



МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Информационных технологий и систем безопасности

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(Бакалавр)

На тему Разработка программной модели для декодирования текстового сообщения с использованием мягкого и жесткого алгоритма Витерби

Исполнитель Суетин Никита Игоревич
(фамилия, имя, отчество)

Руководитель кандидат технических наук, доцент кафедры ИТиСБ
(ученая степень, ученое звание)
Чернецова Елена Анатольевна
(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой

(подпись)

доктор технических наук
(ученая степень, ученое звание)
Бурлов Вячеслав Георгиевич
(фамилия, имя, отчество)

« » _____ 2023г.

Санкт-Петербург

2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	1
1. ОБЩИЕ СВЕДЕНИЯ О КОДИРОВАНИИ И ДЕКОДИРОВАНИИ ИНФОРМАЦИИ.....	4
1.1. Особенности кодирования источника при передаче сообщений.....	4
1.2. Особенности помехоустойчивого кодирования в телекоммуникационных системах	31
2. ОСОБЕННОСТИ ДЕКОДИРОВАНИЯ ИНФОРМАЦИИ ПО АЛГОРИТМУ ВИТЕРБИ.....	45
2.1 . Пример сверточного декодирования Витерби.....	45
2.2. Реализация декодера.....	52
2.2.1. Процедура сложения, сравнения и выбора	52
2.2.2. Вид процедуры сложения, сравнения и выбора на решетке	53
2.2.3. Память путей и синхронизация	55
2.3. Модели каналов: мягкое или жесткое принятие решений	56
3. РАЗРАБОТКА ПРОГРАММНОЙ МОДЕЛИ ДЛЯ ДЕКОДИРОВАНИЯ ТЕКСТОВОГО СООБЩЕНИЯ С ИСПОЛЬЗОВАНИЕМ МЯГКОГО И ЖЕСТКОГО АЛГОРИТМА ВИТЕРБИ	63
3.1. Программная модель для декодирования текстового сообщения.....	63
3.2. Разработка алгоритма программы для разработки модели декодирования	65
Заключение	70
Список литературы	71
Приложение А	74

ВВЕДЕНИЕ

Целью данной работы является разработка программной модели для декодирования текстового сообщения с использованием мягкого и жесткого алгоритма Витерби.

Актуальность работы обусловлена тем, что в настоящее время развитие систем передачи информации по радиоканалам имеет тенденцию, заключающуюся в непрерывном росте количества радиоизлучающих средств за счет развития современных систем передачи информации. Увеличивается разнообразие источников радиоизлучения, в том числе от промышленного оборудования и транспорта. Одновременно при этом должны сохраняться требования на качественные характеристики, что определяет важную научную и практическую проблему, заключающуюся в разработке новых методов и алгоритмов сверточной обработки сигналов для повышения помехоустойчивости систем передачи информации в условиях ухудшающейся помеховой обстановки. Эта проблема может быть разрешена за счет повышения характеристик путем модификации имеющихся алгоритмов сверточной обработки сигналов [1].

Оптимальными декодерами сверточных кодов, которые основаны на их решетчатом представлении являются алгоритмы Витерби и BCJR (Bahl-Cocke-Jelinek-Raviv - первые буквы фамилий ученых, предложивших алгоритм). Декодер Витерби может работать как с жесткими, так и с мягкими входами, в то время как BCJR является декодером с мягким входом [2]. Декодер Витерби является алгоритмом, который минимизирует вероятность ошибки на слово и выбирает декодированное слово u так, чтобы максимизировать условную вероятность поступления на приемник переданного сигнала $P(r|u)$, т.е. является декодером по максимуму правдоподобия (МП). BCJR решает другую задачу и минимизирует вероятность ошибки на бит. В данной работе рассматривается алгоритм декодирования Витерби.

1. ОБЩИЕ СВЕДЕНИЯ О КОДИРОВАНИИ И ДЕКОДИРОВАНИИ ИНФОРМАЦИИ

1.1. Особенности кодирования источника при передаче сообщений

Изначально большинство передаваемой информации (кроме информационного обмена между двумя компьютерами) имеет текстовую или аналоговую форму. Если информация является буквенно-цифровым текстом, то используется один или несколько стандартных форматов – методов знакового кодирования: ASCII (American Standard Code for Information Interchange – Американский стандартный код для обмена информацией), EBCDIC (Extended Binary Coded Decimal Interchange Code – расширенный двоичный код для обмена информацией), код Бодо, код Холлерита и т.д. Таким образом текстовый материал преобразуется в цифровой формат.

