрой-

ства для хранения изображения, поскольку бинарный препарат для записи каждой точки бинарного изображения требует лишь 1 бит памяти, в то время как полутонное изображение для решения той же задачи при наиболее часто применяемом формате представления - 8 бит. Пример бинаризации изображения отпечатка пальца приведен на рис.3.13

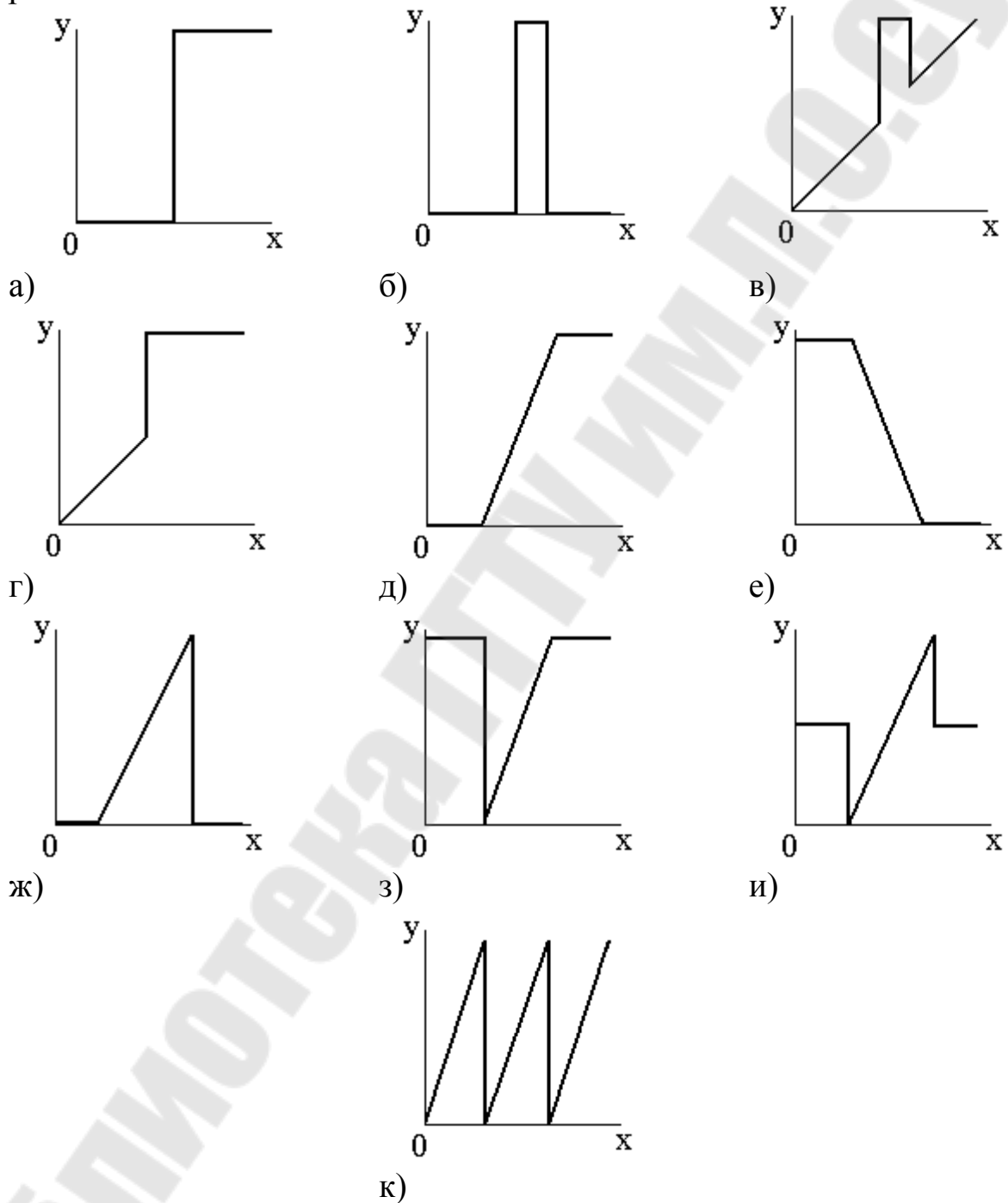


Рисунок 3.11 - Примеры преобразований, используемых при препарировании

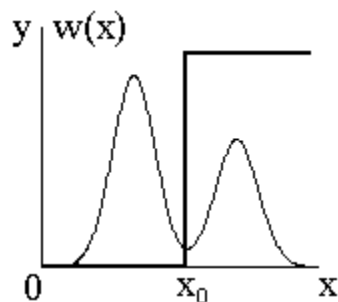


Рисунок 3.12 - К выбору порога бинарного квантования

Смысл других преобразований, представленных на рис. 3.11, нетрудно понять, рассматривая их характеристики. Например, преобразование рис. 3.11.б выполняет яростный срез изображения, выделяя те его участки, где яркость соответствует выделенному интервалу. При этом остальные участки оказываются полностью “погашенными” (имеют яркость, соответствующую уровню черного). Перемещая выделенный интервал по яркостной шкале и изменяя его ширину, можно детально исследовать содержание картины.

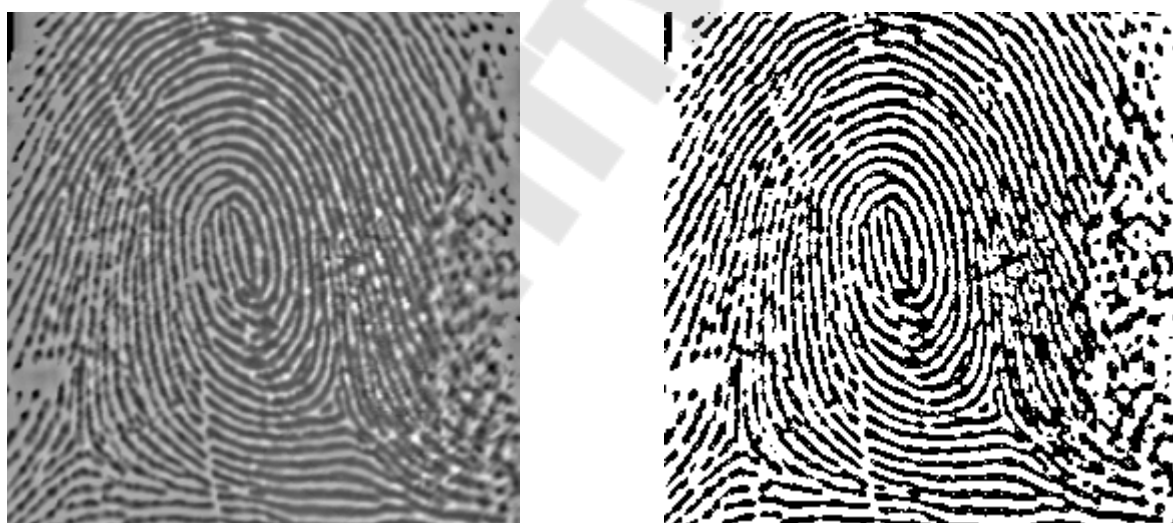


Рисунок 3.13 - Пример бинаризации изображения

Преобразование, приведенное на рис.3.11.ж, также позволяет повысить детальность наблюдаемой картины в выбранном диапазоне яркостей, однако в отличие от предыдущего здесь выходное изображение использует полный динамический диапазон. По существу, это преобразование представляет собой линейное контрастирование, применяемое к *избранному диапазону* входного изображения. Как и в

предыдущем варианте, участки, не попавшие в этот диапазон, образуют после препарирования черный фон.

Иногда наглядность изображения повышается применением преобразования типа пилообразного контрастирования. При этом различные яркостные диапазоны одновременно подвергаются локальному яркостному контрастированию. Однако необходимо иметь в виду, что данное преобразование, как и некоторые другие, может сопровождаться появлением ложных контуров на получаемом препарате.

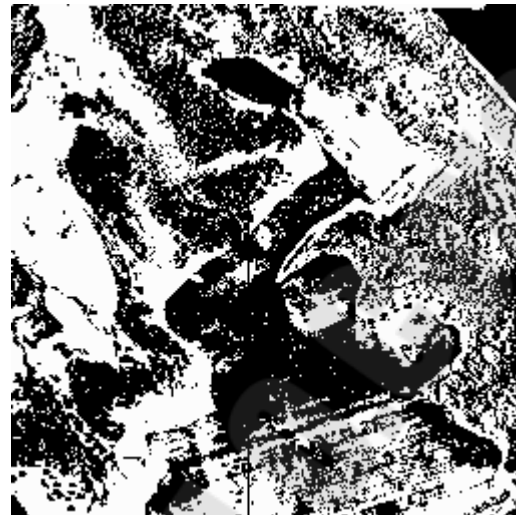
Аналогично можно качественно рассмотреть и остальные процедуры препарирования, представленные на рис 3.11.

На рис. 3.14 приведены результаты эксперимента, в котором к аэроснимку участка земли (рис, 3.14.а) применялись преобразования типа пороговая обработка (рис. 3.14.б) и пилообразное контрастирование (рис. 3.14.в). Первое приводит к выявлению границ отдельных участков, создавая общее интегральное представление о наблюдаемой сцене. Второе, наоборот, дает возможность наблюдения мелких деталей на всех участках изображения. Сочетание двух таких возможностей может оказаться полезным наблюдателю.

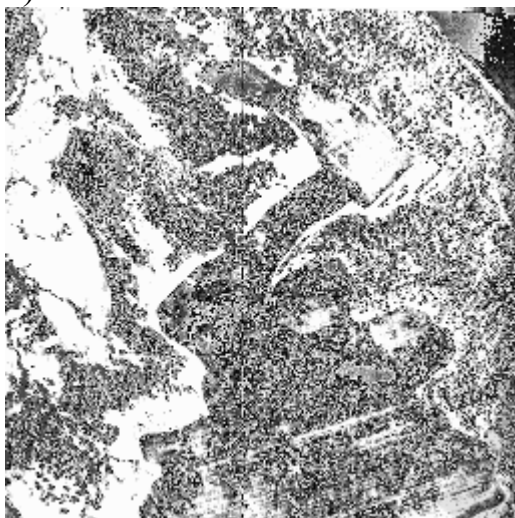
В заключение отметим, что препарирование часто используется и в автоматических системах обработки визуальной информации, поскольку подготавливаемый при этом препарат может содержать всю информацию, необходимую для последующей (вторичной) обработки. Например, если при наблюдении из космоса требуется автоматически обнаружить на изображении некоторый объект, имеющий известную конфигурацию, то для этого может быть достаточно бинарного препарата, передающего эту конфигурацию.



а)



б)



в)

Рисунок 3.14 - Примеры препарирования изображения

### 3.4. Фильтрация изображений

Обычно изображения, сформированные различными информационными системами, искажаются действием помех. Это затрудняет как их визуальный анализ человеком-оператором, так и автоматическую обработку в ПЭВМ. При решении некоторых задач обработки изображений в роли помех могут выступать и те или иные компоненты самого изображения. Например, при анализе космического снимка земной поверхности может стоять задача определения границ между ее отдельными участками - лесом и полем, водой и сушей и т.п. С точки зрения этой задачи отдельные детали изображения внутри разделяемых областей являются помехой.

Ослабление действия помех достигается фильтрацией. При фильтрации яркость (сигнал) каждой точки исходного изображения, искаженного помехой, заменяется некоторым другим значением яркости, которое признается в наименьшей степени искаженным помехой. Что может послужить основой для таких решений? Изображение часто представляет собой двумерную функцию пространственных координат, которая изменяется по этим координатам медленнее (иногда значительно медленнее), чем помеха, также являющаяся двумерной функцией. Это позволяет при оценке полезного сигнала в каждой точке кадра *принять во внимание* некоторое множество соседних точек, воспользовавшись определенной *похожестью* сигнала в этих точках. В других случаях, наоборот, признаком полезного сигнала являются резкие перепады яркости. Однако, как правило, частота этих перепадов относительно невелика, так что на значительных промежутках между ними сигнал либо постоянен, либо изменяется медленно. И в этом случае свойства сигнала проявляются при наблюдении его не только в локальной точке, но и при анализе ее окрестности. Заметим, что понятие окрестности является достаточно условным. Она может быть образована лишь ближайшими по кадру соседями, но могут быть окрестности, содержащие достаточно много и достаточно сильно удаленных точек кадра. В этом последнем случае, конечно, степень влияния далеких и близких точек на решения, принимаемые фильтром в данной точке кадра, будет совершенно различной.

Таким образом, идеология фильтрации основывается на рациональном использовании данных как из рабочей точки, так и из ее окрестности. В этом проявляется существенное отличие фильтрации от рассмотренных выше поэлементных процедур: фильтрация не может быть поэлементной процедурой обработки изображений.

Задача заключается в том, чтобы найти такую рациональную вычислительную процедуру, которая позволяла бы достигать наилучших результатов. Общепринято при решении этой задачи опираться на использование вероятностных моделей изображения и помехи, а также на применение статистических критериев оптимальности. Причины этого понятны - это случайный характер как информационного сигнала, так и помехи и это стремление получить минимальное *в среднем* отличие результата обработки от идеального сигнала. Многообразие методов и алгоритмов связано с большим разнообразием сюжетов, которые приходится описывать различными математическими моделями. Кроме того, применяются различные кри-

терии оптимальности, что также ведет к разнообразию методов фильтрации. Наконец, даже при совпадении моделей и критериев очень часто из-за математических трудностей не удается найти оптимальную процедуру. Сложность нахождения точных решений порождает различные варианты приближенных методов и процедур.

Более подробно вопросы фильтрации изображений рассмотрены в работах [18-20].

### 3.5. Сглаживание цветных изображений

Сглаживание (пространственная фильтрация) монохромных изображений осуществляется умножением значений всех пикселей в пространственной маске на соответствующие коэффициенты, суммированием и делением суммы на общее число элементов маски. Этот процесс (например, в пространстве RGB) определяется так же, как и при работе с монохромными изображениями, с той лишь разницей, что вместо одиночных, скалярных пикселей приходится работать с векторными величинами в представлении.

Пусть  $S_{xy}$  обозначает множество точек окрестности (маски) с центром в точке  $(x, y)$  некоторого цветного изображения. Среднее RGB векторов этой окрестности задается формулой

$$\bar{c}(x, y) = \frac{1}{K} \sum_{(s,t) \in S_{xy}} c(s, t),$$

где  $K$  – это число пикселей окрестности. Из свойства сложения векторов следует, что

$$\bar{c}(x, y) = \begin{bmatrix} \frac{1}{K} \sum_{(s,t) \in S_{xy}} R(s, t) \\ \frac{1}{K} \sum_{(s,t) \in S_{xy}} G(s, t) \\ \frac{1}{K} \sum_{(s,t) \in S_{xy}} B(s, t) \end{bmatrix}$$

Видно, что каждая компонента этого вектора получается при окрестностном осреднении каждой отдельной цветовой компоненты, которую можно выполнять по схеме стандартного осреднения монохромных изображений. Значит, сглаживание пространственным осреднением можно выполнять независимо на покомпонентной основе.



Результат будет совпадать с осреднением в цветовом векторном пространстве.

Повышение резкости цветных RGB изображений с помощью линейного пространственного фильтра выполняется теми же процедурами, что и в предыдущем параграфе, но с использованием фильтра, повышающего резкость. В данном параграфе этот метод демонстрируется на примере лапласиана. Из векторного анализа известно, что лапласиан вектора определяется как вектор, компоненты которого равны лапласиану от соответствующих скалярных компонент входного вектора. В цветовом пространстве RGB лапласиан от вектора с равен:

$$\nabla^2 [c(x, y)] = \begin{bmatrix} \nabla^2 [R(x, y)] \\ \nabla^2 [G(x, y)] \\ \nabla^2 [B(x, y)] \end{bmatrix}.$$

Эта формула означает, что лапласиан от цветного изображения можно находить с помощью вычисления лапласиана от каждой отдельной цветовой компоненты.

*Обнаружение контуров на цветных изображениях с помощью градиента.* Градиентом двумерной функции  $f(x, y)$  называется следующий вектор:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

Модуль (длина) этого вектора равен:

$$|\nabla f| = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} = [(\partial f / \partial x)^2 + (\partial f / \partial y)^2]^{1/2}$$

Часто эту величину можно аппроксимировать с помощью суммы абсолютных величин:

$$|\nabla f| \approx |G_x| + |G_y|$$

Такое приближение позволяет избежать возведения в квадрат и извлечения квадратного корня, но при этом оно ведет себя как производная (т. е. оно равно нулю на областях с постоянным цветом пикселей, и его величина пропорциональна степени изменения цвета на не-

однородных областях). Общепринято называть модуль градиента просто «градиентом». Основное свойство вектора-градиента состоит в том, что он направлен в сторону максимального роста  $f$  в точке с координатами  $(x, y)$ . Угол этого направления равен:

$$a(x, y) = \operatorname{arctg} \left( \frac{G_x}{G_y} \right)$$

Производные принято аппроксимировать разностями значений пикселей в малой окрестности изображения. Приближение частной производной в (вертикальном) направлении  $x$  по отношению к центральной точке окрестности (т. е.  $z_5$ ) определяется разностной формулой:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

Аналогично, частная производная по направлению  $y$  приближается разностным выражением:

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7).$$

Обе эти величины легко вычисляются во всех точках изображения с помощью свертки (функция `imfilter`) изображения с двумя масками. После этого приближение градиентного изображения получается сложением абсолютных величин двух фильтрованных изображений. Указанные выше маски являются масками Собела, которые строятся с помощью функции `f special`.

Описанный выше метод вычисления градиента часто используется в задачах обнаружения контуров на полутоновых изображениях. В данный момент мы интересуемся вычислением градиента в цветовом RGB пространстве. Однако приведенный метод применим только для двумерных пространств, и его нельзя распространить на большие размерности. Остается только подход, основанный на вычислении градиента для каждой цветовой компоненты изображения и сложении этих градиентов. К сожалению, такой путь не всегда приводит к решению задачи обнаружения контуров на цветных RGB изображениях.

Итак, проблема, прежде всего, заключается в том, чтобы определить градиент (модуль и направление) вектора  $s$ . Приведем далее некоторый вариант распространения понятия градиента на вектор-функции. Напомним, что для скалярной функции  $f(x, y)$  вектор гради-

ента указывает в направлении максимального роста функции  $f$  в точке с координатами  $(x, y)$ .

Пусть  $r$ ,  $g$  и  $b$  – единичные векторы, направленные вдоль осей  $R$ ,  $G$  и  $B$  цветового пространства RGB. Зададим векторы:

$$u = \frac{\partial R}{\partial x} r + \frac{\partial G}{\partial x} g + \frac{\partial B}{\partial x} b$$

$$v = \frac{\partial R}{\partial y} r + \frac{\partial G}{\partial y} g + \frac{\partial B}{\partial y} b$$

Пусть следующие величины  $g_{xx}$ ,  $g_{yy}$ , и  $g_{xy}$  задаются соответственно с помощью скалярных произведений этих векторов:

$$g_{xx} = u \cdot u = u^T u = \left| \frac{\partial R}{\partial x} \right|^2 + \left| \frac{\partial G}{\partial x} \right|^2 + \left| \frac{\partial B}{\partial x} \right|^2,$$

$$g_{yy} = v \cdot v = v^T v = \left| \frac{\partial R}{\partial y} \right|^2 + \left| \frac{\partial G}{\partial y} \right|^2 + \left| \frac{\partial B}{\partial y} \right|^2$$

$$g_{xy} = u \cdot v = u^T v = \frac{\partial R}{\partial x} \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} \frac{\partial B}{\partial y}.$$

Следует помнить, что величины  $R$ ,  $G$  и  $B$ , а, следовательно, и все  $g$  являются функциями от  $x$  и  $y$ . С помощью введенных обозначений можно показать, что угол направления максимального изменения (роста и убывания) вектора  $c(x, y)$  как функции  $(x, y)$  удовлетворяет уравнению:

$$\operatorname{tg} 2\theta = \frac{2g_{xy}}{(g_{xx} - g_{yy})},$$

а величина скорости изменения (т.е. модуль градиента) в направлении этого угла  $\theta$  задается выражением:

$$F_\theta(x, y) = \left\{ \frac{1}{2} [(g_{xx} + g_{yy}) + (g_{xx} - g_{yy}) \cos 2\theta + 2g_{xy} \sin 2\theta] \right\}^{1/2}.$$

Напомним, что  $\theta(x, y)$  и  $F_\theta(x, y)$  являются изображениями, размеры которых совпадают с размерами входного изображения. Элементами  $\theta(x, y)$  являются углы градиентов в каждой точке исходного изображения, а  $F_\theta(x, y)$  — модуль этого градиента.