Знаки вначале кодируются в последовательность битов, которая называется потоком битов или узкополосным сигналом. После этого формируются группы из k бит, называемые символами, причем число всех символов конечно, а их совокупность называется алфавитом. Размер алфавита равен $M = 2^k$. Выбор значений M , k является важным первоначальным этапом проектирования любой цифровой системы. При $k=1$ $M=2$, а система называется бинарной. Модулятор при этом использует один из двух различных сигналов для представления значения «один» и «нуль». В этом частном случае символ и бит – это одно и то же.

На рис. 1.1.1. приведен пример разделения потока битов для различных значений M и k . Текстовое сообщение на рисунке – это слово «THINK». Использование 6-битовой кодировки ASCII дает поток, состоящий из 30 бит. Если размер алфавита выбрать $M=8$, то биты должны группироваться по три ($k = \log_2 8$), а передатчик должен иметь набор из восьми сигналов $s_i(t)$, которые ставятся в соответствие символам. Если выбрать $M=32$, то $k = \log_2 32 = 5$. Биты сообщения группируются по пять для создания символов. Разбиение потока битов на символы не обязательно должно совпадать с границами кодированных текстовых знаков, т.к. передается строка сообщения, и только конечный

пользователь приписывает текстовое значение полученной последовательности битов.

Если информация является аналоговой, ее знаковое кодирование (как в случае текстовой информации) невозможно. Вначале информацию следует перевести в цифровой формат. Для этого его нужно дискретизировать по времени и квантовать по амплитуде.

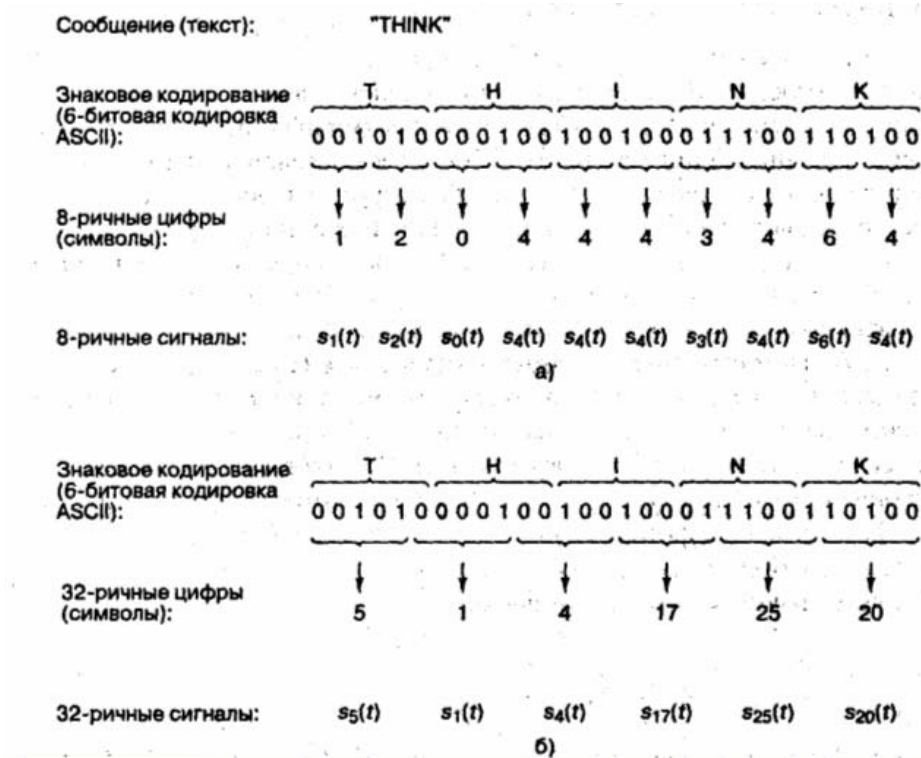


Рис. 1.1.1.— Сообщения, знаки и символы: а) 8-ричный пример; б) 32-ричный пример [3]