Заметим, что  $\operatorname{tg}(\alpha) = \operatorname{tg}(\alpha \pm \pi)$ , поэтому если  $\theta_0$  является решением уравнения с  $\operatorname{tg}$ , то и величина  $\theta_0 + \pi/2$  также является его решением.

Кроме того,  $F_{\theta}(x,y) = F_{\theta+\pi}(x,y)$ , т.е. функцию  $F_{\theta}$ , достаточно вычислять лишь для тех величин  $\theta$ , которые расположены в полуоткрытом интервале  $[0, \pi)$ . Тот факт, что уравнения для  $\theta$  имеет два решения, которые различаются на  $90^{\circ}$ , означает, что с каждой точкой  $(x, y)$  изображения связаны два направления, расположенные ортогонально друг к другу. Вдоль одного из этих направлений скорость изменения функции  $F$  максимальна, а вдоль другого — минимальна, поэтому окончательный результат получается выбором максимума из этих двух чисел в каждой точке изображения. Частные производные, необходимые при решении приведенных выше уравнений, можно вычислять, например, с помощью операторов Собела.

### 3.6. Стандарт сжатия JPEG 2000 и система ROI

Одно из успешных применений вейвлетов [21-22] - их использование для сжатия изображений. Многочисленные исследования в этом направлении вылились в конце концов в принятие нового стандарта - **JPEG2000**. Пока изображения в этом формате еще не очень распространены, но нет никаких сомнений в том, что он полностью вытеснит нынешний популярный **JPEG**. Поэтому темой данной курсовой работы является описание и основные параметры алгоритма сжатия **JPEG 2000**, а так же описания технологии сжатия отдельных частей изображения (**Region of Interests, ROI**).

Разработка **JPEG2000** началась в 1996 году. Как раз к этому времени стало ясно существенное преимущество алгоритмов сжатия на основе вейвлетов перед ДКП, использованным в **JPEG**. Однако, только лишь ради увеличения степени сжатия вряд ли бы стали тратить громадные деньги. При создании нового стандарта наряду с достижением большей эффективности сжатия ставились еще и следующие цели:

Объединение в рамках единой технологии сжатия с потерями и без потерь. В дополнениях к **JPEG** есть режим сжатия без потерь, но эта технология защищена патентами.

1. Устойчивость алгоритма к ошибкам канала связи при передаче сжатого изображения. Здесь видна нацеленность нового стандарта на мобильные приложения, на передачу изображений по радиоканалу.

2. Унифицированная архитектура декодера. В **JPEG** имеется порядка 44 различных режимов декодирования, в зависимости от при-

ложения. Синтаксис **JPEG2000** таков, что в независимости от применяемого способа кодирования используется один и тот же декодер.

3. Масштабируемость. В зависимости от потребности, это может быть масштабируемость по размеру, разрешению, частотному содержанию, количеству цветов.

4. Обработка отдельных областей на изображении (Region Of Interest, смотрите далее). Например, пользователя может интересовать не все изображение улицы, а лишь фото отдельной машины, он выделяет его (мышью), и декодер с высоким качеством восстанавливает этот фрагмент. (Все невозможно "поднять" с высоким качеством из-за ограничений на объем передаваемой информации). Кстати говоря, такой режим имеется в видеокодеке Analog Devices **ADV611/ADV613**.

5. Сжатие изображений больших размеров.

6. Возможность обработки сжатого изображения без декомпрессии.

Рассмотрим основные блоки, входящие в структурную схему алгоритма сжатия **JPEG2000**.

1. *Предварительная обработка.* Изображение, как правило представляет собой набор неотрицательных целых чисел. На этапе предварительной обработки из него вычитают среднее. Кроме того, если изображение большого размера, то оно может быть разбито на части. Тогда каждая часть сжимается отдельно, а для предотвращения появления заметных линий на стыке восстановленных частей применяются специальные меры.

2. *Вейвлет-преобразование.* В первой части определены два вейвлет-фильтра: фильтр Добеши, для сжатия с потерями и биортогональный фильтр с целочисленными коэффициентами для сжатия без потерь. Во второй части стандарта разрешается применение любых фильтров, а также не только октавополосное разбиение, но и произвольное (вейвлет-пакеты и т.д.).

3. *Квантование.* В первой части стандарта определен равномерный квантователь с мертвой зоной.

4. *Энтропийное кодирование.* Применяется адаптивный арифметический кодер (в **JPEG** используется кодер Хаффмана). Ввиду патентных ограничений используется не QM-кодер разработки IBM, а чуть худший MQ-кодер, специально разработанный для **JPEG2000**. Кодирование ведется не всего изображения в целом и даже не отдельных субполос, а более мелких объектов - кодируемых блоков (КБ). Размер кодируемого блока может быть не более 4096

пикселей, высота не менее 4 пикселей. Такое разбиение хотя и снижает несколько коэффициент сжатия, но повышает устойчивость сжатого потока к ошибкам канала связи: ошибка испортит лишь небольшой блок. Кодирование блоков ведется в три этапа, битовыми плоскостями.

5. Сжатый поток данных упаковывается в пакеты. Именно благодаря гибкой и продуманной структуре пакетов возможно достижение целей разработки стандарта.

Алгоритм **JPEG-2000** разработан той же группой экспертов в области фотографии, что и **JPEG**. Основные отличия алгоритма в **JPEG 2000** от алгоритма в **JPEG** заключаются в следующем:

1. **Лучшее качество изображения при сильной степени сжатия.** Или, что тоже самое, большая степень сжатия при том же качестве для высоких степеней сжатия. Фактически это означает заметное уменьшение размеров графики «WEB-качества», используемой большинством сайтов.

2. **Поддержка кодирования отдельных областей с лучшим качеством.** Известно, что отдельные области изображения критичны для восприятия человеком (например, глаза на фотографии), в то время как качеством других можно пожертвовать (например, задний план). При «ручной» оптимизации увеличение степени сжатия проводится до тех пор, пока не будет потеряно качество в какой-то важной части изображения. Сейчас появляется возможность задать качество в критичных областях, сжав остальные области сильнее, т.е. мы получаем еще большую окончательную степень сжатия при субъективно равном качестве изображения.

3. **Основной алгоритм сжатия заменён на алгоритм wavelet.** Помимо указанного повышения степени сжатия это позволило избавиться от 8-пиксельной блочности, возникающей при повышении степени сжатия. Кроме того, плавное проявление изображения теперь изначально заложено в стандарт (**Progressive JPEG**, активно применяемый в Интернет, появился много позднее **JPEG**).

4. **Для повышения степени сжатия в алгоритме используется арифметическое сжатие.** Изначально в стандарте **JPEG** также было заложено арифметическое сжатие, однако позднее оно было заменено менее эффективным сжатием по Хаффману, поскольку арифметическое сжатие было защищено патентами. Сейчас срок действия основного патента истек, и появилась возможность улучшить алгоритм.

5. **Поддержка сжатия без потерь.** Помимо привычного сжатия с потерями новый **JPEG** теперь будет поддерживать и сжатие без потерь. Таким образом, становится возможным использование **JPEG** для сжатия медицинских изображений, в полиграфии, при сохранении текста под распознавание **OCR** системами и т.д.

6. **Поддержка сжатия однобитных (2-ух цветных) изображений.** Для сохранения однобитных изображений (рисунки тушью, отсканированный текст и т.п.) ранее повсеместно рекомендовался формат **GIF**, поскольку сжатие с использованием **ДКП** весьма неэффективно к изображениям с резкими переходами цветов. В **JPEG** при сжатии 1-битная картинка приводилась к 8-битной, т.е. увеличивалась в 8 раз, после чего делалась попытка сжимать, нередко менее чем в 8 раз. Сейчас можно рекомендовать **JPEG 2000** как универсальный алгоритм.

7. **На уровне формата поддерживается прозрачность.** Плавно накладывать фон при создании **WWW** страниц теперь можно будет не только в **GIF**, но и в **JPEG 2000**. Кроме того, поддерживается не только 1 бит прозрачности (пиксель прозрачен/непрозрачен), а отдельный канал, что позволит задавать плавный переход от непрозрачного изображения к прозрачному фону.

*Идея алгоритма (базовая схема).*

Базовая схема **JPEG-2000** очень похожа на базовую схему **JPEG**. Отличия заключаются в следующем:

1. Вместо дискретного косинусного преобразования (**DGT**) используется дискретное wavelet-преобразование (**DWT**).

2. Вместо кодирования по Хаффману используется арифметическое сжатие.

3. В алгоритм изначально заложено управление качеством областей изображения.

4. Не используется явно дискретизация компонент **U** и **V** после преобразования цветовых пространств, поскольку при **DWT** можно достичь того же результата, но более аккуратно.

### **3.7. Последовательность операций, используемых в алгоритме JPEG-2000**

#### **Шаг 1. Сдвиг по яркости (DC level shift)**

В **JPEG-2000** предусмотрен сдвиг яркости (**DC level shift**) каждой компоненты (**RGB**) изображения перед преобразованием в **YUV**.

Это делается для выравнивания динамического диапазона (приближения к  $U$  гистограммы частот), что приводит к увеличению степени сжатия. Формулу преобразования можно записать как:

$$I'(x,y) = I(x,y) - 2^{ST-1}$$

Значение степени  $ST$  для каждой компоненты  $R$ ,  $G$  и  $B$  свое (определяется при сжатии компрессором). При восстановлении изображения выполняется обратное преобразование:

$$I'(x,y) = I(x,y) + 2^{ST-1}$$

### Шаг 2. Изменение цветового пространства (RGB – YUY)

Переводим изображение из цветового пространства **RGB**, с компонентами, отвечающими за красную (Red), зеленую (Green) и синюю (Blue) составляющие цвета точки, в цветовое пространство **YUV**. Этот шаг аналогичен **JPEG**, за тем исключением, что кроме преобразования с потерями, предусмотрено также и преобразование без потерь. Его матрица выглядит так:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} \left[ \frac{R + 2G + B}{4} \right] \\ R - G \\ B - G \end{pmatrix}$$

Обратное преобразование осуществляется с помощью обратной матрицы:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} U + G \\ Y - \left[ \frac{U + V}{4} \right] \\ V + G \end{pmatrix}$$

### Шаг 3. Дискретное wavelet преобразование (DWT)

Дискретное вэйвлет(wavelet)-преобразование (**DWT**) может быть двух видов: для случая сжатия с потерями и для сжатия без потерь. Его коэффициенты задаются таблицами, приведенными ниже.

Для сжатия с потерями коэффициенты выглядят как:



Коэффициенты при упаковке		
$i$	Низкочастотные коэффициенты $h_L(i)$	Высокочастотные коэффициенты $h_H(i)$
0	1.115087052456994	0.6029490182363579
$\pm 1$	0.5912717631142470	-0.2668641184428723
$\pm 2$	-0.05754352622849957	-0.07822326652898785
$\pm 3$	-0.09127176311424948	0.01686411844287495
$\pm 4$	0	0.02674875741080976
Другие $i$	0	0

Коэффициенты при распаковке		
$i$	Низкочастотные коэффициенты $g_L(i)$	Высокочастотные коэффициенты $g_H(i)$
0	0.6029490182363579	1.115087052456994
$\pm 1$	-0.2668641184428723	0.5912717631142470
$\pm 2$	-0.07822326652898785	-0.05754352622849957

$\pm 3$	0.01686411844287495	-0.09127176311424948
$\pm 4$	0.02674875741080976	0
Другие $i$	0	0

Для сжатия без потерь коэффициенты задаются так:

$i$	При упаковке		При распаковке	
	Низкочастотные коэффициенты $h_L(i)$	Высокочастотные коэффициенты $h_H(i)$	Низкочастотные коэффициенты $g_L(i)$	Высокочастотные коэффициенты $g_H(i)$
0	6/8	1	1	6/8
$\pm 1$	2/8	-1/2	1/2	-2/8
$\pm 2$	-1/8	0	0	-1/8

Само преобразование в одномерном случае представляет собой скалярное произведение коэффициентов фильтра на строку преобразуемых значений (например, на строку изображения). При этом четные выходящие значения формируются с помощью низкочастотного преобразования, а нечетные - с помощью высокочастотного:

$$y_{output}(2n) = \sum_{j=0}^{N-1} x_{input}(j) \square h_H(j-2n)$$

$$y_{output}(2n+1) = \sum_{j=0}^{N-1} x_{input}(j) \square h_L(j-2n-1)$$

Поскольку большинство  $h_L(i)$ , кроме окрестности  $i=0$ , равны 0, то можно переписать приведенные формулы с меньшим количеством операций.

Для простоты рассмотрим случай сжатия без потерь.

$$y_{out}(2n) = \frac{-x_{in}(2n-1) + 2x_{in}(2n) + 6x_{in}(2n+1) + 2x_{in}(2n+2) - x_{in}(2n+3)}{8}$$

$$y_{out}(2n+1) = -\frac{x_{in}(2n)}{2} + x_{in}(2n+1) - \frac{x_{in}(2n+2)}{2}$$

Легко показать, что данную запись можно эквивалентно переписать, уменьшив еще втрое количество операций умножения и деления (однако теперь необходимо будет подсчитать сначала все нечетные  $y$ ). Добавим также операции округления до ближайшего целого, не превышающего заданное число  $L$ , обозначаемые как  $\lfloor a \rfloor$ :

$$y_{out}(2n+1) = x_{in}(2n+1) - \left\lfloor \frac{x_{in}(2n) + x_{in}(2n+2)}{2} \right\rfloor$$

$$y_{out}(2n) = x_{in}(2n) + \left\lfloor \frac{y_{out}(2n-1) + y_{out}(2n+1) + 2}{4} \right\rfloor$$

Рассмотрим на примере, как работает данное преобразование. Для того, чтобы преобразование можно было применять к крайним пикселям изображения, оно симметрично достраивается в обе стороны на несколько пикселей, как показано на рисунке ниже. В худшем случае (сжатие с потерями) нам необходимо достроить изображение на 4 пикселя.



Рисунок 3.15 - Симметричное расширение изображения (яркости АБ .. Е) по строке вправо и влево

Пусть мы преобразуем строку из 10 пикселей. Расширим её значения вправо и влево и применим **DWT** преобразование.

$$\begin{array}{r}
 n \\
 x_{in} \\
 y_{out}
 \end{array}
 \begin{array}{r}
 -2 \\
 3 \\
 0
 \end{array}
 \begin{array}{r}
 -1 \\
 2 \\
 0
 \end{array}
 \left| \begin{array}{cccccccccccc}
 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 1 & 2 & 3 & 7 & 10 & 15 & 12 & 9 & 10 & 5 \\
 1 & 0 & 3 & 1 & 11 & 4 & 13 & -2 & 8 & -5
 \end{array} \right| \begin{array}{r}
 10 \\
 11 \\
 10 \\
 9
 \end{array}$$

Получившаяся строка 1, 0, 3, 1, 11, 4, 13, -2, 8, -5 и является цепочкой, однозначно задающей исходные данные. Совершив аналогичные преобразования с коэффициентами для распаковки, приведенными выше в таблице, получим необходимые формулы:

$$\begin{aligned}
 x_{out}(2n) &= y_{out}(2n) - \left[ \frac{y_{out}(2n-1) + y_{out}(2n+1) + 2}{4} \right] \\
 x_{out}(2n+1) &= y_{out}(2n+1) + \left[ \frac{x_{out}(2n) + x_{out}(2n+2)}{2} \right]
 \end{aligned}$$

Легко проверить (используя преобразование упаковки), что значения на концах строк в  $y_{out}$  также симметричны относительно  $n=0$  и 9. Воспользовавшись этим способом, расширим строку вправо и влево и применим обратное преобразование:

$$\begin{array}{r}
 n \\
 y_{out} \\
 x_{out}
 \end{array}
 \begin{array}{r}
 -2 \\
 0 \\
 1
 \end{array}
 \begin{array}{r}
 -1 \\
 0 \\
 1
 \end{array}
 \left| \begin{array}{cccccccccccc}
 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 1 & 0 & 3 & 1 & 11 & 4 & 13 & -2 & 8 & -5 \\
 1 & 2 & 3 & 7 & 10 & 15 & 12 & 9 & 10 & 5
 \end{array} \right| \begin{array}{r}
 10 \\
 11 \\
 8 \\
 -2 \\
 10
 \end{array}$$

Как видим, мы получили исходную цепочку ( $x_{in}=x_{out}$ ).

Далее к строке применяется чересстрочное преобразование, суть которого заключается в том, что все четные коэффициенты переписываются в начало строки, а все нечетные - в конец. В результате этого преобразования в начале строки формируется «уменьшенная копия» всей строки (низкочастотная составляющая), а в конце строки информация о колебаниях значений промежуточных пикселей (высокочастотная составляющая).

$$\begin{array}{r}
 y_{out} \\
 y'_{out}
 \end{array}
 \left| \begin{array}{cccccccc}
 1 & 0 & 3 & 1 & 11 & 4 & 13 & -2 & 8 & -5 \\
 1 & 3 & 11 & 13 & 8 & 0 & 1 & 4 & -2 & -5
 \end{array} \right|$$

Это преобразование применяется сначала ко всем строкам изображения, а затем ко всем столбцам изображения. В результате изображение делится на 4 квадранта. В первом квадранте будет сформирована уменьшенная копия изображения, а в остальных трех высокочастотная информация. После чего преобразование повторно применяется уже только к первому квадранту изображения по тем же правилам (преобразование второго уровня).

Для корректного сохранения результатов под данные 2 и 3 квадрантов выделяется на один бит больше, а под данные 4-го квадранта на 2 бита больше. Т.е. если исходные данные были 8-битные, то на 2 и 3 квадранты нужно 9 бит, а на 4-й 10, независимо от уровня применения **DWT**. При записи коэффициентов в файл можно использовать иерархическую структуру **DWT**, помещая коэффициенты преобразований с большего уровня в начало файла. Это позволяет получить «изображение для предварительного просмотра», прочитав небольшой участок данных из начала файла, а не распаковывая весь файл, как это приходилось делать при сжатии изображения целиком. Иерархичность преобразования может также использоваться для плавного улучшения качества изображения при передаче его по сети.

#### **Шаг 4. Квантование.**

Так же, как и в алгоритме **JPEG**, после **DWT** применяется квантование. Коэффициенты квадрантов делятся на заранее заданное число. При увеличении этого числа снижается динамический диапазон коэффициентов, они становятся ближе к **U**, и мы получаем большую степень сжатия. Варьируя эти числа для разных уровней преобразования, для разных цветовых компонент и для разных квадрантов, мы очень гибко управляем степенью потерь в изображении. Рассчитанные в компрессоре оптимальные коэффициенты квантования передаются в декомпрессор для однозначной распаковки.

#### **Шаг 5. Арифметическое сжатие.**

Для сжатия получающихся массивов данных в **JPEG 2000** используется вариант арифметического сжатия, называемый **MQ-кодер**, прообраз которого (**QM-кодер**) рассматривался еще в стандарте **JPEG**, но реально не использовался из-за патентных ограничений. Ввиду этого, используется не **QM-кодер** разработки **IBM**, а чуть худший, **MQ-кодер**, специально разработанный для **JPEG2000**. Кодирование ведется не всего изображения в целом и даже не отдельных субполос, а более мелких объектов - кодируемых блоков. Размер кодируемого блока может быть не более 4096 пикселей, высота не менее 4 пикселей. Такое разбиение хотя и снижает несколько коэффициент сжатия, но повышает устойчивость сжатого потока к ошибкам канала связи: ошибка испортит лишь небольшой блок. Кодирование блоков ведется в три этапа, битовыми плоскостями.

Области повышенного качества (**Regions of Interest**). Основная задача, которую мы решаем - повышение степени сжатия изображений. Когда практически достигнут предел сжатия изображения в целом и

различные методы дают очень небольшой выигрыш, мы можем существенно (в разы) увеличить степень сжатия за счет изменения качества разных участков изображения.

Проблемой этого подхода является то, что необходимо каким-то образом получать расположение наиболее важных для человека участков изображения. Например, таким участком на фотографии человека является лицо, а на лице - глаза. Если при сжатии портрета с большими потерями будут размыты предметы, находящиеся на заднем плане - это будет несущественно, Однако, если будет размыто лицо или глаза - экспертная оценка степени потерь будет хуже.



Рисунок 3.16 - Локальное улучшение качества областей изображения

Работы по автоматическому выделению таких областей активно ведутся. В частности, созданы алгоритмы автоматического выделения лиц на изображениях. Продолжаются исследования методов выделения наиболее значимых (при анализе изображения мозгом человека) контуров и т.д. Однако очевидно, что универсальный алгоритм в ближайшее время создан не будет, поскольку для этого требуется построить полную схему восприятия изображений мозгом человека.

На сегодня вполне реально применение полуавтоматических систем, в которых качество областей изображения будет задаваться интерактивно. Данный подход уменьшает количество возможных областей применения модифицированного алгоритма, но позволяет достичь большей степени сжатия.

Такой подход логично применять, если:

1. Для приложения должна быть критична (максимальна) степень сжатия, причем настолько, что возможен индивидуальный подход к каждому изображению.

2. Изображение сжимается один раз, а разжимается множество раз.

В качестве примеров приложений, удовлетворяющим этим ограничениям, можно привести практически все мультимедийные продукты на CD-ROM. И для CD-ROM энциклопедий, и для игр важно записать на диск как можно больше информации, а графика, как правило, занимает до 70% всего объема диска. При этом технология производства дисков позволяет сжимать каждое изображение индивидуально, максимально повышая степень сжатия.

Интересным примером являются WWW-сервера. Для них тоже, как правило, выполняются оба изложенных выше условия. При этом совершенно не обязательно индивидуально подходить к каждому изображению, поскольку по статистике 10% изображений будут запрашиваться 90% раз. Т.е. для крупных справочных или игровых серверов появляется возможность уменьшать время загрузки изображений и степень загруженности каналов связи адаптивно.

### **3.8. Особенности сжатия видеоданных в формате MPEG**

Рассмотрим структурную схему алгоритма сжатия стандарта MPEG 1 (рис. 3.17). Можно выделить три основные подсистемы кодера: подсистема устранения пространственной избыточности, подсистема устранения временной избыточности и подсистема управления скоростью потока.

Избыточность является следствием определенных корреляционных связей. Наличие корреляции означает, что некоторый элемент изображения находится в определенной зависимости от соседних элементов в пространстве и во времени. Избыточность по восприятию (психовизуальная) связана с особенностями зрения человека. Например, цветовое разрешение человеческого зрения ниже яркостного, и эта особенность учтена во всех стандартных аналоговых системах цветового кодирования. В NTSC, PAL, SECAM цветовое разрешение существенно понижено по отношению к яркостному.

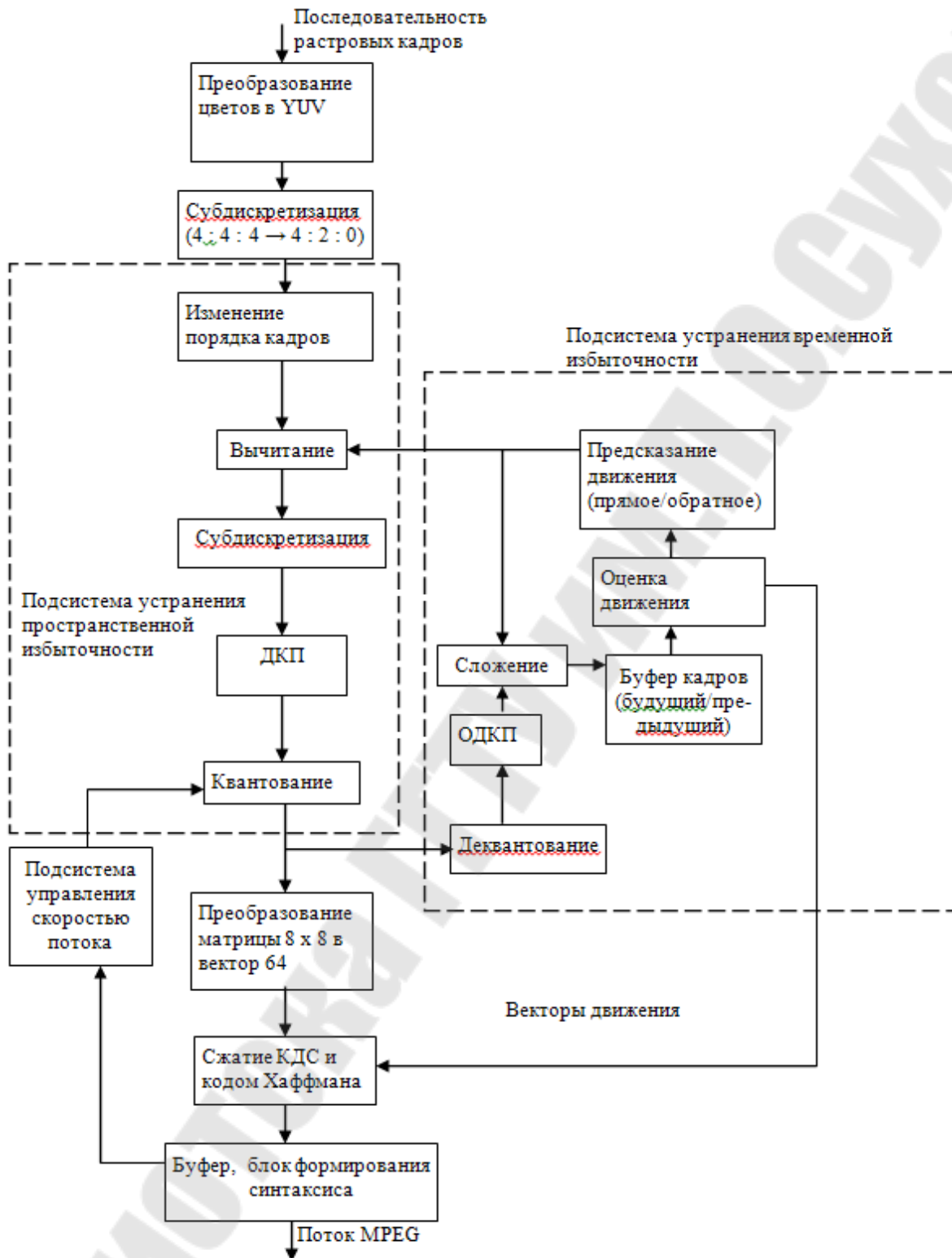


Рисунок 3.17 - Структурная схема сжатия в стандарте MPEG 1

В первой подсистеме блок преобразования цветового пространства и субдискретизации (уменьшения разрешения компонентов

цветности) предназначен для первичного устранения психовизуальной избыточности кадра. Блок смены порядка обработки кадров необходим для работы системы компенсации движения, блок дискретного косинусного преобразования – для получения спектральных коэффициентов и устранения пространственной корреляции пикселей, блок квантования – для устранения вторичной психовизуальной избыточности. Блок преобразования матрицы коэффициентов ДКП (Дискретного Косинусного Преобразования) в вектор зигзагообразным сканированием – для увеличения эффективности кода Хаффмана. Данная подсистема позволяет получить коэффициент сжатия до 10 : 1.

Во второй подсистеме устранение временной избыточности производится на основе межкадровой обработки потока, объединения кадров в группы, предсказания межкадровых сдвигов и компенсации ошибок предсказания. Блоки деквантования и обратного ДКП (ОДКП) необходимы потому, что в алгоритме компенсации движения совпадающие области ищутся не в оригинальном видеопотоке, а в кадрах, уже подвергшихся сжатию и распаковке, так как у распаковывающего модуля нет доступа к исходным видеокдрам. Данная подсистема позволяет получить коэффициент сжатия до 30 : 1

Подсистема управления обеспечивает постоянство скорости видеопотока, управляя числом частотных коэффициентов и длиной группы кадров.

Исходные видеоданные представляются в виде потока растровых изображений. Растр – форма представления изображения в виде элементов (пикселей), упорядоченных в строки и столбцы. Название “пиксель” образовано как сокращение от английского pixel (picture cell – ячейка изображения) – это наименьший элемент, из множества которых создается растровое изображение. Несмотря на четкую структуру пиксельной сетки, тоновые переходы в растровых картинках не выглядят дискретными. Это обусловлено, во-первых, тем, что пиксели обычно очень малы и при отображении без увеличения практически неразличимы. Во-вторых, плавные изменения цвета или света передаются за счет постепенного изменения значений соседних пикселей от одного к другому.

При записи изображения обычно используется по 8 бит (1 байт) для представления 256 уровней яркости красного, зеленого и синего цветов (RGB). Таким образом, для хранения одного элемента изображения (пикселя) требуется 3 байта памяти. Например, стан-



дартный видеокادر формата 352 x 288 пикселей требует 304 128 байтов.

При записи изображений традиционно используется RGB-представление, когда на каждую цветовую составляющую приходится по одному байту. Также существуют и другие способы записи изображений.

Цветное изображение содержит как яркостную, так и цветовую информацию. Яркостная информация существенно важнее для качества изображения, чем цветовая, поскольку зрение человека значительно сильнее реагирует на небольшие изменения яркости, чем на небольшие изменения цветового тона. Это обусловлено тем, что в сетчатке глаза плотность палочек, пригодных для восприятия только яркостной информации, гораздо выше, чем колбочек.

В связи с этим в схеме кодирования MPEG, видеоданные переводятся в YUV-представление, также называемое YCrCb (Y – компонент яркости и U (или Cr) и V (или Cb) – два компонента цветности, где Cr – Chromatic red (хроматический красный), Cb – Chromatic blue (хроматический синий)). Это необходимо потому, что при представлении изображений в RGB-виде нельзя различить яркостную и цветовую информацию.

Преобразование цветового пространства RGB в цветовое пространство YUV представляется с помощью матрицы перехода в виде

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,229 & 0,587 & 0,144 \\ 0,5 & -0,4187 & -0,0813 \\ 0,1687 & -0,3313 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}.$$

Обратное преобразование осуществляется с помощью обратной матрицы перехода в виде

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1,402 \\ 1 & -0,34414 & -0,71414 \\ 1 & 1,772 & 0 \end{pmatrix} \cdot \begin{pmatrix} Y \\ U \\ V \end{pmatrix} - \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}.$$

Теоретически каждый элемент изображения требует 3 байта. Такое представление, когда и яркость, и компоненты цветности имеют равное число независимых значений, обычно обозначают как схема 4 : 4 : 4. Способ сжатия цветовой информации называется субдискретизацией и заключается в объединении цветовой информации для соседних элементов изображений. При использовании данного спосо-

ба значения яркости  $Y$  запоминаются для каждого элемента изображения. Для значений  $U$  и  $V$  вычисляется и запоминается только среднее значение для четырех (схема субдискретизации  $4 : 2 : 2$ ), восьми (схема субдискретизации  $4 : 1 : 1$ ) и 16 элементов (схема субдискретизации  $16 : 1 : 1$ ). Требуемый объем памяти на каждый пиксел при таких способах составляет, соответственно, 12, 10 и 9 битов. Таким образом, количество данных на элемент изображения можно сократить с 24 до 12, 10 или 9 битов соответственно.

*Дискретное косинусное преобразование.* Для уменьшения корреляции соседних пикселей применяются различные обратимые преобразования, представляющие исходные данные в виде некоррелированных коэффициентов. ДКП является одним из эффективных преобразований для решения задачи декорреляции и концентрации энергии в спектральных составляющих. Косинусное преобразование в отличие от преобразования Фурье применяется только для симметричных функций.

При использовании частотно-временных преобразований используется понятие периодического расширения функции, заключающееся в следующем: если преобразуется дискретный ряд отсчетов, то его спектр становится периодичным, а в случае преобразования частотного спектра периодически продолжается восстановленный дискретный ряд данных.

*Квантование как средство управления соотношением качество-сжатие.* Динамический диапазон коэффициентов ДКП превышает в 8 раз динамический диапазон значений пикселей исходного изображения. Например, при 8-битном представлении отсчетов изображения его динамический диапазон составляет 256 дискретных уровней (значения от 0 до 255), а динамический диапазон коэффициентов спектра ДКП – 2 048 значений уровней (от 0 до 2 048 для коэффициентов постоянной составляющей DC, и от  $-1\,023$  до  $+1\,024$  для переменных составляющих AC).

Кодирование коэффициентов ДКП в таком широком динамическом интервале потребует в последующих блоках MPEG кодера перехода от 8-битного к 11-битному коду. Для предотвращения усложнения кодера после ДКП производится сжатие динамического диапазона сигналов коэффициентов ДКП за счет увеличения шага квантования в 8 раз. Эта операция сводится к делению полученных в матрице значений коэффициентов ДКП на 8. Результат деления затем округляется до ближайших целых значений уровней новой шкалы квантова-

ния. Например, если исходное значение коэффициента ДКП было 22, то после деления на 8 ( $22/8 = 2,75$ ) и округления до ближайшего целого значения новое значение будет 3. При этом новый динамический интервал составит 256 дискретных уровней от  $-128$  до  $+127$ .

После выравнивания динамического диапазона над коэффициентами ДКП осуществляется взвешенное квантование для сокращения избыточности в высокочастотной области. Математически данное квантование представляет собой деление рабочей матрицы  $\|B\|$  на матрицу квантования  $\|Q\|$  поэлементно (6). Для каждого компонента ( $Y$ ,  $U$  и  $V$ ) в общем случае, задается своя матрица квантования ( $Q_Y[p,q]$ ,  $Q_U[p,q]$ ,  $Q_V[p,q]$ , которая позволяет выбирать уровни сжатия и качества изображения:

$$B_Q[p,q] = \text{Round}\left(\frac{B[p,q]}{Q_Y[p,q]}\right).$$

Матрицы квантования в формуле (11.23) задаются таким образом, чтобы обеспечить более точную передачу низкочастотной информации, так как чувствительность глаз в данной области наибольшая, и обнулить большинство высокочастотных коэффициентов. Это обеспечивает значительное сжатие потока данных, но приводит к снижению разрешения изображения и возможному появлению ложных деталей (в частности на границах блоков). Чем грубее используемое квантование, тем больше степень сжатия, но ниже качество результирующего сигнала. Точность кодирования коэффициентов зависит от шага квантования, который выбирается разным для различных коэффициентов матрицы ДКП.

Коэффициент, соответствующий постоянной составляющей ТВ сигнала, кодируется с использованием 10 бит, потому что при более грубом квантовании соседние блоки начинают отличаться по яркости. На экране это проявляется в виде шахматной структуры.

*Структура потока данных MPEG.* Поток MPEG разделен на несколько иерархических уровней (рис.3.18) для улучшения обработки ошибок и упрощения произвольного доступа и редактирования. Каждый слой имеет свой уникальный заголовок длиной в 32 бита.

Первый уровень – это собственно поток видео (Video sequence layer). Второй уровень – групповой кадр (ГК, Group of Pictures – GOP), состоящий из нескольких кадров разного типа:  $I$  кадры (Intra-coded – внутрикадровое кодирование) кодируются (сжимаются) без учета соседних кадров; предсказываемые  $P$  кадры (Predicted – кодирование с предсказанием) кодируются с учетом предыдущего  $I$  или  $P$

кадра; *B* кадры двунаправленного предсказания (Bidirectional – двунаправленное предсказание) кодируются с учетом предыдущего и последующего *I* или *P* кадров. ). Каждая GOP обязательно начинается с *I* и с определенной периодичностью содержит *P*-кадры. Ее структуру описывают как  $M/N$ , где  $M$  - общее число кадров в группе, а  $N$  - интервал между *P*-кадрами.

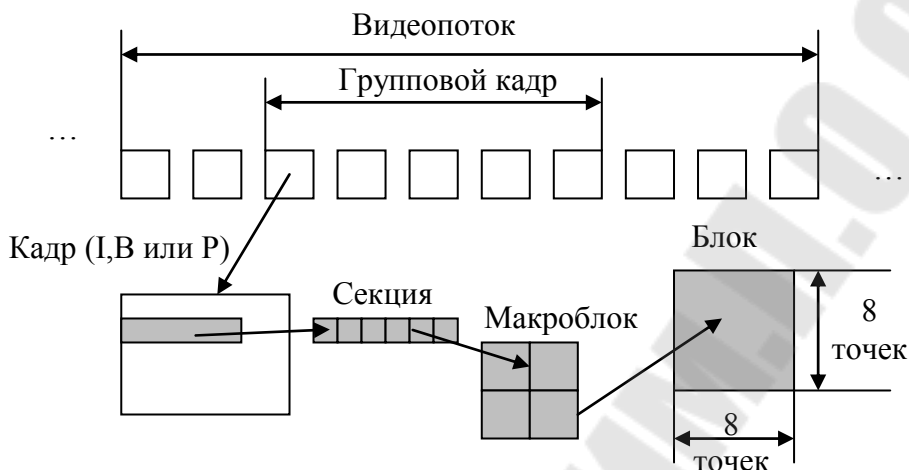


Рисунок 3.18 - Структура потока MPEG

Третий уровень потока данных – это слой отдельных кадров (Picture layer) того или иного типа, а четвертый уровень – секционный (Slice layer). Секция (обычно ее ширина равна ширине кадра) состоит из определенного количества макроблоков размером 16 x 16 пикселей. Пятый уровень потока данных – уровень макроблоков. В *I* кадре макроблоки должны быть закодированы как внутренние, т.е. без ссылок на предыдущие или последующие. Макроблоки в *P* кадре могут быть как внутренними *I* блоками, так и использовать данные предыдущих кадров. Алгоритмы кодирования *B* кадров зависят от динамики изображения.

В MPEG-2 предусмотрено пять способов кодирования. Первый – это компенсация движения и предсказание вперед по ближайшим предшествующим *I* или *P* кадрам. При появлении в кодируемом *B* кадре новых объектов применяется предсказание назад по ближайшим последующим *I* или *P* кадрам вместе с компенсацией движения. Третий алгоритм включает в себя компенсацию движения и двунаправленное предсказание по предшествующим и последующим *I* или *P* кадрам. Четвертый основан на внутрикадровом предсказании без компенсации движения (он чаще всего используется при резкой смене плана или высоких скоростях движения отдельных фрагментов кар-

тинки). Пятый способ представления необходим, если в нескольких кадрах изменений нет или они незначительны. В этом случае блок пропускается, а декодер (декомпрессор) при восстановлении использует самый ранний вариант блока. В заголовке каждого макроблока есть элемент, определяющий его тип.

*Межкадровое предсказание движения.* Компенсация движения является одной из важнейших составных частей стандартов MPEG 1 и MPEG 2 (рис.3.19). Метод компенсационного предсказания движения позволяет значительно уменьшить временную избыточность видеопотока. Если следующий кадр содержит сдвинутые части предыдущего кадра, то в этом случае выгодно передавать не весь кадр, а только информацию о движении и изменении сдвинутого пикселя. В виду высокой пространственной корреляции достаточно передавать один общий вектор движения для макроблока размером 16 x 16 пикселей.

Для каждого блока первого кадра производится поиск наиболее похожего блока во втором и вычисляется вектор движения, указывающий направление движения блока от первого кадра ко второму. Поправка компенсации движения  $E[m,n]$  вычисляется вычитанием значений пикселей найденного сдвинутого блока из значений пикселей исходного блока первого кадра:

$$E[m,n] = A_i[m,n] - A_j([m,n] + M_{ij}) \quad ,$$

где  $E[m,n]$  – значения ошибок предсказания для каждого пикселя;  $A_i[m,n]$  – значения  $(m,n)$ -го пикселя в блоке  $i$ -го кадра;  $A_j[m,n]$  – значения  $(m,n)$ -го пикселя в блоке  $j$ -го кадра;  $M_{ij}$  – вектор движения для макроблока в  $i$ -м кадре относительно макроблока в  $j$ -м кадре, состоящий из двух значений, задающих смещение макроблока по вертикали и горизонтали.