В случае акустических данных (голоса) информацию можно представить непосредственно с помощью электромагнитного сигнала, имеющего такой же спектр. Впрочем, существует необходимость нахождения компромисса между точностью воспроизведения звука, передаваемого электрическими средствами и стоимостью самой передачи, повышающейся при увеличении ширины полосы. Спектр речевого сигнала находится в диапазоне приблизительно от 100 Гц до 7 кГц, хотя и более узкая полоса также даст приемлемое воспроизведение речи. Стандартным для звукового канала является спектр от 300 до 3400 Гц. Этого достаточно для передачи речи, также такой выбор сводит к минимуму требуемую пропускную способность передачи и позволяет использовать

достаточно недорогие телефонные аппараты. Телефонный микрофон преобразует входящий акустический сигнал в электромагнитный сигнал, находящийся в диапазоне от 300 до 3400 Гц. Затем этот сигнал передается через телефонную сеть приемнику, который и воспроизводит его как акустический сигнал.

Важнейшее преимущество цифровой связи состоит в том, что цифровая техника позволяет обеспечить любое наперед заданное качество связи. Для цифровой передачи речи необходимо произвести аналого-цифровое преобразование речевого сигнала: подвергнуть аналоговый сигнал дискретизации, квантованию и кодированию. Совокупность этих операций называется импульсно-кодовой модуляцией (ИКМ). Импульсно-кодовая модуляция (pulse-codemodulation — PCM) — это название, данное классу узкополосных сигналов, полученных из сигналов импульсно-амплитудной модуляции (pulse-amplitudemodulation — PAM) путем кодирования каждой квантованной выборки цифровым словом. Исходная информация дискретизируется и квантуется в один из L уровней; после этого каждая квантованная выборка проходит цифровое кодирование для превращения в l -битовое ($l = \log_2 L$) кодовое слово. Для узкополосной передачи биты кодового слова преобразовываются в импульсные сигналы.

Для точного описания формы речевого сигнала, согласно теореме Котельникова, его дискретизацию приходится проводить с частотой 8 кГц (т. е. брать отсчеты через каждые 125 мкс), а для получения нормального качества воспроизведения речи квантовать каждый отсчет по шкале, разбитой на 8192 уровня (при выборе равномерной шкалы квантования). Чтобы закодировать каждое значение отсчета с помощью двоичного числа, потребуется 13 разрядов. В результате для передачи телефонного разговора с помощью последовательности двоичных импульсов необходима скорость $8 \times 13 = 104$ кбит/с (что соответствует при оптимальном кодировании полосе частот 52 кГц). Сравнивая это число с полосой частот в 3100 Гц, которая требуется для аналоговой передачи, нельзя не поразиться колоссальному росту необходимой

полосы, которым приходится расплачиваться за преимущества цифровой передачи. Естественно попытаться при реализации цифровой системы передачи понизить скорость передачи.

Первый шаг в этом направлении довольно очевиден. Квантование на 2^{13} уровней оказывается необходимым потому, что уровни аналоговых речевых сигналов могут изменяться в диапазоне 60 дБ. При этом сигналы высокого уровня при равномерной шкале квантования квантуются с таким же шагом, что и сигналы низкого уровня. Однако человеческая речь характеризуется уникальными статистическими свойствами, одно из которых проиллюстрировано на рис. 1.1.2. На оси абсцисс отложены амплитуды сигнала, нормированные на среднеквадратическое значение величины таких амплитуд в типичном канале связи, а на оси ординат – вероятность. Для большинства каналов речевой связи доминируют очень низкие тона; 50% времени напряжение, характеризующее энергию обнаруженной речи, составляет менее четверти среднеквадратического значения. Значения с большими амплитудами встречаются довольно редко; только 15% времени напряжение превышает среднеквадратическое значение. Поэтому можно сделать вывод о том, что так как восприятие сигналов органами слуха человека пропорционально логарифму уровня сигнала, то естественно было бы сигналы высокого уровня квантовать более грубо, а низкого уровня - более точно. Применяя нелинейное квантование с использованием логарифмического закона, можно обойтись восемью разрядами на отсчет, сохранив почти такое же качество передачи. В результате скорость передачи двоичных разрядов окажется равной 64 кбит/с. Именно эта скорость получила самое широкое распространение, она зафиксирована в рекомендации МККТТ С.711, и на ней работает аппаратура ИКМ во многих странах.