Зона поиска должна быть достаточно большой, чтобы быстро движущийся макроблок изображения первого кадра не вышел из зоны поиска второго кадра. Размеры зоны поиска ограничиваются объемом вычислений, которые необходимо выполнить в реальном масштабе времени.

Чаще всего размеры зоны поиска задаются в 4 раза больше размеров отдельного макроблока (64 x 64). Например, во время формирования  $P$  кадра надо определить координаты движения при предсказании вперед. Для этого берется макроблок текущего  $P$  кадра и ищется его новое положение в зоне поиска предыдущего  $I$  или  $P$  кадра, затем вычисляются межкадровые разности точек. Положение макроблока, при котором суммарное значение модулей межкадровых

разностей макроблока получается наименьшим, принимается за его реальное перемещение, после чего координаты вектора движения рассчитываются как смещение макроблока по вертикали и горизонтали относительно его начального положения

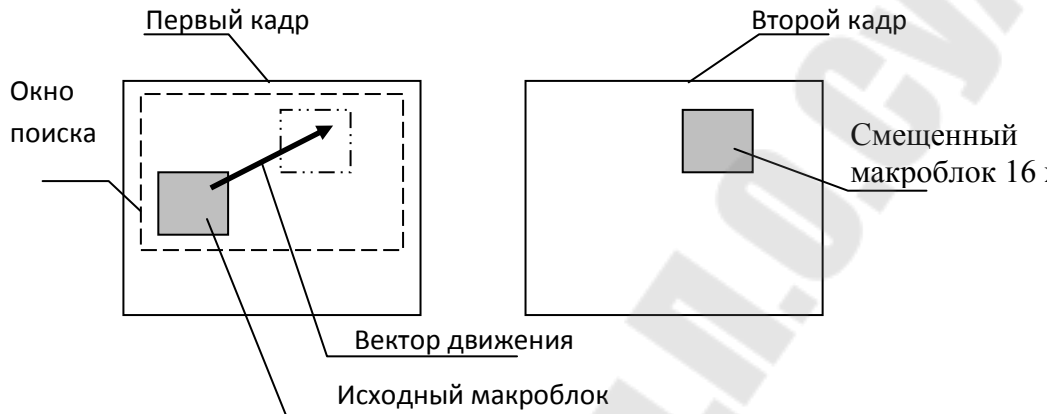


Рисунок 3.19 - Иллюстрация принципа компенсации движения

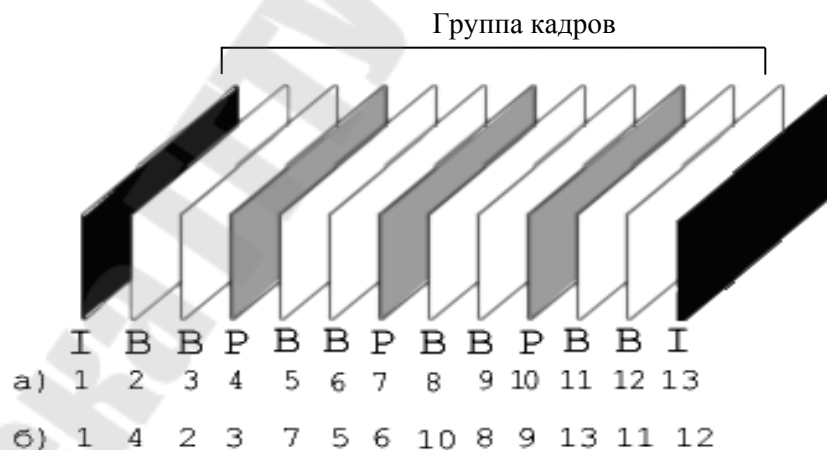


Рисунок 3.20 - Порядок обработки кадров в группе кадров:  
 а – порядок передачи и демонстрации кадров,  
 б – порядок декодирования кадров

Для корректного восстановления видеопотока последовательность декодирования кадров (рис.3.20, б) должна отличаться от последовательности их передачи и показа зрителю (рис.3.20, а). Так как внутри группы кадров, состоящей обычно из 12 кадров, каждый *B* - кадр восстанавливается по окружающим его *P*-кадрам (в начале и конце группы – по *I* и *P*), а в свою очередь каждый *P* - кадр - по пре-

дыдущему  $P$  - (или  $I$  -) кадру.  $I$  - кадры могут быть восстановлены независимо от других, они являются опорными для всех  $P$  - и  $B$  - кадров группы. Соответственно у  $I$  наименьшая степень компрессии, а у  $B$  - наибольшая. По размеру типичный  $P$  - кадр составляет  $1/3$  от  $I$ , а  $B$  –  $1/8$ . Два последовательных  $B$  кадра, формирующиеся по одному алгоритму и использующие одни и те же опорные кадры, не являются одинаковыми, так как представляют разные моменты времени видео-показа.

*Сжатие информации после компенсации движения.* Вектор коэффициентов ДКП сжимается с помощью алгоритма кодирования длин серий КДС (RLE – Run Length Encoding). При этом получаются пары типа (“пропустить”, “число”), где “пропустить” является счетчиком пропускаемых нулей, а “число” – значение, которое необходимо поставить в следующую ячейку. Так, вектор (42, 3, 0, 0, 0, -2, 0, 0, 0, 0, 1, ...) будет свернут в пары (0, 42) (0, 3) (3, -2) (4, 1) ... . Для обозначения конца вектора применяется маркер EOB (End Of Block).

Последний из алгоритмов сокращения избыточности связан с кодами переменной длительности. При этом те коэффициенты ДКП, которые повторяются наиболее часто, кодируются короткими кодовыми комбинациями, а редкие значения коэффициентов – более длинными. В MPEG применяется код Хаффмана с фиксированными таблицами, определенными в стандарте.

*Управление скоростью и качеством потока.* Управление степенью сжатия осуществляется с помощью изменения степени квантования. Для реализации эффективного управления применяется двухпроходное кодирование, при котором каждый кадр просматривается дважды. Во время первого прохода анализируется сложность изображения, чтобы обнаружить момент смены плана, выбрать тип кадра и задать матрицу квантования для наилучшего кодирования данного изображения. При втором проходе создается бинарный поток и выполняется оптимизация распределения бит с учетом заданной скорости потока на выходе компрессора.

Задавая матрицу квантования с большими коэффициентами, можно получить больше нулей и, следовательно, большую степень сжатия. Это необходимо в тех случаях, когда при кодировании сложной сцены со слабокоррелированными кадрами резко возрастает исходящий поток и переполняется буфер на выходе компрессора. Тогда компрессор увеличивает коэффициент квантования, тем самым уменьшая количество обрабатываемых частотных коэффициентов

изображения и соответственно увеличивая степень сжатия. Это делается для того, чтобы у зрителя не возникал неприятный с точки зрения визуального восприятия эффект «зависания» кадров. В стандарте MPEG задается минимальное значение объема памяти для буфера декодера, которое является максимальным для буфера кодера. Это сделано для уменьшения ошибок восстановления потока, вызванных разностями скоростей кодера (компрессора) и декодера (декомпрессора).



## 4. ЛАБОРАТОРНАЯ РАБОТА №1

### Моделирование дискретных сигналов в Matlab и Simulink

**Цель работы:** освоение приемов моделирования дискретных сигналов в средах Matlab и Simulink

#### 4.1. Теоретические сведения

Подавая электрический сигнал с выхода микрофона на вход звуковой платы компьютера, полезно представлять себе, как аналоговый сигнал преобразуется в дискретный, и как затем дискретный сигнал преобразуется в последовательность чисел. В данном разделе мы рассмотрим первый этап – преобразование аналогового сигнала в дискретный. Такое преобразование принято называть «дискретизацией».

Возможные варианты сигналов показаны на рис. 0. Сигнал, изображенный на рис.4.1.а, будем называть *исходным аналоговым*. На рис. 4.1.б представлена дискретная версия исходного сигнала, обычно именуемая *данными, оцифрованными естественным способом*, или *данными с амплитудно-импульсной модуляцией* (pulse amplitude modulation — PAM). Данные на рис.5.1.б еще несовместимы с цифровой системой, поскольку амплитуда каждой естественной выборки все еще может принимать бесконечное множество возможных значений, а цифровая система работает с конечным набором значений. На рис. 4.1.в и рис.4.1.г показано представление исходного сигнала такими дискретными импульсами, вершина которых плоская. Если значения импульсов образуют несчетное множество, они называются *дискретными отсчетами*. Далее эти импульсы можно подать на устройство квантования, преобразующее импульсы так, что их значения образуют счетное множество - такие импульсы называются *квантованными отсчетами*. Данные в таком формате совместимы с цифровой системой.

Импульсы рис.4.1.г отличаются от импульсов рис.4.1.в тем, что полностью заполняют промежуток между моментами обновления значения сигнала. Такой способ дискретизации, именуемый «выборка-хранение» [23], наиболее эффективен с точки зрения помехоустойчивости.

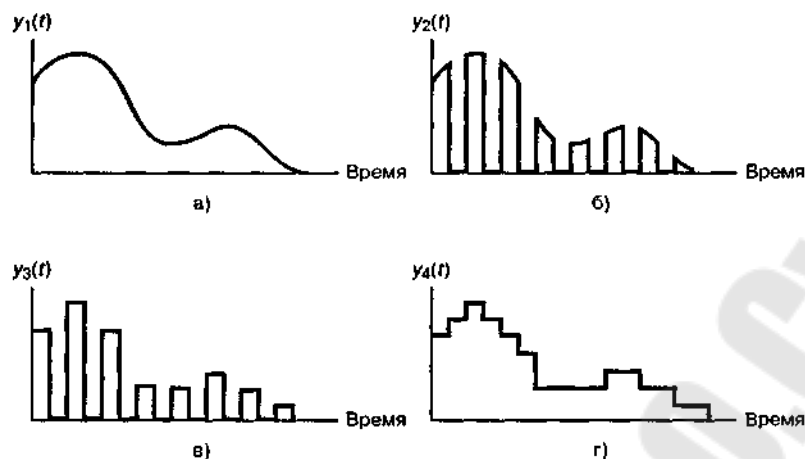


Рисунок 4.1 - Исходные данные в системе координат "время-амплитуда":

а) исходный аналоговый сигнал; б) данные в естественной дискретизации; в) квантованные выборки; г) выборка-хранение

### Моделирование дискретных сигналов в Matlab

Генерировать сигналы в **Matlab** можно тремя способами:

- в диалоговом режиме, с помощью последовательности команд в командном окне;
- в автоматическом режиме, путем создания и запуска на выполнение m-скрипта;
- в автоматическом режиме, путем создания и вызова m-функции.

**Генерирование сигналов в диалоговом режиме.** Этот способ наиболее трудоемок, поскольку требует каждую команду набирать с клавиатуры в командном окне. Чтобы повысить производительность труда, можно всю последовательность команд предварительно набрать в любом текстовом редакторе (обычно это **Notebook** или **Word**), а затем, скопировав текст в буферную память (**Clipboard**), вставить его в командное окно. Недостаток этого способа в том, что необходимо одновременно держать активными две программы – **Matlab** и текстовый редактор. Достоинство данного способа проявляется тогда, когда работу в **Matlab** производят, следуя некоей инструкции, в которой теоретические сведения чередуются с практическими заданиями в виде фрагментов текстов m-скриптов. Такой стиль работы типичен, например, при проведении лабораторных работ [1].

Например, так выглядит в текстовом редакторе последовательность команд генерирования  $N$  отсчетов тонального сигнала ампли-

тудой  $A$ , частотой  $f_0$ , начальной фазой  $Fi_0$ , с частотой дискретизации  $fs$ :

```
% гармонический сигнал
A=1; f0=100; Fi0=pi/2; fs=1000; N=20; % параметры сигнала
t=(0:N-1)/fs; % моменты времени
s=A*sin(2*pi*f0*t+Fi0); % вычисление отсчетов сигнала
plot(t,s) % вывод графика
title('Гармонический сигнал') % заголовок
xlabel('Время, с'); ylabel('Уровень'); % надписи вдоль осей
grid on % координатная сетка
```

Полученный график отображается в специальном окне с надписью Figure #1 (если это первый строящийся график). График удобно сохранять путем экспорта в экономном формате \*.jpg (рис.4.2).

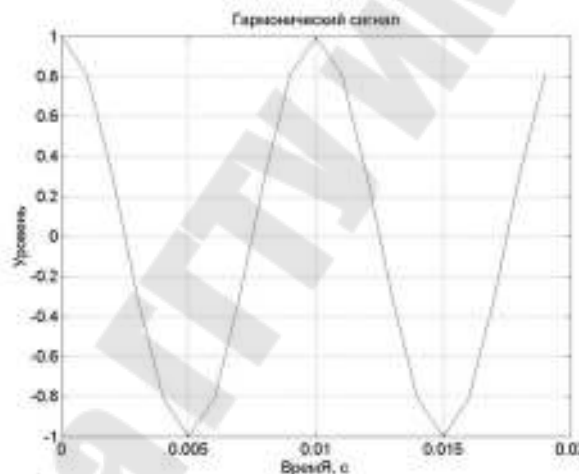


Рисунок 4.2 – Гармонический сигнал

**Примечание:** при использовании символов кириллицы в тексте команд (комментарии, заголовки и т.п.) следует учитывать особенности «отношения» каждой конкретной версии Matlab к кириллице. Так, в Matlab версии 6.1 нельзя употреблять строчную букву «я» - вместо нее следует писать прописную букву «Я». Именно по этой причине в тексте на рис.1 вместо «Время» получилось «ВремЯ». Впрочем, эту надпись можно отредактировать (кнопка со стрелкой Edit Plot в графическом окне) перед тем, как сохранять рисунок на диске.

**Генерирование сигналов путем создания m-скрипта.** Данный способ отличается тем, что все команды набираются в специальном окне редактора m-файлов (рис.4.3).

```

1 % гармонический сигнал
2 A=1; f0=100; Fi0=pi/2; fs=1000; N=20; % параметры сигнала
3 t=(0:N-1)/fs; % моменты времени
4 s=A*sin(2*pi*f0*t+Fi0); % вычисление отсчетов сигнала
5 plot(t,s) % вывод графика
6 title('Гармонический сигнал') % заголовок
7 xlabel('Время, с'); ylabel('Уровень'); % надписи вдоль осей
8 grid on % координатная сетка

```

Рисунок 4.3 – Пример использования окна редактора

Данный способ хорош тем, что вместо сторонних программных продуктов используется собственный инструментарий Matlab, специализированный для написания и отладки m-скриптов.

**Генерирование сигналов путем создания m-функции.** Данный способ отличается тем, что входные данные записываются как аргумент некоей функции  $y = f(x)$ , а выходные – как значение этой функции. Удобство в том, что символьные обозначения данных могут отличаться от обозначений, используемых в теле функции. Более того, числовые значения входных данных можно просто задавать в наименовании вызываемой функции. Последнее обстоятельство продемонстрируем на примере.

Создадим подпрограмму - m-скрипт **ton.m** вида:

```

% скрипт ton
s=A*sin(2*pi*f0*t+Fi0); % вычисление отсчетов сигнала

```

Команду выполнения этого скрипта нужно «окружить» командами подготовки входных данных и вывода выходных данных:

```

A=1; f0=100; Fi0=pi/2; fs=1000; N=20; % параметры сигнала
t=(0:N-1)/fs; % моменты времени
ton; % вычисление отсчетов сигнала
plot(t,s) % вывод графика
title('Гармонический сигнал') % заголовок
xlabel('Время, с'); ylabel('Уровень'); % надписи вдоль осей
grid on % координатная сетка

```

Очевидно, обозначения входных и выходных данных вызывающей программы должны совпадать с обозначениями соответствующих данных вызываемой подпрограммы.

Теперь поступим по-иному – напишем и сохраним m-функцию под именем **ton\_sig.m**:

```
%-----функция ton_sig.m -----
% [s,t]=ton_sig(B,f1,Fi1,Fs,N1)
%-----
% генерирование гармонического сигнала
%  $y = B * \sin(2*\pi*f1*x + Fi1)$ ,
% B - амплитуда;
% N1 - количество отсчетов сигнала;
% f1 - частота;
% Fs - частота дискретизации;
% Fi1 - начальная фаза сигнала
%-----
function [y,x] = ton_sig( B, f1, Fi1, Fs, N1 )
%-----
x = (0:N1-1)/Fs; % моменты времени
y = B * sin( 2*pi*f1*x + Fi1 );
%----- конец функции ton_sig.m -----
```

Теперь m-скрипт генерирования того же отрезка косинусоиды будет выглядеть так:

```
% гармонический сигнал
[s,t]=ton_sig(1,100,pi/2,1000,20) % вычисление отсчетов сигнала
ла
plot(t,s) % вывод графика
title('Гармонический сигнал') % заголовок
xlabel('Время, с'); ylabel('Уровень'); % надписи вдоль осей
grid on % координатная сетка
```

Как видим, теперь числовые значения входных данных задаются как аргументы m-функции **ton\_sig.m**. Выходные данные функции используются для построения графика.

Очевидно, применение m-функций выгодно тогда, когда алгоритм формирования значений функции достаточно сложный: содержится много команд и обращений к разнообразным библиотечным функциям с непростым синтаксисом.

Очевиден и недостаток m-функций – необходимо помнить их синтаксис. Впрочем, получить нужную информацию можно, если в командном окне задать команду **help**:

```
>> help ton_sig
```

В результате на мониторе отобразится комментарий, с которого начинается m-функция. Для приведенного выше примера текст помощи имеет следующий вид:

```
%-----функция ton_sig.m -----  
% [s,t]=ton_sig(B,f1,Fi1,Fs,N1)  
%-----  
% генерирование гармонического сигнала  
%  $y = B * \sin(2*\pi*f1*x + Fi1)$ ,  
% B - амплитуда;  
% N1 - количество отсчетов сигнала;  
% f1 - частота;  
% Fs - частота дискретизации;  
% Fi1 - начальная фаза сигнала  
%-----
```

Таким образом, очевиден вывод: **очень важно при программировании m-функций снабжать их качественным и подробным комментарием.**

### Моделирование дискретных сигналов в Simulink

Генерирование сигналов в **Simulink**, естественно, имеет свои особенности. Рассмотрим их.

Возьмем из библиотеки блоков **Simulink** два блока: **Sine Wave** (из раздела **Sources**) и **Scope** (из раздела **Sinks**). Соединив их, получим следующую схему (рис.4.4).



Рисунок 4.4 – Схема исследования

Затем двойным щелчком по блоку осциллоскопа активизируем окно, имитирующее экран осциллоскопа, и запустим модель (кнопка **Start simulation**). В результате получим изображение отрезка синусоиды (рис.4.5).

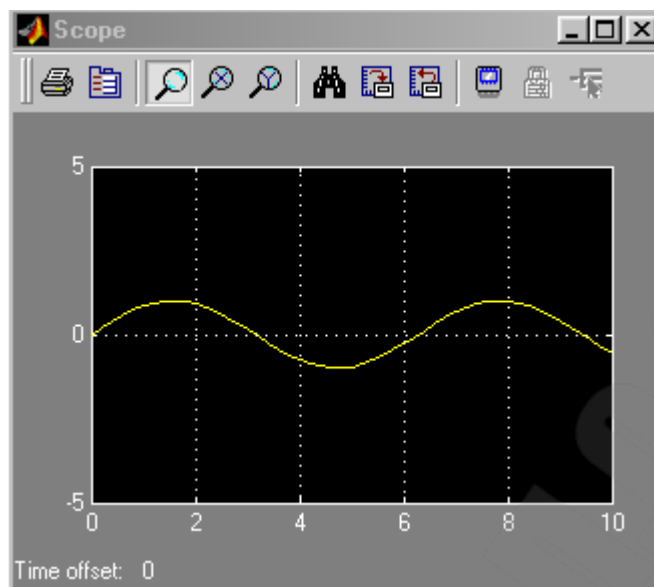


Рисунок 4.5 – Изображение синусоиды

Как видим, генерировать гармонический сигнал в среде Simulink даже проще, чем в среде Matlab. Однако это первое впечатление весьма обманчиво. Действительно, ведь важно еще уметь управлять параметрами гармонического сигнала. То, что амплитуда гармонического сигнала оказалась равной единице – нам просто «повезло». Действительно, по умолчанию амплитуда генерируемого сигнала принята равной единице. Однако частотой, начальной фазой и длительностью сигнала мы пока не управляем.

Дважды щелкнем по блоку **Sine Wave** – в результате появится окно настроек параметров (рис.5). Щелкнув по кнопке **Help**, получим инструкцию по данному блоку, сущность которой сводится вкратце к тому, что в данном блоке выполняется операция

$$y = \textit{Amplitude} \times \sin(\textit{frequency} \times \textit{time} + \textit{phase}) + \textit{bias}$$

Из приведенной формулы и надписей на рис.4.6 становится понятным смысл четырех переменных: амплитуды, угловой частоты, начальной фазы и постоянной составляющей. Остается пока зашифрованным смысл переменной “время”.

Останавливаясь на этом важном вопросе, отметим различие понятий “время” и “модельное время”. Так, генерирование отрезка сигнала длительностью 1 с (модельное время) может длиться значительно более короткий промежуток времени, например, 0.1 с (реальное время). Скорость генерирования зависит от объема вычислений, быстродействия компьютера, от выбранного “решателя”, т.е. алгоритма модели-

рования, и т.д. Кстати, вполне возможен обратный эффект - для сложного алгоритма процедура моделирования отрезка сигнала длительностью 0.1 с может растянуться на несколько секунд.

Сигнал может генерироваться двух типов: непрерывный **time-based** и дискретный **sample-based**. Для моделирования работы непрерывных систем рекомендуют использовать непрерывный тип **time-based**, а для моделирования работы дискретных систем – дискретный тип **sample-based** [24].

Если установлен тип **time-based**, тогда параметр **Sample time** может принимать значения:

- 0 (по умолчанию) – блок работает в непрерывном режиме;
- >0 - блок работает в дискретном режиме;
- -1 – блок наследует тот же режим, что и принимающий блок.

Как указывается в **Help**, работа в непрерывном режиме может приводить к большим погрешностям генерации на больших промежутках модельного времени.

Работа в дискретном режиме заставляет блок вести себя так, как если бы к выходу непрерывного генератора был присоединен блок **Zero-Order Hold**. Действительно, собрав две схемы (рис.4.7) и задав в обоих случаях значение параметра **Sample time**, равное 0.5 (окно настройки блока **Zero-Order Hold** показано на рис.4.8), получаем идентичные результаты (рис.4.9).

Таким образом, блок **Zero-Order Hold** можно трактовать как “дискретизатор”, т.е. часть АЦП, ответственную за дискретизацию сигнала. Иногда блок **Zero-Order Hold** именуют АЦП [25]. По нашему мнению, это не корректно, поскольку дискретизированный сигнал в “подлинном” АЦП подвергается еще и квантованию по уровню. В блоке **Zero-Order Hold**, однако, квантование не производится.

Несколько слов о построении графиков. Помимо блока **Scope**, график можно построить и с помощью блока **X-Y-Graf**, на верхний вход X которого нужно подать последовательность моментов времени с помощью блока **Clock** (часы), а на нижний вход Y – значения генерируемого сигнала (рис.9).



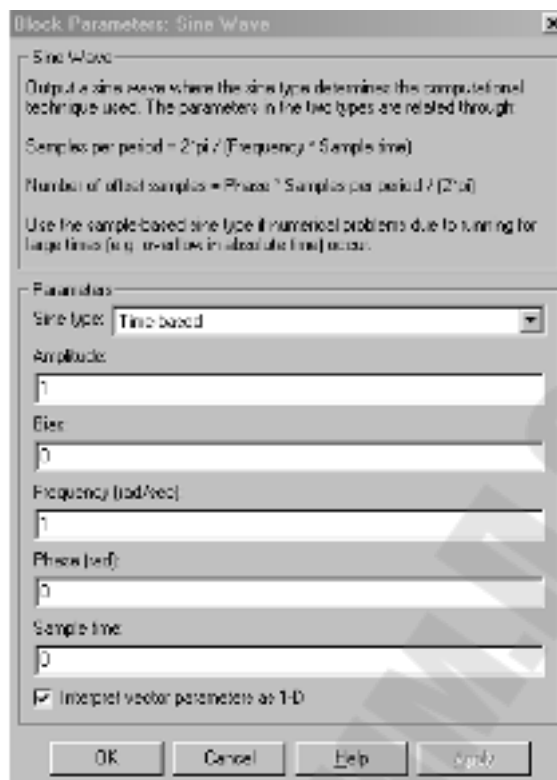


Рисунок 4.6 – Окно настройки генератора синусоидальных колебаний

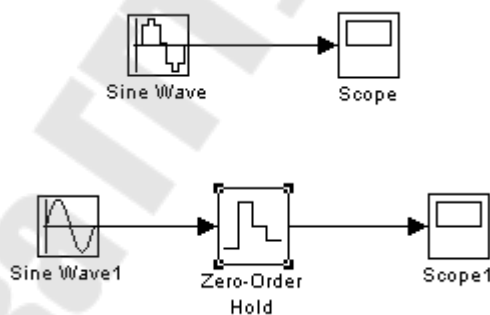


Рисунок 4.7 – Схема исследований

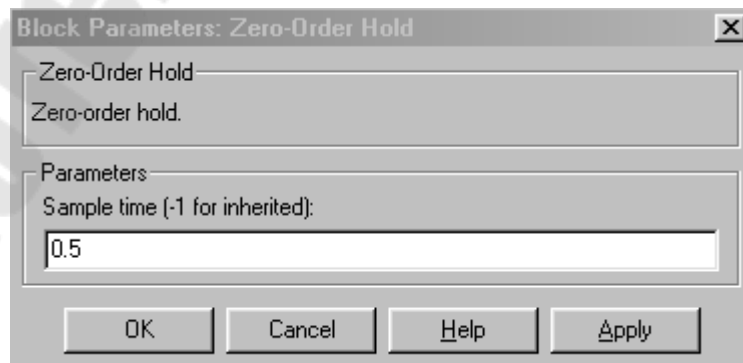


Рисунок 4.8 – Окно настройки параметров

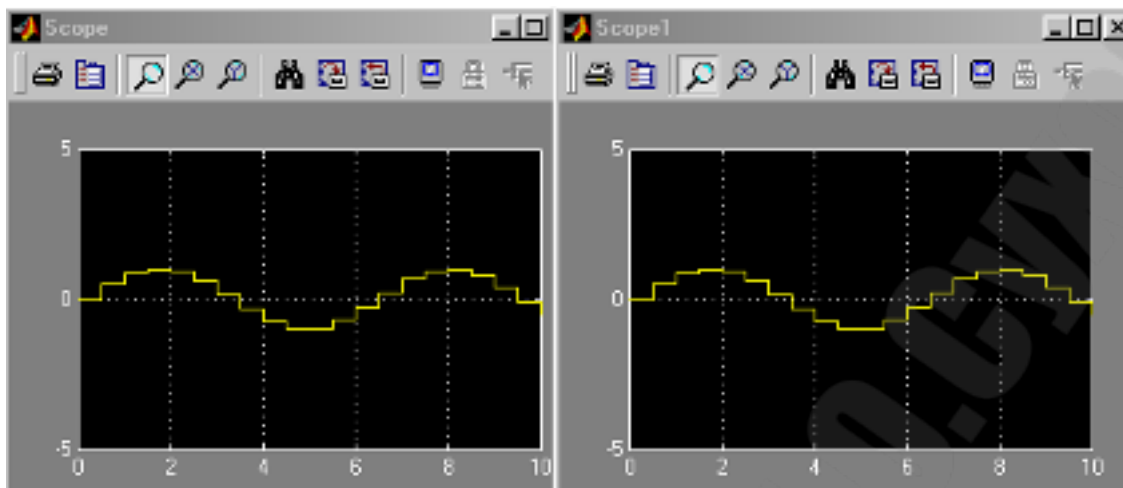


Рисунок 4.9 – Результаты исследований

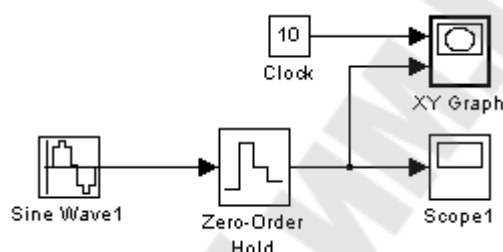


Рисунок 4.10 – Пример использования компонента X-Y-Graf

В результате предварительно настроенный (в соответствующем окне настройки задаются граничные значения аргумента и функции, а также указывается значение параметра **Sample time**) графопостроитель выдаст показанный на рис.10 график, если для блока **X-Y-Graf** задано **Sample time=-1** (т.е. период дискретизации наследуется).

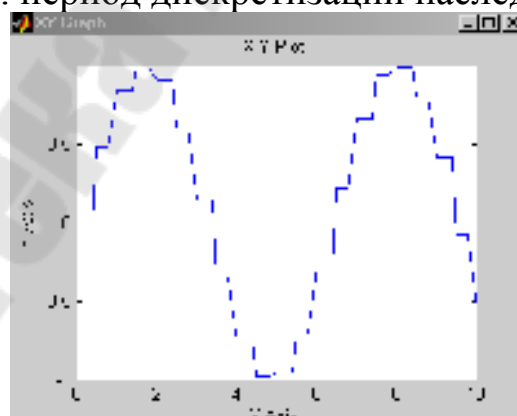


Рисунок 4.11 – Результат использования компонента X-Y-Graf

График будет несколько иным (рис.4.12), если для блока **X-Y-Graf** задано **Sample time=0.5**.

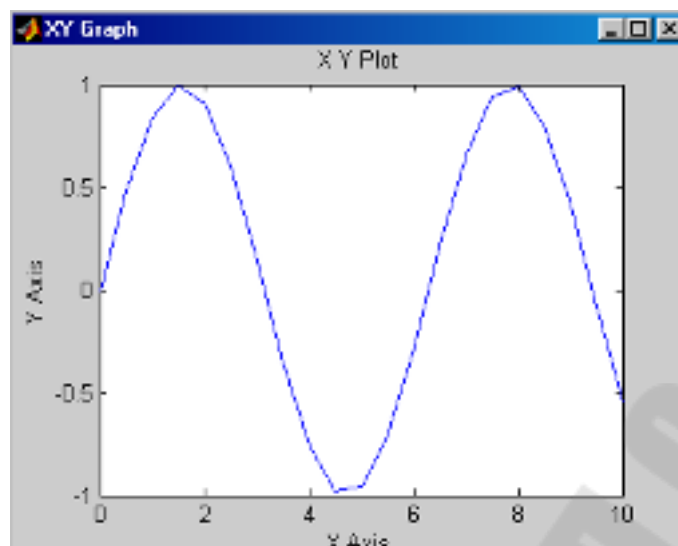


Рисунок 4.12 – Результат использования компонента X-Y-Graf с параметром **Sample time=0.5**.

Еще об одном способе построения графиков. Массивы отсчетов моментов времени и соответствующих значений сигнала можно с помощью блока **To Workspace** экспортировать из среды **Simulink** в среду **Matlab** (рис.4.13).

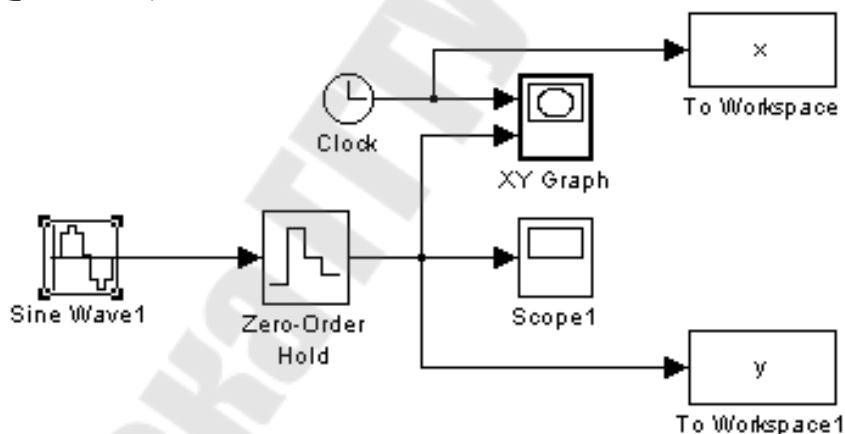


Рисунок 4.13 – Вариант экспорта данных в среду Matlab  
При этом, как показывает практика, лучше всего задать формат **array** для экспортируемых данных (рис. 4.14).



Рисунок 4.14 – Параметры блока

Дальнейшее построение графика в среде **Matlab** с помощью команды **plot(x,y)** не представляет никакого труда (рис.4.15).

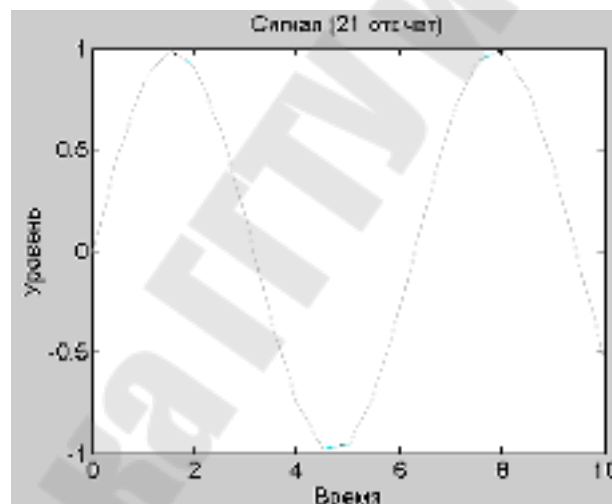


Рисунок 4.15 – Пример построения зависимости по сохраненным Отсчетам

## 4.2. Ход работы

Смоделировать в среде Matlab сигнал  $s_2(t)$  на выходе дискретизатора (частота дискретизации  $F_s$ ), если на вход дискретизатора подается сигнал:

$$s_1(t) = A_1 \cos(2\pi f_{01}t + \varphi_{01}) + A_2 \cos(2\pi f_{02}t + \varphi_{02}), \quad 0 \leq t \leq T.$$

Варианты значений параметров сигнала  $s_1(t)$  приведены в табл.4.1.

## Варианты заданий

Парам. \ Вар.	1	2	3	4	5	6	7	8
$A_1$	1	1	2	1	1	1	1	2
$A_2$	1	2	1	1	1	2	2	1
$f_{01}, \text{Гц}$	100	100	100	100	100	100	100	100
$f_{02}, \text{Гц}$	200	200	200	200	200	200	200	200
$\varphi_{01}, \text{рад}$	0	0	0	0	$\pi$	0	$\pi$	0
$\varphi_{02}, \text{рад}$	0	0	0	$\pi$	0	$\pi$	0	$\pi$

Частоту дискретизации  $F_s$  задать двумя способами:

а) исходя из инженерной версии теоремы Найквиста-Котельникова;

б) увеличив выбранное по п.а) значение в 5 раз.

Длительность  $T$  выбрать так, чтобы на ней уложилось два периода.

1. Смоделировать сигнал  $s_2(t)$  в среде Matlab
2. Смоделировать сигнал  $s_2(t)$  в среде Simulink.
3. Сделать общие выводы по работе, сравнив между собой моделирование в среде и моделирование в среде Simulink.

### 4.3. Содержимое отчета

В отчете описать перечень команд Matlab, с помощью которых происходит вычисление сигнала  $s_2(t)$ , представить результаты моделирования: график сигнала  $s_2(t)$  и составляющих его гармонических компонентов, блок-схему моделирования.

#### 4.4. Контрольные вопросы

1. Что такое «инженерная версия теоремы Найквиста-Котельникова»?
2. Что собой представляет способ дискретизации, именуемый «выборка-хранение»?
3. Как реализовать способ дискретизации, именуемый «выборка-хранение», в среде Matlab и в среде Simulink?
4. Варианты передачи информации в среде Matlab.
5. Варианты передачи информации в среде Simulink.
6. Представление аналогового сигнала в дискретном виде.
7. Преимущества представления аналогового сигнала в дискретной форме.
8. Недостатки представления аналогового сигнала в дискретной форме.

## 5. ЛАБОРАТОРНАЯ РАБОТА №2

### Моделирование квантования в Matlab и Simulink

**Цель работы:** освоение приемов квантования дискретных сигналов в средах Matlab и Simulink

#### 5.1. Теоретические сведения

Квантование - это процесс перехода от непрерывного множества значений сигнала к дискретному множеству, объем которого равен количеству уровней квантования.

**Равномерное (линейное) квантование.** Рассмотрим рис.5.1, на котором изображены  $L$  уровней квантования аналогового сигнала с полным диапазоном напряжений, равным  $V_{pp} = V_p - (-V_p) = 2V_p$  вольт. Квантованные импульсы могут иметь положительные и отрицательные значения. Шаг между уровнями квантования, называемый *интервалом квантования*, составляет  $q$  вольт. Если уровни квантования равномерно распределены по всему диапазону, устройство квантования именуется *равномерным*, или *линейным*. Каждое дискретное значение аналогового сигнала аппроксимируется квантованным импульсом: аппроксимация дает ошибку, не превышающую  $q/2$  в положительном направлении или  $-q/2$  в отрицательном. Таким образом, ухудшение сигнала вследствие квантования ограничено половиной квантового интервала,  $\pm q/2$  вольт.

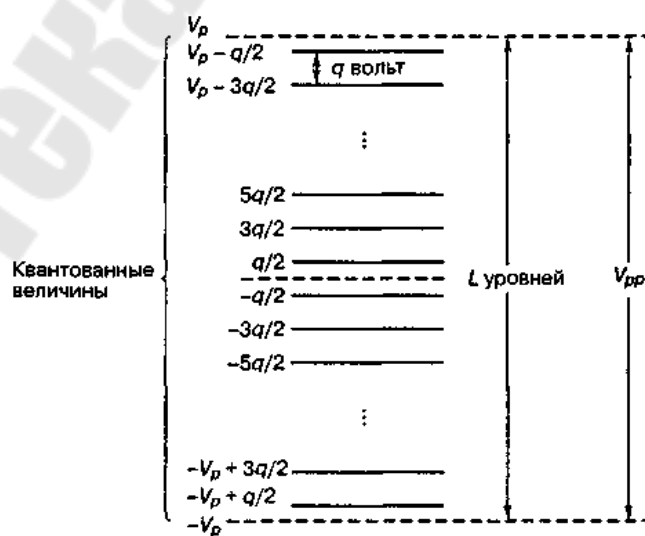


Рисунок 5.1 - Уровни квантования

Хорошим критерием качества равномерного устройства квантования является его дисперсия (средний квадрат ошибки  $e_i = \tilde{x}_i - x_i$  при подразумеваемом нулевом среднем, где  $\tilde{x}_i$  - квантованный отсчет,  $x_i$  - квантуемый отсчет). Если считать, что ошибка квантования  $e$  равномерно распределена в пределах интервала квантования шириной  $q$  (т. е. принимает все возможные значения с равной вероятностью), то дисперсия ошибок для устройства квантования составляет

$$\sigma^2 = \int_{-q/2}^{q/2} e^2 p(e) de = \int_{-q/2}^{q/2} e^2 \frac{1}{q} de = \frac{q^2}{12} \quad (5.1)$$

где  $p(e) = 1/q$  при  $-q/2 < e < q/2$  и 0 для др.  $q$  - плотность вероятностей ошибки квантования. Дисперсия  $\sigma^2$  соответствует *средней мощности шума квантования*. Пиковую мощность аналогового сигнала (нормированную на 1 Ом) можно выразить как

$$V_P^2 = \left( \frac{V_{PP}}{2} \right)^2 = \left( \frac{q(L-1)}{2} \right)^2 \approx \frac{q^2 L^2}{2} \quad (5.2)$$

где  $L$  - число уровней квантования. Объединение выражений (5.1) и (5.2) дает отношение *пиковой* мощности сигнала к *средней* мощности квантового шума  $(S/N)_q$ :

$$(S/N)_q = \frac{q^2 L^2 / 4}{q^2 / 12} = 3L^2, \quad (5.3)$$

где  $N = \sigma^2$  — средняя мощность шума квантования. Очевидно, что отношение  $(S/N)_q$  квадратично растет с числом уровней квантования. В пределе ( $L \rightarrow \infty$ ) сигнал становится аналоговым (бесконечное число уровней квантования и нулевой шум квантования). Отметим, что для случайных сигналов в параметр  $(S/N)_q$  входит не максимальная, а средняя мощность сигнала. В этом случае для получения средней мощности сигнала требуется знать функцию плотности вероятности.