Рис.1.1.2. – Распределение амплитуд речи одного лица [4]

Аналоговый сигнал имеет большую избыточность. Это позволяет предсказывать очередной отсчет и передавать только разницу между фактическим и предсказанным значением каждого отсчета. Если применить хорошую схему предсказания, изменение амплитуды приращения сигнала окажется меньше изменения амплитуды самого сигнала, что приведет к уменьшению количества передаваемой информации. На этом принципе строится дифференциальная ИКМ (ДИКМ) и адаптивная дифференциальная ИКМ (АДИКМ), которая позволяет понизить скорость передачи речи до 32 кбит/с и ниже за счет дальнейшего усложнения приемопередающей аппаратуры. Продолжая усложнять аппаратуру, можно довести скорость передачи речи до 100-300 бит/с. Можно представить себе, например, на передающей стороне преобразователь речи в текст, а на приемной стороне - читающую машину. Известны пути дальнейшего снижения скорости передачи речи, но не будем на этом останавливаться. Дело в том, что аппаратура цифровой передачи речи со скоростью 64 кбит/с всех удовлетворила потому, что она оказалась работоспособной при использовании самых простых симметричных кабелей с парной скруткой. Аппаратура ИКМ-30

начала свое триумфальное шествие с уплотнения соединительных линий между городскими телефонными станциями. Там, где раньше по кабельной паре можно было организовать соединительную линию для передачи лишь одного разговора, аппаратура ИКМ-30 позволила организовать по этой же паре передачу 30 разговоров. О лучшем использовании такой пары с помощью аналоговой аппаратуры многоканальной связи не могло быть и речи. Позднее появились аппаратура ИКМ-120 и другие высокопроизводительные системы, работающие по коаксиальным кабелям и волоконным световодам, и острота вопроса об уменьшении скорости передачи разговорных сигналов ниже 64 кбит/с в сетях проводной связи практически была снята. Даже многочисленные разработки аппаратуры цифровой передачи со скоростью 32 кбит/с, реализованные во многих странах на основе принципа АДИКМ, не получили достаточно широкого применения. Баланс между увеличением пропускной способности каналобразующей аппаратуры и сложностью оконечного оборудования в проводной связи пока так и не склонился в пользу первого решения.

Совсем другие перспективы открылись в конце 1980-х - начале 1990-х годов, когда начали развиваться сотовые системы цифровой радиотелефонной связи. В отличие от проводных сетей, где расширение пропускной способности возможно за счет прокладки новых линий, т. е. возобновления ресурсов пропускной способности, в радиосетях действует жесткий закон тесноты в эфире, и приходится иметь дело с невозобновляемым ресурсом радиочастот. Правда, идея сотовой связи как раз и состоит в возобновлении ресурса радиочастот путем повторения частоты передачи на территории, до которой не доходит сигнал той же частоты от мешающей радиостанции. Но возможности такого возобновления ресурса и здесь ограничены, поэтому дальнейшее усложнение аппаратуры ради снижения скорости передачи оказывается оправданным.

Например, в принятой в большинстве стран Европы системе сотовой цифровой связи GSM стандартные скорости передачи речи составляют 13 и 6,5

кбит/с. Для осуществления подобной системы передачи пришлось обратиться к более глубокому проникновению в механизм речеобразования.

Как известно, один из важнейших результатов современной теории передачи информации состоит в рекомендации разделения задач кодирования источника и кодирования канала. В задачу кодирования источника информации входит описание передаваемого сообщения в максимально экономной форме, т. е. удаление избыточности в сообщении. Полученное таким образом сжатое сообщение становится более уязвимым к воздействию помех и может оказаться искаженным при передаче. Поэтому после кодирования источника применяется кодирование канала, в задачу которого входит защита передаваемого сообщения от помех. Кодирование канала требует внести в передаваемое сообщение некоторую избыточность, но не случайную, которая присутствовала в первоначальном сообщении, а строго обоснованную теоретически и которая гарантирует оговоренное качество передачи.

Итак, имеется цифровая версия аналогового речевого сигнала, т. е. функция, описывающая, например, закон изменения тока во времени. Из такого сигнала нужно попытаться удалить избыточность. Эту задачу можно решать несколькими методами. Один из них - попытаться найти избыточность путем чисто математического анализа рассматриваемой функции. Другой путь решения задачи - анализ акустических характеристик этой функции (с точки зрения ее восприятия органами слуха). Наконец, можно искать избыточность моделированием самого процесса речеобразования. Именно последний из перечисленных методов нашел применение в современных системах цифровой радиосвязи.