**Неравномерное (нелинейное) квантование.** Линейные устройства квантования легко реализовать и легко понять — в этом их очевидное достоинство. Вместе с тем, выбор параметров устройств равномерного квантования не предполагает никаких знаний о статистике амплитуд и корреляционных свойствах входного сигнала.

Нелинейные устройства квантования, обеспечивающие неравномерное квантование, применяются тогда, когда возникает желание



учесть статистику амплитуд и корреляционные свойства входного сигнала.

Существуют приложения, для которых равномерные устройства квантования являются наилучшими. Это – обработка музыкальных сигналов, обработка изображений, контроль процессов и ряд других. Для некоторых иных приложений более приемлемы неравномерные квантующие устройства. Важнейшим примером такого рода является обработка речевых сигналов в системах связи.

Человеческая речь характеризуется уникальными статистическими свойствами, одно из которых проиллюстрировано на рис. 5.2.

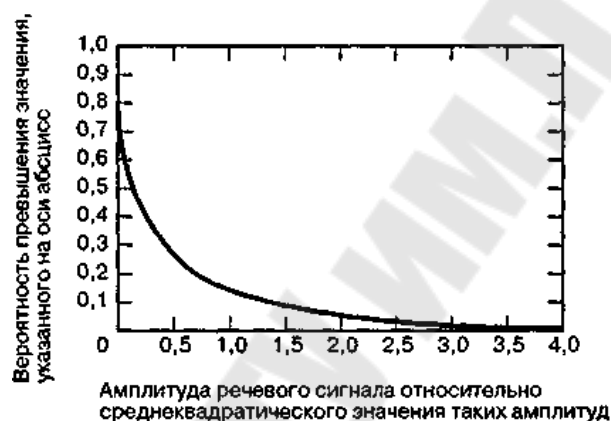


Рисунок 5.1 - Статистическое распределение амплитуд речи одного лица

На оси абсцисс отложены амплитуды речевого сигнала, нормированные по среднеквадратическому значению амплитуд, а на оси ординат — вероятность. Для большинства каналов речевой связи доминируют очень низкие уровни сигналов: 50% времени напряжение, характеризующее энергию обнаруженной речи, составляет менее четверти среднеквадратического значения. Значения с большими амплитудами встречаются относительно редко: только 15% времени напряжение превышает среднеквадратическое значение.

При передаче речи система с равномерным квантованием будет неэкономной - многие уровни квантования будут использоваться довольно редко. Кроме того, в такой системе шум квантования будет одинаковым для всех амплитуд сигнала. Следовательно, при таком квантовании отношение сигнал/шум будет хуже для сигналов низких уровней, чем для сигналов высоких уровней.

Неравномерное квантование может обеспечить лучшее квантование слабых сигналов и грубое квантование сильных сигналов. Значит, в этом случае шум квантования может быть пропорциональным сигналу. Результатом является повышение общего отношения сигнал/шум — уменьшение шума для доминирующих слабых сигналов за счет повышения шума для редко встречающихся сильных сигналов. На рис. 5.3 сравнивается квантование слабого и сильного сигналов при равномерном и неравномерном квантовании. Ступенчатые сигналы представляют собой аппроксимации аналоговых сигналов (после введения искажения вследствие квантования).

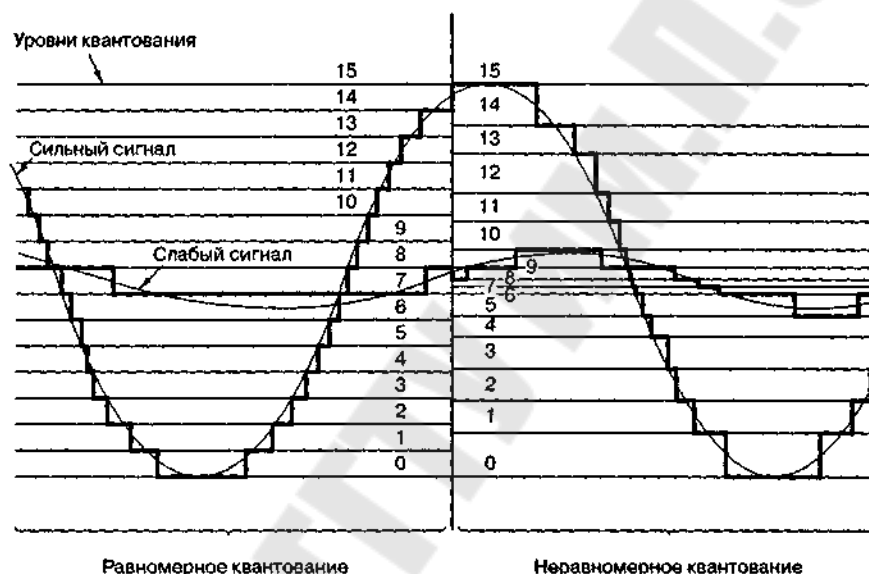


Рисунок 5.3 - Равномерное и неравномерное квантование сигналов

Одним из способов получения неравномерного квантования является использование устройства с неравномерным квантованием с характеристикой, показанной на рис. 5.4.а. Гораздо чаще неравномерное квантование реализуется следующим образом: вначале исходный сигнал деформируется с помощью устройства, имеющего логарифмическую характеристику сжатия, показанную на рис. 5.4.б, а потом используется устройство квантования с равномерным шагом. Для сигналов малой амплитуды характеристика сжатия имеет более крутой фронт, чем для сигналов большой амплитуды. Следовательно, изменение данного сигнала при малых амплитудах затронет большее число равномерно размещенных уровней квантования, чем то же изменение при больших амплитудах. Характеристика сжатия эффективно меняет распределение амплитуд входного сигнала, так что на выходе системы

сжатия уже не существует превосходства сигналов *малых* амплитуд. После сжатия деформированный сигнал подается на вход равномерного (линейного) устройства квантования с характеристикой, показанной на рис. 5.4.в. После приема сигнал пропускается через устройство с характеристикой, обратной к показанной на рис. 5.4.б и называемой *расширением*, так что общая передача не является деформированной. Описанная пара этапов обработки сигнала (сжатие и расширение) в совокупности обычно именуется *командированием*.

В Северной Америке  $\mu$ -уровневая характеристика устройства сжатия описывается следующим законом:

$$y = y_{\max} \frac{\ln[1 + \mu(|x|/x_{\max})]}{\ln(1 + \mu)} \operatorname{sgn} x, \quad (5.4)$$

где

$$\operatorname{sgn} x = \begin{cases} +1 & \text{при } x \geq 0 \\ -1 & \text{при } x < 0 \end{cases}$$

$\mu$  - положительная константа,  $x$  и  $y$  - напряжения на входе и выходе, а  $x_{\max}$  и  $y_{\max}$  - максимальные амплитуды напряжений на входе и выходе.

В Европе для описания характеристики устройства сжатия используется несколько иной закон:

$$y = \begin{cases} y_{\max} \frac{A(|x|/x_{\max})}{1 + \ln A} \operatorname{sgn} x, & 0 < \frac{|x|}{x_{\max}} \leq \frac{1}{A}; \\ y_{\max} \frac{1 + \ln[A(|x|/x_{\max})]}{1 + \ln A} \operatorname{sgn} x, & \frac{1}{A} < \frac{|x|}{x_{\max}} \leq 1; \end{cases} \quad (5.5)$$

Здесь  $A$  - положительная константа, а  $x$  и  $y$  определены так же, как и в формуле (5.4).

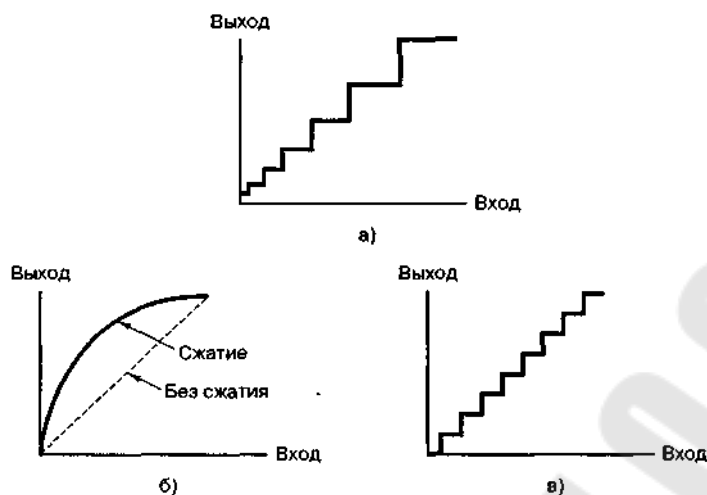


Рисунок 5.4 - Примеры характеристик: а) характеристика неравномерного устройства квантования; б) характеристика сжатия; в) характеристика равномерного устройства квантования.

Вид характеристик (5.4) и (5.5) приведен на рис.5.5.

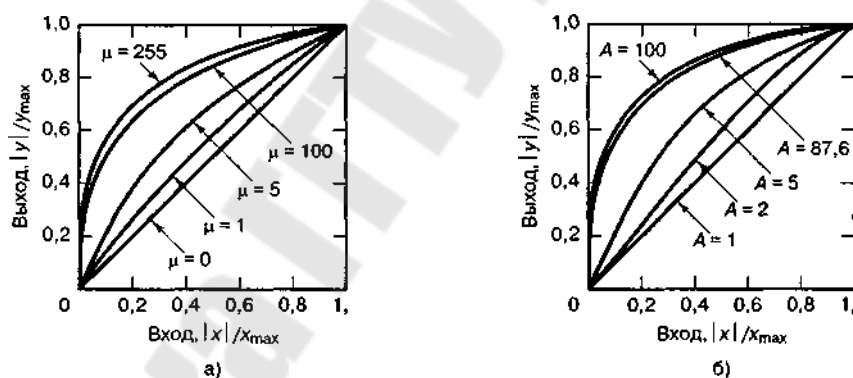


Рисунок 5.5 - Характеристики устройств сжатия: а) для различных значений  $\mu$ ; б) для различных значений  $A$

**Математическое описание процедуры квантования.** Процедуру квантования описывают с помощью вектора (размерностью  $L-1$ ) деления области определения сигнала  $P_1, P_2, \dots, P_{L-1}$  (vector partition) и вектора (размерностью  $L$ ) значений кодированного сигнала  $C_1, C_2, \dots, C_L$  (codebook). Если  $x$  - текущее значение сигнала, тогда на выход квантователя выдаются значения по правилу, приведенному в таблице 5.1:

Таблица 5.1

## Кодовая книга квантования

Вход $x$	Выход $y$
$x \leq P_1$	$C_1$
$P_1 \leq x \leq P_2$	$C_2$
$P_2 \leq x \leq P_3$	$C_3$
...	...
$x \geq P_{L-1}$	$C_L$

## Моделирование квантования в Matlab

M-функция квантования в **Matlab** имеет вид:

**[ind, xQ, D] = quantiz(x, partition, codebook),**

где **x** – вектор отсчетов сигнала; **partition** – вектор границ интервалов  $P_1, P_2, \dots, P_{L-1}$ ; **codebook** – вектор  $C_1, C_2, \dots, C_L$  значений кодовой книги; **ind** – вектор-столбец индексов (номеров) интервалов; **xQ** – вектор-строка квантованных значений сигнала; **D** – средний квадрат ошибки квантования:

$$D = \frac{1}{N} \sum_{n=1}^{N_t} (x_n - \tilde{x}_n)^2 .$$

## Пример применения m-функции:

```
% квантование гармонического (50 Гц) сигнала единичной амплитуды
% частота дискретизации 1000 Гц
i=1:20;
x=cos(2*pi*50*i/1000);
partition=[-.75 -.25 .25 .75];
codebook=[-.825 -.5 0 .5 .825];
[ind, xQ, D] = quantiz(x, partition, codebook);
plot(i,x), hold on, stem(i,xQ)
legend('Исходный сигнал','Результат квантован.')
for n=1:4
prt(n,i)=partition(n); plot(i,prt,'k')
end
for m=1:5
cdbk(m,i)=codebook(m); plot(i,cdbk,'r-')
end
figure
stem(i,ind), title('Индексы')
ind, xQ, D
```

Исходный и квантованный сигналы приведены на рис.5.6, график значений индексов приведен на рис.5.7.

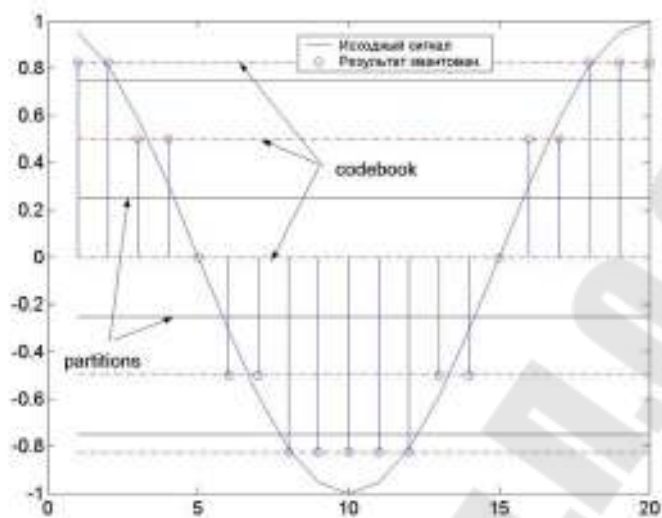


Рисунок 5.6 – Исходный и квантованный сигнал

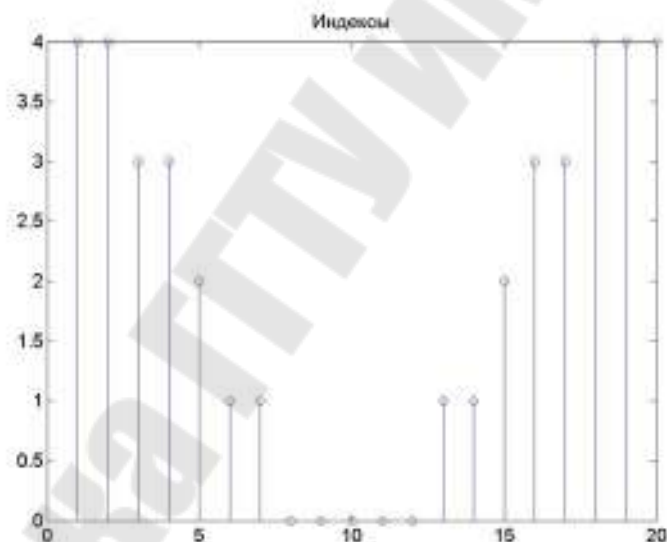


Рисунок 5.7 – График значений индексов

### Моделирование квантования в Simulink

Блок **Sampled quantizer encode** (рис.5.8) находится (для Matlab вер.6.1) в разделе Communications Blockset\Source Coding.

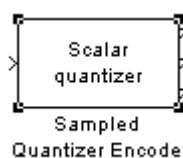


Рисунок 5.8 – Внешний вид блока Sampled quantizer encode

Данный блок имеет один вход и три выхода. На вход подается квантуемый сигнал. На выходах, сверху вниз: вектор  $ind$ , вектор  $xQ$ , вектор квадратов ошибки квантования (а не средний квадрат ошибки, как в случае  $m$ -функции `quantiz`!).

### Пример применения блока `Sampled quantizer encode`:

Собираем схему, показанную на рис.5.9.

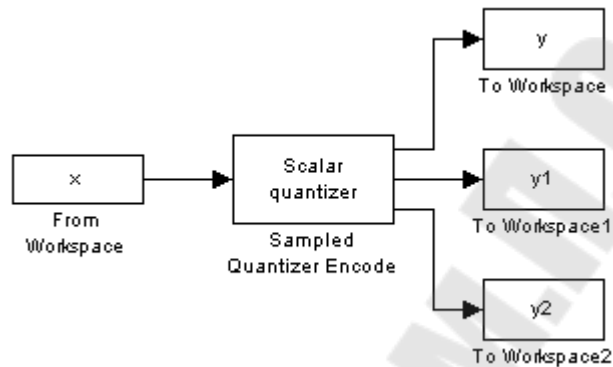


Рисунок 5.9 - Квантование сигнала блоком `Sampled quantizer encode`

### Настраиваем блок `From Workspace`:

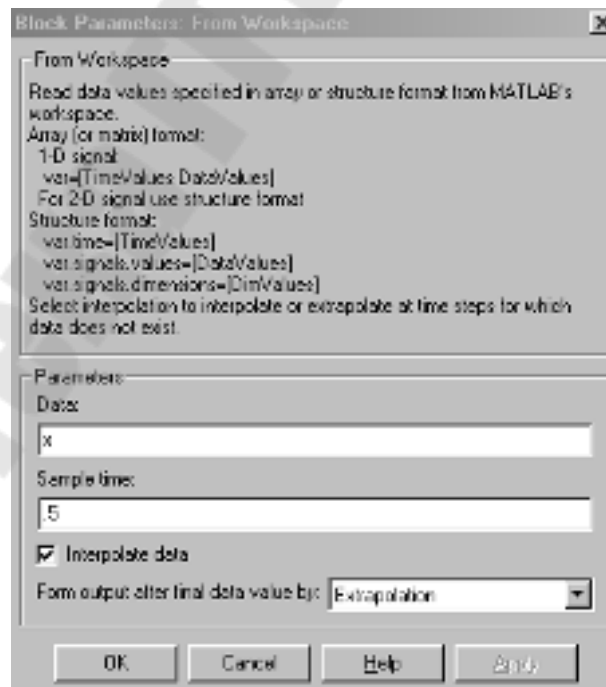


Рисунок 5.10 - Окно-маска для настройки блока `From Workspace`

### Настраиваем блок `Sampled quantizer encode`:

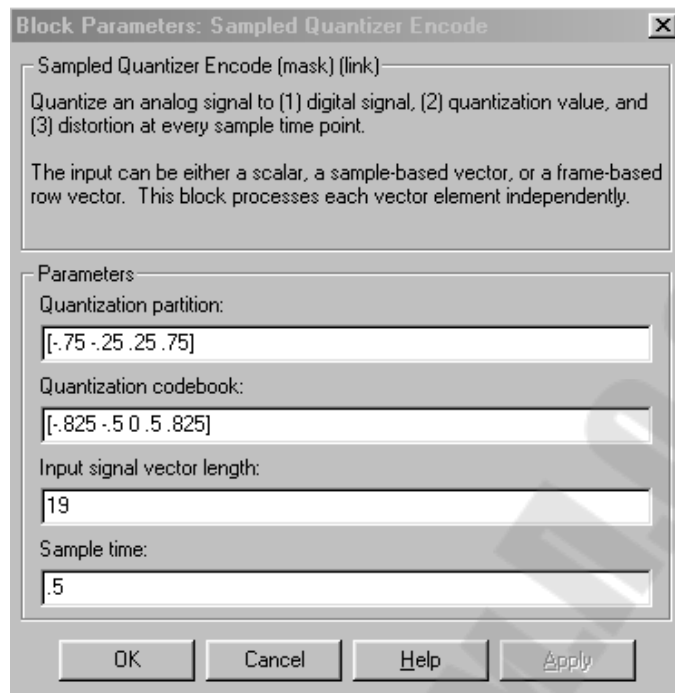


Рисунок 5.11 - Окно-маска для настройки блока Sampled quantizer encode

Заметим, что длину входного вектора берем не 20, а 19 – это необходимо, чтобы схема работала нормально (разумеется, в каждом из вариантов лабораторной работы длина входного вектора будет иной!).

Блоки **To Workspace** настраиваем следующим образом:

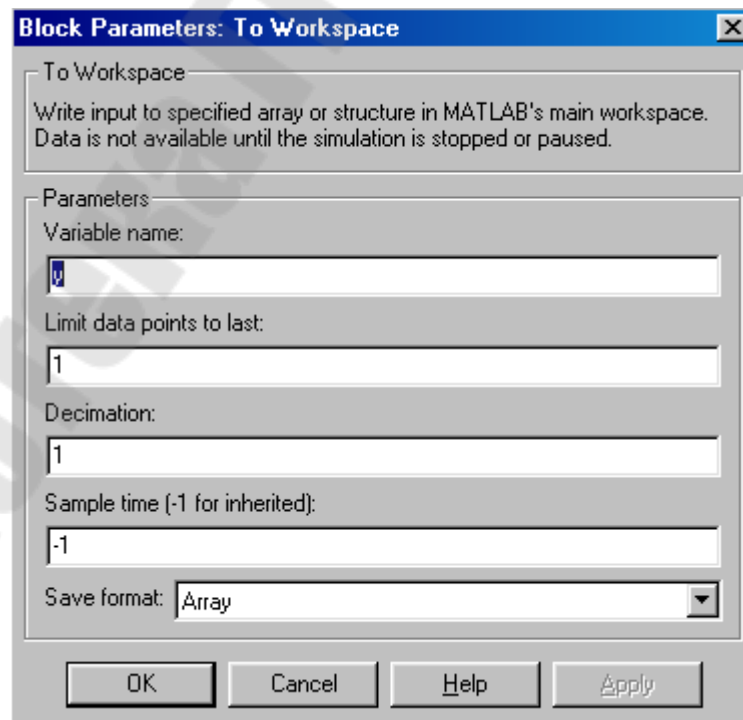


Рисунок 5.12 - Окно-маска для настройки блока To Workspace



Обратите внимание: формат результата – массив (array), а не структура (structure). Данные в формате массива удобнее анализировать, поскольку не нужно тратить время на их извлечение из структуры с помощью специальных довольно громоздких команд.

Далее в командном окне **Matlab** строим график фрагмента массива **y**:

```
stem(y)
```

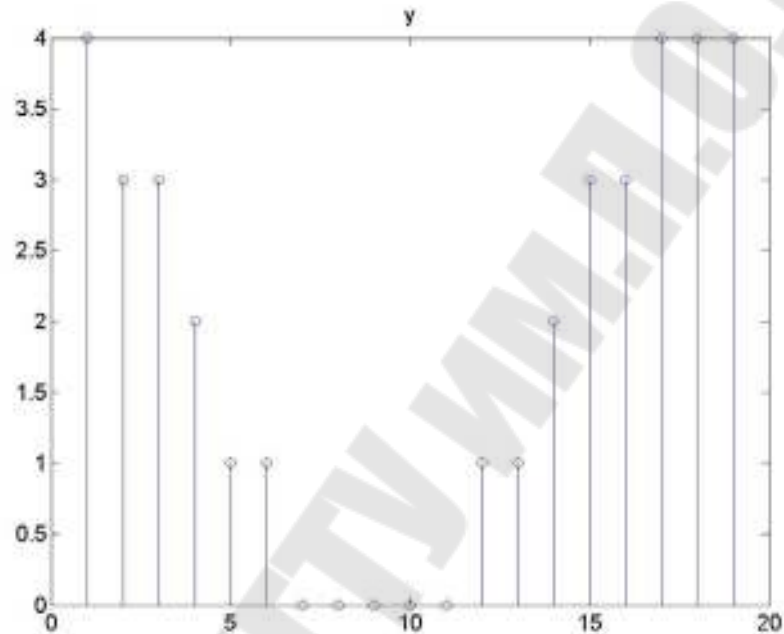


Рисунок 5.13 – График фрагмента массива

Сравнивая рис. 5.13 с рис.5.7, видим, что на рис.5.13 изображен массив индексов, но без первого элемента. Таким образом, «потерялся» один (первый) элемент. Это понятно, поскольку размер входной выборки мы задавали 19.

Аналогично строим график выходного сигнала **y1** со второго выхода (рис.5.14).

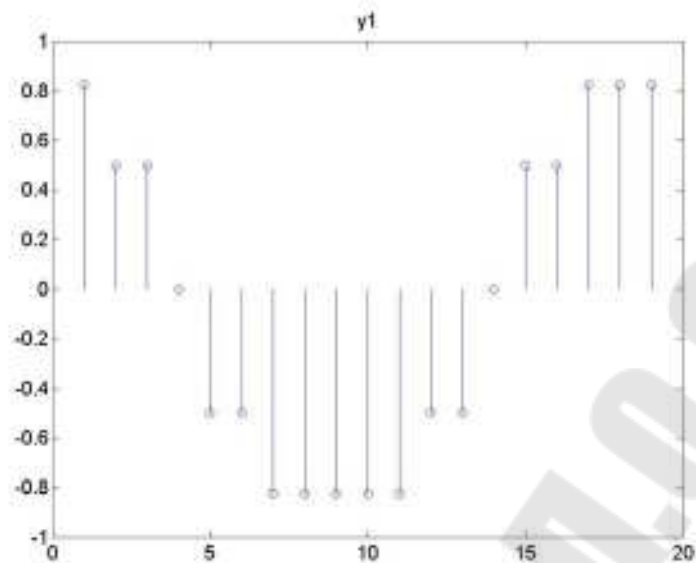


Рисунок 5.14 - График фрагмента массива

Сравнивая рис.5.14 с рис.5.6 видим, что на рис.5.14 изображен квантованный сигнал, но также без первого отсчета.

Наконец, строим график массива  $y_2$  (рис.5.15).

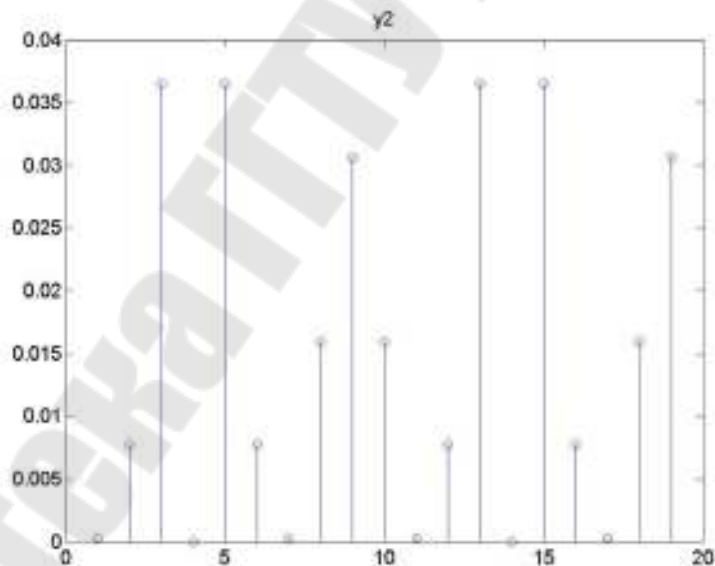


Рисунок 5.15 - График массива  $y_2$

Сопоставляя рис.5.15 с рис.5.6, заключаем, что на рис.5.15 изображен квадрат ошибки (массив), а не средний квадрат ошибки (число).

Таким образом, реализация квантования с помощью  $m$ -функции `quantiz` и с помощью блока `Sampled quantizer encode` приводит к несколько отличающимся результатам.

## Квантование реальных сигналов в Matlab и Simulink

Программа линейного квантования реальных сигналов (в данном конкретном случае – это сигнал **vorona.wav**) имеет следующий вид:

```
% lin_quant.m - программа линейного квантования
[y,Fs,bits]=wavread('vorona.wav'); % чтение сигнала (Fs – ч-та дискретизации; bits -
кол-во бит)
ymax = max(abs(y)); y = y/ymax; % нормализация сигнала
T = length(y)/Fs; disp('протяженность сигнала T = '); disp(T); % протяж-сть сигнала в
секундах
t = 1/Fs:1/Fs:length(y)/Fs; figure; plot(t,y);
title(['Исходный сигнал, ' num2str(bits) ' бит, ' num2str(Fs) ' Гц'],'FontSize', 14, 'Font-
Name', 'MS Sans Serif');
xlabel('Время, с','FontSize', 14, 'FontName', 'MS Sans Serif');grid on;
wavplay(y,Fs); pause;
b = 8; % новое количество бит
L = 2^b; % новое количество уровней
q = 2/(L-1); % квант уровня
partition = -1+q/2:q:1-q/2;
codebook = -1:q:1;
[ind, ylinq, D] = quantiz(y, partition, codebook);
figure; plot(t,ylinq);
title(['Рез-тат лин. квантования, ' num2str(b) ' бит, ' num2str(Fs) ' Гц'],'FontSize', 14,
'FontName', 'MS Sans Serif');
xlabel('Время, с','FontSize', 14, 'FontName', 'MS Sans Serif');grid on;
wavplay(ylinq,Fs);
```

Результат линейного 4-битного квантования реального сигнала показан на рис. 5.16.



Рисунок 5.16 - Результат линейного 4-битного квантования реального сигнала

На графике рис. 5.16 отчетливо видно, что малые уровни речевого сигнала существенно искажены.

Программа нелинейного квантования реальных сигналов по А-закону имеет вид:

```

% nonlin_quant.m - программа линейного квантования
[y,Fs,bits]=wavread('vorona.wav'); % чтение сигнала (Fs – ч-та дискретизации; bits -
кол-во бит)
ymax = max(abs(y)); y = y/ymax; % нормализация сигнала
T = length(y)/Fs; disp('протяженность сигнала T = '); disp(T); % протяж-сть сигнала в
секундах
t = 1/Fs:1/Fs:length(y)/Fs; figure; plot(t,y);
title(['Исходный сигнал, ' num2str(bits) ' бит, ' num2str(Fs) ' Гц'],'FontSize', 14, 'Font-
Name', 'MS Sans Serif');
xlabel('Время, с','FontSize', 14, 'FontName', 'MS Sans Serif');grid on;
wavplay(y,Fs); pause;
%=== предыскажение сигнала по А-закону ===
A = 100; % значение параметра А
if 0<abs(y)<=1/A
    yn = A*abs(y).*sign(y)/(1+log(A));
else
    if y == 0
        y = 1e-3 ;
    end
    yn = (1+log(A*abs(y))).*sign(y)/(1+log(A));
end
sum_nan = sum(isnan(yn)); % кол-во NaN
yn = yn(~isnan(yn)); % удаление NaN
t1 = 1/Fs:1/Fs:(length(y)-sum_nan)/Fs;
figure; plot(t1,yn);
title(['Предыскаженный сигнал, ' num2str(bits) ' бит, ' num2str(Fs) ' Гц'],'FontSize', 14,
'FontName', 'MS Sans Serif');
xlabel('Время, с','FontSize', 14, 'FontName', 'MS Sans Serif');grid on;
wavplay(yn,Fs); pause;
%=== квантование предыскаженного сигнала =====
b = 8; % новое количество бит
L = 2^b; % новое количество уровней
q = 2/(L-1); % квант уровня
partition = -1+q/2:q:1-q/2;
codebook = -1:q:1;
[ind, yq, D] = quantiz(yn, partition, codebook);
figure; plot(t1,yq);
title(['Квантование предыскаж. сигнала, ' num2str(b) ' бит, ' num2str(Fs) '
Гц'],'FontSize', 14, 'FontName', 'MS Sans Serif');
xlabel('Время, с','FontSize', 14, 'FontName', 'MS Sans Serif');grid on;
% === обратное нелинейное искажение сигнала =====
if 0<abs(yq)<=1/(1+log(A))
    yv = (1+log(A))*abs(yq).*sign(yq)/A;
else
    yv = exp((1+log(A))*abs(yq)-1).*sign(yq)/A;
end
figure; plot(t1,yv);
title(['Рез-тат нелин. квантования, ' num2str(b) ' бит, ' num2str(Fs) ' Гц'],'FontSize', 14,
'FontName', 'MS Sans Serif');
xlabel('Время, с','FontSize', 14, 'FontName', 'MS Sans Serif');grid on;
wavplay(yv,Fs);

```

Результат нелинейного 4-битного квантования по А-закону (A=100) реального сигнала показан на рис. 5.17.

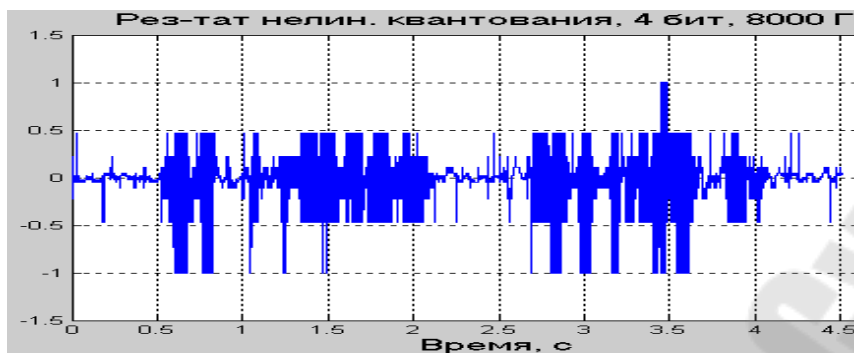


Рисунок 5.17 - Результат нелинейного ( $A=100$ ) 4-битного квантования реального сигнала

На графике рис. 5.17 отчетливо видны, что малые уровни речевого сигнала в целом искажены значительно меньше, однако почему-то наблюдаются локальные всплески. Предварительные исследования этого явления показали, что причиной тому – нулевые значения исходного сигнала.

Кстати, нелинейное квантование сигнала в среде Simulink (рис. 5.18) при тех же параметрах (4 бита,  $A=100$ ) приводит к существенно лучшим результатам в области малых уровней сигнала (рис.5.19).

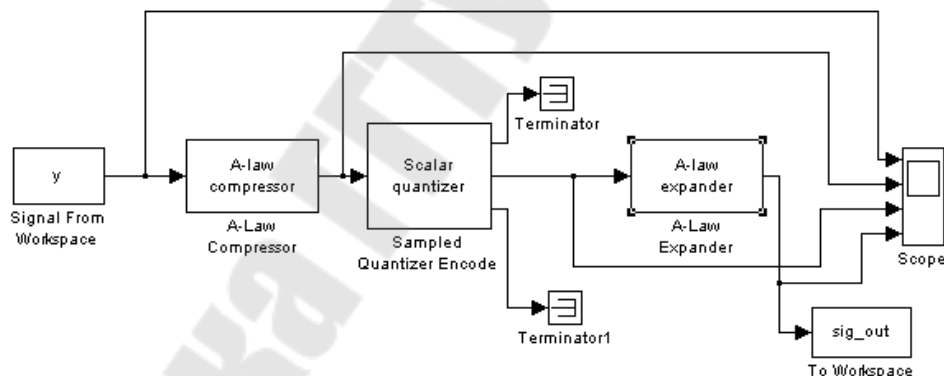


Рисунок 5.18 – Схема исследований

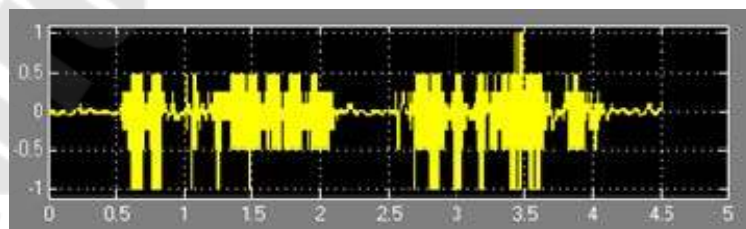


Рисунок 5.19 – Результат исследования

### Работа со звуковыми файлами в среде Matlab

1. Создать wav-файл формата PCM (использовать cooleedit, brainwave и пр.)

2. Поместить данные из wav-файла в рабочую область MATLAB, т.е. инициализировать переменную.

```
Y=wavread('pink22050_16',10*22050);
```

Здесь Y-имя переменной, которая будет хранить данные из файла;

pink22050\_16 – имя wav-файла;

22050 – частота дискретизации, Гц;

10 – количество секунд звука, помещаемых в переменную;

3. Если звук в файле стереофонический, а нужен только один канал, то один из каналов можно отбросить:

```
Y=Y(:,1);
```

4. Если необходимо уменьшить размер переменной, то можно ограничить данные напр. 5-ю секундами

```
Y=Y(1:5*22050);
```

5. Прослушать данные из рабочей области позволяет строка

```
wavplay(Y,22050);
```

6. Создать задержку на 32 отсчета

```
Y=[eye(32,1);Y];
```

### Построение цифрового фильтра, моделирующего АЧХ КТЧ

В таблице первые 2 столбца из графика, остальные рассчитаны

частота, кГц	затухание Нп (мощность) $a_{нп}$	затухание Дб (мощность) $a=a_{нп} * 8,688$	коэф. передачи (мощность) раз $K_M=10^{a_{нп}/10}$	коэф. пер (напр) раз $K=\sqrt{K_M}$
0				0
0,3	0,75	6,52	0,223	0,472
0,4	0,3	2,61	0,548	0,740
0,6	0	0	1	1
2,4	0,15	1,3	0,741	0,861
3	0,25	2,172	0,606	0,778
3,4	0,75	6,52	0,223	0,472
4				0

$f = [0 \ 0.3 \ 0.32 \ 0.4 \ 0.6 \ 2.4 \ 3 \ 3.38 \ 3.4 \ 4];$  %точки на оси частот, для которых будет задан коэф прд.

Теперь нормируем частоту

```
f = f ./ 4 % 4 -- половина частоты дискретизации
```

```
a = [ 0 0 0.47 0.740 1.0 0.86 0.780 0.4720 0 0] %Kпрд
```

$V = \text{fir2}(51,f,a);$  % в V получаем ИХ фильтра 51 порядка, окно Хэмминга

```
impz(V) % чертим полученную ИХ (рис.5.20)
```

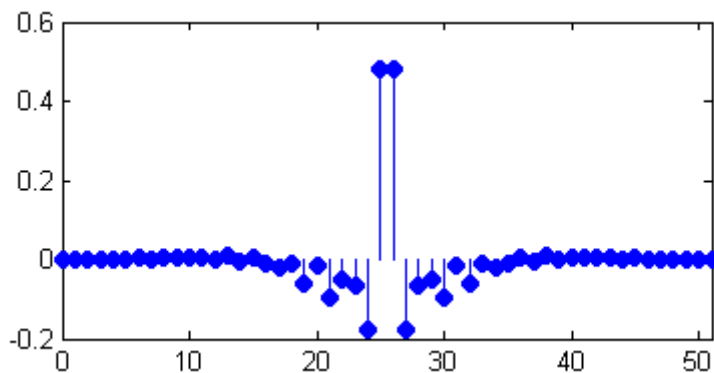


Рисунок 5.20 - Импульсная характеристика фильтра

`[h, fn] = freqz(B, 1, [], 8000) % получим в h – комплексный Кпрд, fn – вектор частот 8000 – частота дискретизации`  
`plot(fn, abs(h)) % чертим АЧХ (рис.5.21)`  
`plot(fn, angle(h)) % ФЧХ (рис.5.22)`

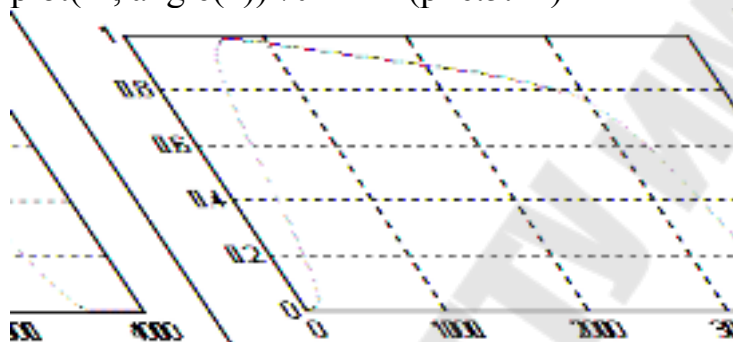


Рисунок 5.21 - Амплитудно-частотная характеристика фильтра

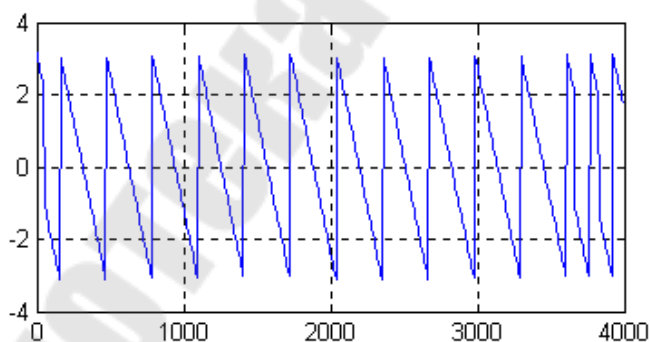


Рисунок 5.22 - Фазо-частотная характеристика фильтра

## 5.2. Ход работы

**1. Смоделировать в среде Matlab гармонический сигнал**  $s(t) = A\cos(2\pi f_0 t + \varphi)$ . Параметры сигнала приведены в таблице 5.2. Частота дискретизации  $F_s = 1000$  Гц. Длительность  $T$  гармонического сигнала выбрать так, чтобы на ней уложилось два периода.