Под формантами понимаются частотные резонансы (полуса передаточной функции) речевой акустической системы. Параметры формант (частота, ширина, уровень) опеределаются акустическими свойствами системы. Наиболее важный параметр - частота форманты, тесно связан с геометрической конфигурацией речевого тракта. Поскольку в процессе речи конфигурация

речевого тракта меняется, то соответственно меняются формантные частоты (рис. 1.1.3).

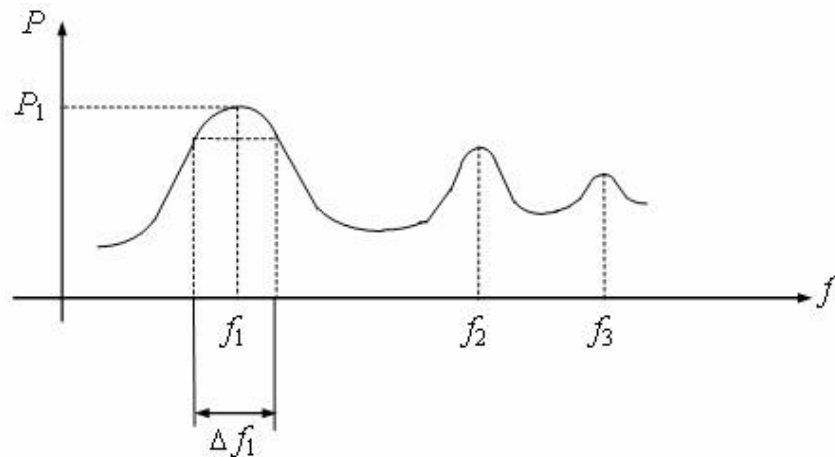


Рис.1.1.3. – Пример частотного спектра речи [5]

Для удовлетворительного синтеза речи обычно нужны две - четыре формантные частоты. Они лежат в диапазоне от 200 (первая форманта мужского голоса) до 2000 Гц (третья форманта женского голоса). Точным расположением формантных частот в звуковом спектре и определяется звук, который мы интерпретируем как речь. Причем, все формантные частоты присутствуют в речи одновременно и непрерывно перемещаются вверх-вниз по частотному спектру в соответствии с особенностями произносимого слова. Поэтому, слушая говорящего человека, вы слышите звук не какой-либо одной частоты, а множество обертонов, которые образуются при фильтрации импульсов, формируемых на выходе голосового тракта.

Итак, в основе формантного синтеза лежит аналогия с моделью речеобразования человека. Рассмотрим формирование гласных звуков на модели (рис.1.1.4).

Источник возбуждения создает импульсы основного тона, частота следования которых непрерывно меняется в процессе формирования речи. Речевой тракт при образовании гласных звуков работает как набор резонаторов, в которых происходит фильтрация сигнала возбуждения. В результате образуется спектральная картина, содержащая ряд максимумов. Максимумы соответствуют резонансам тракта (это и есть форманты).



Рис.1.1.4. – Модель речеобразования

Таким образом, форманты - это некоторая частотная область концентрации энергии в спектре звука. Используют от двух до шести формант в зависимости от требуемой точности анализа речи. Суммарный выходной сигнал формантных фильтров (резонаторов) достаточно близко соответствует частотному спектру речи человека, и наш слух воспринимает его как речевое сообщение.

В таблице 1.1.1 представлены формантные частоты для некоторых фонем гласных звуков.

Таблица 1.1.1¹

Фонема	Формантные частоты		
	F_1	F_2	F_3
о	275	850	2400
и	250	2300	3000
а	575	900	2450

Путем одновременной генерации формантных частот F_1 , F_2 , F_3 согласно таблице 1.1.1 можно получить гласные звуки.

Структурная схема формантного синтезатора гласных звуков приведена на рис.1.1.5.

Структурная схема форматного синтезатора гласных звуков включает задающий генератор частоты основного тона, полосовые фильтры, перестраиваемые на формантные частоты, соответствующие синтезируемой фонеме с помощью переменных резисторов R_1 .. R_3 и сумматор, суммирующий сигналы с трех фильтров. В спектрограмме выходного сигнала этой схемы содержатся три формантные частоты, идентичные формантным частотам в спектрограмме речи человека, произносящего те же гласные.