**2. Смоделировать в среде Matlab процедуру линейного квантования гармонического сигнала.**

Параметры codebook и partition задайте, исходя из требуемого количества уровней  $L$  (табл.5.2);

**3. Смоделировать в среде Simulink процедуру нелинейного квантования гармонического сигнала.**

Нелинейное квантование осуществить с принятой в Европе характеристикой устройства сжатия (значение параметра  $A$  приведено в табл.5.2), приняв  $u_{\max} = x_{\max}$  (количество уровней квантования  $L$  оставить прежним).

**4. Сделать общие выводы по работе, сравнив между собой моделирование в среде Matlab и моделирование в среде Simulink**

Таблица 5.2

Варианты заданий

Вар. \ Парам.	1	2	3	4	5	6	7	8
$A$	1	1,1	1,2	1,3	1,4	1,5	1,6	1,7
$f_0$ , Гц	40	50	60	70	60	50	40	50
$\varphi$ , рад	0	$\pi$	0	0	$\pi$	0	$\pi$	0
$T$	5	6	7	6	5	6	7	6
$L$	5	10	15	20	25	30	35	40



### 5.3. Содержимое отчета

В отчете описать листинг программы Matlab для линейного квантования с числом уровней  $L$ , а также соответствующие графики исходного и квантованного сигналов; блок-схему моделирования в среде Simulink, а также графики результатов квантования в среде Simulink.

### 5.4. Контрольные вопросы

1. Что такое квантование?
2. В чем преимущество нелинейного квантования перед линейным?
3. Какие проблемы возникают при реализации нелинейного квантования, и как их решают?
4. Процесс моделирования процедуры линейного квантования
5. Процесс моделирования процедуры нелинейного квантования
6. На что влияет шаг квантования

## 6. ЛАБОРАТОРНАЯ РАБОТА №3

### Форматы изображений

**Цель работы:** освоение основных приемов работы с изображениями в системах телекоммуникаций

#### 6.1. Теоретические сведения

Рассмотрим, например, передачу данных по телефонному каналу с помощью стандартного модема со скоростью 33600 бит/с. Если необходимо передать изображение размером 240 на 352 точек, и для представления каждой точки затрачивается 3 байта, то, не трудно подсчитать, что на передачу одного изображения потребуется около минуты. В то же время, уже существуют и зарегистрированы в качестве международных стандартов алгоритмы сжатия изображений, позволяющие использовать телефонный канал для проведения видеоконференции.

В качестве второго примера рассмотрим запись подвижного изображения (видеосигнала) на компакт-диски. Одна секунда полноэкранный видеofilm с разрешением 480 на 720 точек при записи со скоростью 30 кадров в секунду потребует 20,736 Мбайт. Таким образом, только 31с фильма может быть записана на диск емкости 650 Мбайт. Использование современных алгоритмов сжатия подвижных изображений позволяет записать на такой компакт-диск 74-минутный фильм.

Системы сжатия изображений можно разделить на два больших класса: сжатие изображений без потери качества и сжатие с потерей качества.

При сжатии без потерь восстановленные на приемной стороне данные в точности совпадают с входным сигналом кодера. Для сжатия изображений без потерь можно использовать известные из теории информации алгоритмы кодирования источников

Эффективность кодирования обычно оценивается в битах на отсчет или в битах в секунду. Распространенным критерием качества при кодировании с потерями является отношение сигнал/шум (SNR) на выходе декодера

$$SNR = 10 \log_{10} (E_{inp} / E_n) \text{ (дБ)},$$

где  $E_{inp}$  - это энергия входного сигнала кодера,  $E_n$ -это энергия шумового сигнала

Под энергией шумового сигнала здесь понимается энергия разности между входным сигналом кодера и выходным сигналом декодера. Отношение сигнал/шум измеряется в децибелах. В случае обработки изображений вместо SNR часто используется пиковое отношение сигнал/шум, которое определяется по формуле

$$PSNR = 10 \log_{10} ((255)^2 / E_n),$$

где 255-это максимальное десятичное значение беззнакового 8-битового числа, соответствующего одной точке черно-белого изображения.

Рассмотрим стандартные форматы, используемые для представления цветных изображений. В цифровой форме любое изображение представляется в виде двумерного поля отсчетов (точек), называемых пикселями. Большинство цветных сканеров генерируют изображения с красной, зеленой и голубой цветовыми компонентами или так называемые изображения в формате RGB. В простейшем варианте каждому пикселю сопоставляется 3 числа, характеризующие интенсивность красной, зеленой и голубой составляющих изображения в данной точке. Таким образом, изображение описывается тремя цветовыми компонентами, каждая из которых представляет собой прямоугольный массив чисел. Значение интенсивности компоненты в одной точке характеризуется целым числом из интервала 0...255 и для ее хранения отводится 1 байт. Всего изображение размера  $M \times N$  в формате RGB занимает  $3MN$  байт в памяти компьютера.

В типичных изображениях в формате RGB имеется существенная корреляция между цветными компонентами и с точки зрения сжатия изображений формат RGB является заведомо избыточным. Как известно, в стандартах телевизионного вещания используется другое представление изображений, при котором также используются 3 компоненты сигнала, но эти три компоненты почти некоррелированы друг с другом. Компоненты R,G и B преобразуются в яркостную компоненту Y и две цветоразностных компоненты U и V, формата YUV.

Преобразование формата RGB в формат YUV выполняется по

формулам

$$Y = 0.299 R + 0.587G + 0.114 B ,$$

$$U = ( B - Y )0.5643 + 128 ,$$

$$V = ( R - Y )0.7132 + 128 .$$

Обратное преобразование выполняется по формулам

$$G = Y - 0.714(V - 128) - 0.334(U - 128) ,$$

$$R = Y + 1.402(V - 128) ,$$

$$B = Y + 1.772(U - 128) .$$

В формате YUV компоненты слабо коррелированы. Более того, так как большая часть информации сосредоточена в яркостной компоненте, то мы теряем мало информации, если выполним децимацию (прореживание) компонент U и V с коэффициентом 2. При таком прореживании 4 соседние точки изображения, образующие квадрат размера 2\*2, описываются 4 значениями яркостных компонент Y, одним значением компоненты U и одним значением компоненты V. Каждая из цветоразностных компонент вычисляется как округленное до ближайшего целого среднее арифметическое соответствующих четырех значений рассматриваемого квадрата. Результатом является стандартный формат YUV 4:1:1, который, как правило, является входным для большинства видеокодеров. Нетрудно подсчитать, что на квадрат 2\*2 будет затрачено уже не 12, а 6 байт. Таким образом, получается сжатие в 2 раза без сколько-нибудь заметного искажения изображения.

Описанное выше представление цветных изображений в виде RGB является фактически вариантом более общей конструкции формата BMP (от Bitmap). Этот формат, будучи одним из самых распространенных форматов хранения растровой графической информации, является стандартным для операционных систем Windows 3.x, Windows 95 и Windows xxx. В формате BMP изображение может храниться как без сжатия, так и со сжатием без потерь с использованием метода кодирования длинами серий. Изображения могут быть монохромными (1 бит/пиксел) или цветными (4,8,16,24 или 32 бита/пиксел).

Файл в формате BMP состоит из четырех частей:

BITMAPFILE
BITMAPINFO
RGBQUADS
Pixels

В BITMAPFILEHEADER и BITMAPINFOHEADER содержатся параметры файла и изображения, в RGBQUADS записывается цветовая палитра, а затем хранятся собственно пиксели изображения (как индексы палитры или как величины красной, зеленой и голубой составляющей цвета).

Формат BITMAPFILEHEADER:

Название поля	Число	Комментарий
bfType	2	Тип файла. Должен быть "BM".
bfSize	4	Размер файла в байтах
bfReserved1	2	Зарезервировано. Должно быть 0.
bfReserved2	2	Зарезервировано. Должно быть 0.
BfOffBits	4	Расстояние в байтах от BITMAPFILEHEADER до пикселей изображения

Формат BITMAPINFOHEADER:

Название поля	Число	Комментарий
biSize	4	Размер структуры BITMAPINFOHEADER в байтах
biWidth	4	Ширина изображения в пикселах
biHeight	4	Высота изображения в пикселах
biPlanes	2	Число плоскостей на устройстве вывода. Должно быть 1
biBitCount	2	Число бит на пиксел (1,4,8,16,24,32)
biCompression	4	Метод хранения пикселей (BI_RGB, BI_RLE8, BI_RLE4, BI_BITFIELDS)
biSizeImage	4	Размер изображения в байтах (Может быть 0, если biCompression=BI_RGB)

biXPelsPerMeter	4	Горизонтальное разрешение устройства вывода (в пикселах/метр)
biYPelsPerMeter	4	Вертикальное разрешение устройства вывода (в пикселах/метр)
biClrUsed	4	Число цветовых индексов в таблице цветов, которые используются в изображении
biClrImportant	4	Число цветовых индексов, которые считаются важными при выводе изображения

Если величина biHeight положительна, то изображение записано снизу-вверх и начало изображения - левый нижний угол. Если величина biHeight отрицательна, то изображение записано сверху-вниз и начало изображения в левом нижнем углу.

Поле biCompression может принимать следующие значения:

BI\_RGB (0)- формат без сжатия.

BI\_RLE8 (1)- сжатие длинами серий, 8 бит/пиксел. Каждая запись состоит из 2х байтов, в первом байте хранится число цветовых индексов в серии, во втором байте цветовой индекс.

BI\_RLE4 (2) - сжатие длинами серий, 4 бит/пиксел.

BI\_BITFIELDS (3) изображение хранится без сжатия, цветовая таблиц состоит из трех четырехбайтовых масок для выделения красной, зеленой и голубой составляющей каждого пиксела. Этот режим используется при 16 и 32 битах/пиксел

RGBQUADS состоит из четверок байт

rgbBlue,rgbGreen,rgbRed,rgbReserved, которые определяют голубую, зеленую и красную составляющую цвета. Размер массива RGBQUADS зависит от числа бит на пиксел и метода сжатия.

biBitPerPi	Значение
1	Изображение монохромное. RGBQUADS содержит две четверки, определяющие цветовые компоненты “черных” и “белых” пикселей. В этом случае каждый бит массива задает один пиксел
4	Изображение содержит до 16 цветов, в RGBQUADS записано до 256 четверок, определяющих палитру изображения.
8	В изображении до 256 цветов. RGBQUADS содержит до 256 четверок, определяющих палитру
16	До $2^{16}$ цветов. Если biCompression=BI_RGB, то массив RGBQUADS пуст, на каждый пиксел изображения отводится 2 байта, в которых записаны B,G,R цветовые компоненты (5 бит/компоненту, старший бит двухбайтового слова не используется.) Если biCompression=BI_BITFIELDS, то RGBQUADS состоит из трех четырехбайтовых масок, определяющих R,G,B компоненты.
24	Если biBitPerPixel=24, то массив RGBQUADS пуст и пиксели изображения хранятся в виде троек байт Blue,Green,Red.
32	Аналогично biBitPerPixel=16, только на пиксел отводится 4 байта, три байта на R,G и B, старший байт не используется.

В настоящее время в файлах BMP изображения обычно хранятся без сжатия в формате либо 8 бит/пиксел(с палитрой) либо 24 бит/пиксел.

Пиксели изображения хранятся в файле строка за строкой. Представление каждой строки должно быть выравнено на четырехбайтовую границу. Недостающие байты заполняются нулями.

### Работа с изображением в среде Matlab

**A = imread(filename)** – чтение графического файла

A – трехмерная матрица((x,y,n), где x, y – координаты пикселя, а n – номер цветовой компоненты. Например – A(10,20,2) – показывает для rgb значение зеленой компоненты пикселя с координатами (10,20) в диапазоне от 0 до 255)

Filename – имя файла, передается как строка. Матлаб ищет данный файл в текущей директории, поэтому лучше записывать полный адрес.

**imwrite(A,filename,fmt)** – запись графического файла

A – трехмерная матрица((x,y,n)

Filename – имя файла, передается как строка. Матлаб ищет данный файл в текущей директории, поэтому лучше записывать полный адрес.

Fmt – формат файла, передается как строка. Наиболее распространенные форматы: ‘bmp’, ‘jpeg’, ‘jpg’.

**imtool(A)** или **imshow(A)** – отображает изображение, загруженное в A с помощью функции imread

**info = imfinfo(filename, fmt)** – запись в структуру info информации о файле. Например, размер файла, последнее редактирование и т.п.

*Пример:*

```
A = imread('pic', 'jpeg'); % load image
imshow(A); % show image
h = fspecial('gaussian'); % make filter
B = imfilter(A, h); % use filter h on image A (blur image)
imwrite(B, 'pic1', 'jpeg'); % save image
imshow(B); % show new image B
```

Вся дальнейшая обработка файлов осуществляется через массив A как A(i,j,z), где i и j – координаты пикселя, z задает цветовую компоненту ( для цветных изображений). Для более подробной информации о функциях обработки графических файлов и всего матлаба в целом, используйте Help матлаба.

## 6.2. Ход работы

### Задание 1

1. Дан файл с изображением размером 160X120 точек в формате RGB. Преобразовать его в формат YUV.
2. Выполнить децимацию компонент U и V.
3. Выполнить обратное преобразование YUV в RGB.
4. Вывести изображение на экран.
5. Оценить пиковое отношение сигнал/шум для всех компонент изображения.



6. Выполнить статистический анализ изображений в форматах YUV и RGB. Оценить энтропию значений компонент и достижимое сжатие при одномерном побуквенном кодировании компонент изображения без потерь.

## **Задание 2**

1. Дан файл в формате BMP, 8 бит/пиксел. Прочитать файл, вывести размеры, число цветов и другие поля в заголовке изображения, выделить и напечатать массив цветов- палитру.

2. Дан файл в формате BMP, 24бит/пиксел. Прочитать исходное изображение, растянуть его в 2 раза по высоте и ширине и записать в новый BMP-файл.

3. Дан файл в формате BMP, 24бит/пиксел. Прочитать изображение, в каждом пикселе поменять между собой красную и голубую компоненту и записать результат в новый BMP-файл.

4. Создать 24-битовый BMP-файл, записать в него изображение, в котором зеленая компонента каждого пиксела не изменяется, красная компонента растет снизу вверх, а зеленая - слева направо.

## **6.5. Содержимое отчета**

В отчете описать исходные данные, программную среду обработки, алгоритмы, листинг программ в соответствии с заданием, результаты выполнения заданий.

## **6.6. Контрольные вопросы**

1. Основные форматы изображений.
2. Характеристики основных форматов изображений.
3. Способы компрессии изображений.
4. Параметры влияющие на коэффициент компрессии.
5. Параметры файла класса изображение.
6. Что такое отношение сигнал/шум в изображении.
7. Команды работы с изображением в среде Matlab.
8. Что такое разрешение изображения.
9. Как вычислить объем изображения в битах.

## Список литературы

1. Покровский Н.Б. Расчет и измерение разборчивости речи. – М.,Связьиздат,1962,390с.
2. Хорев А.А. Технические каналы утечки акустической (речевой) информации (<http://st.ess.ru/publications/articles/tspi/tspi.htm>)
3. Хорев А.А. Классификация и характеристика технических каналов утечки информации, обрабатываемой ТСПИ и передаваемой по каналам связи (<http://st.ess.ru/publications/articles/tspi/tspi.htm>)
4. ГОСТ Р 50840-95. Государственный стандарт Российской Федерации. Передача речи по трактам связи. Методы оценки качества, разборчивости и узнаваемости. Издание официальное. – М.: Госстандарт России, 1997.
5. Вахитов Я.Ш. Слух и речь. – Л., изд.ЛИКИ, 1973.
6. Сапожков М.А., Михайлов В.Г. Вокодерная связь. –М.,Радио и связь,1983,247с.
7. Хорев А.А., Макаров Ю.К. К оценке эффективности защиты акустической (речевой) информации (<http://st.ess.ru/publications/articles/tspi/tspi.htm>)
8. Ковалгин Ю.А., Володин Э.И. Цифровое кодирование звуковых сигналов. С-Пб, КОРОНА принт, 2004, 231с.
9. Котович А.Е., Рябенский В.М. Спектрально-адаптированные нормы защиты речевой информации от утечки по акустическому каналу. – Безопасность информации, №1(9), 1998,с.32-35.
10. Железняк В.К., Макаров Ю.К.,Хорев А.А. Некоторые методические подходы к оценке эффективности защиты речевой информации//Специальная техника. – М.: 2000.– № 4.
11. Хорев А.А., Макаров Ю.К. К оценке эффективности защиты акустической (речевой) информации (<http://st.ess.ru/publications/articles/tspi/tspi.htm>)
12. Хекл М., Мюллер Х.А. Справочник по технической акустике. – Л.,Судостроение, 1980.
13. Семенов В.Ю. Разработка адаптивных методов коррекции речевых сигналов на основе авторегрессионной модели голосового тракта. – Диссертация канд.техн.наук, К., 2004, 172с.
14. Назаров М.В., Прохоров Ю.Н. Методы цифровой обработки и передачи речевых сигналов.–М.:Радио и связь,1985. –176 с.

15. 10. Сэломон. Сжатие данных, изображений и звука. – М.: Техносфера, 2004. – 368 с.
16. Чемпен Н., Чемпен Д. Цифровые технологии мультимедиа. – М.: Вильямс, 2006. – 624 с.
17. Чемпен Н., Чемпен Д. Цифровые графические инструменты. – М.: Вильямс, 2006. – 656 с.
18. Прэтт У. Цифровая обработка изображений.– М.: Мир, 1982.– 790 с.
19. Вудс Р., Гонсалес Р. Цифровая обработка изображений. – М.: Техносфера, 2005.– 1072 с.
20. Грузман Н.С. , Киричук В.С., Косых В.П. и др. Цифровая обработка изображений в информационных системах: Уч. пособие. – Новосибирск, изд НГТУ, 2000.– 440 с.
21. С. Уэлстид. Фракталы и вейвлеты для сжатия изображений в действии. – М.: Изд. Триумф, 2003 – 320 с.
22. Ватолин Д., Ратушняк А., Смирнов М, Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: Диалог-МИФИ, 2003. – 384 с.
23. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. – М., С-Пб, К., изд. дом «Вильямс», 2003. – 1092 с.
24. Гультияев А. Имитационное моделирование в среде Windows. – С-Пб, КОРОНА принт, 1999. – 287 с.
25. Калюжний О.Я. Моделювання систем передачі сигналів в обчислювальному середовищі MATLAB-Simulink. – К., “Політехніка”, 2004. – 135 с.

**Ковалев Алексей Викторович**  
**Литвинов Дмитрий Александрович**

## **ОБРАБОТКА АУДИО- И ВИДЕОСИГНАЛОВ В СИСТЕМАХ ТЕЛЕКОММУНИКАЦИЙ**

**Лабораторный практикум  
по курсу «Системы телекоммуникаций»  
для студентов специальности 1-36 04 02  
«Промышленная электроника»  
дневной формы обучения  
В четырех частях  
Часть 1**

Подписано к размещению в электронную библиотеку  
ГГТУ им. П. О. Сухого в качестве электронного  
учебно-методического документа 04.10.11.

Рег. № 32Е.  
E-mail: [ic@gstu.by](mailto:ic@gstu.by)  
<http://www.gstu.by>