¹Таблица составлена автором по данным, полученным/используемым в ходе исследования

Гораздо сложнее формировать согласные звуки. Согласные - звуки речи, при произношении которых в полости рта образуются преграды для выдыхаемого воздуха:

- взрывные - при полном смыкании органов речи (п, т, к);
- фрикативные - образуется щель (с, ф, х);
- носовые согласные (н, м);
- аффриката - согласный звук, представляющий слитное сочетание (ч -тщ, ц -тс).

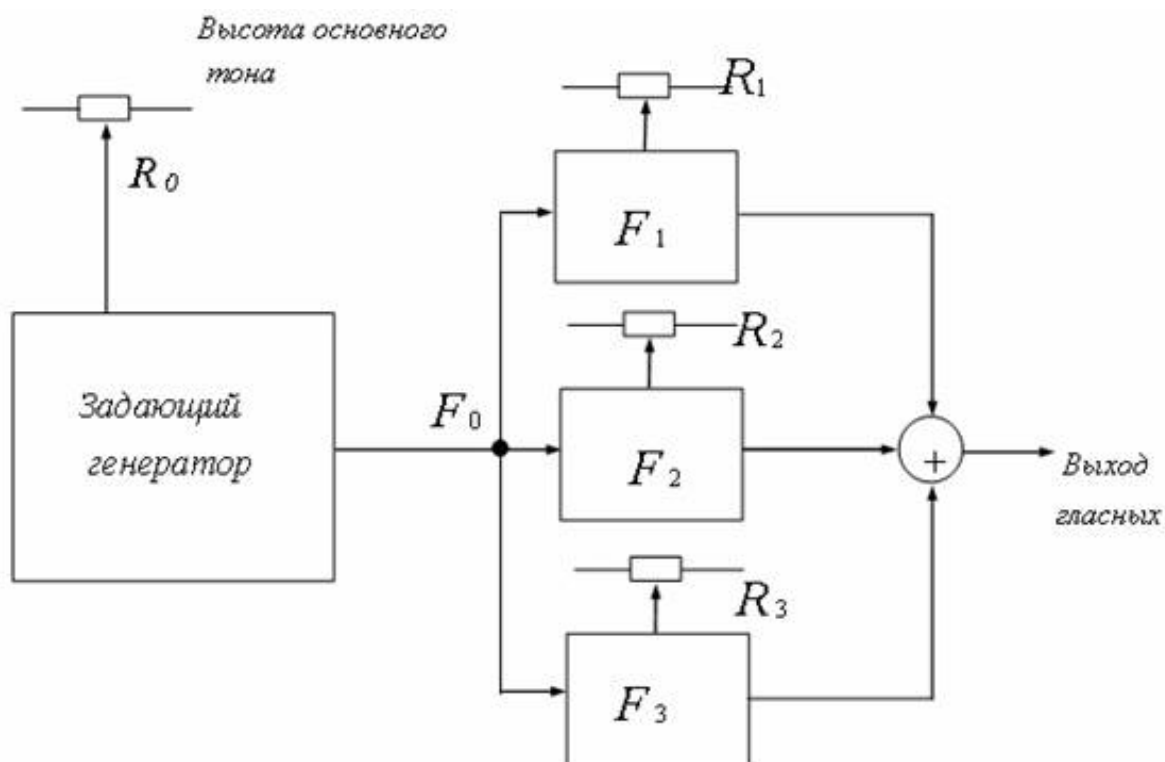


Рис.1.1.5. – Структурная схема формантного синтезатора гласных звуков

Чтобы расширить диапазон синтезатора, необходимо ввести источник шума для формирования взрывных и фрикативных согласных, а также аналог носовой резонансной полости, имитирующий носовые согласные. Структура этого расширенного формантного синтезатора приведена на рис.1.1.6 [6].

Структура полного формантного синтезатора речи (рис.1.1.6) усложняется не очень сильно, по сравнению с синтезатором гласных звуков. Значительно увеличилось количество регулировок в схеме. Три из них служат для управления амплитудой фрикативных, гласных и носовых звуков, один - для регулировки высоты тона, а пять остальных - для регулирования частот

различных резонансов. Применяв в качестве устройства управления регуляторами микропроцессор с соответствующим количеством портов ввода-вывода мы получим устройство, способное производить все необходимые регулировки со скоростью, достаточной для приемлемого приближения к нормальной речи человека.

Естественно, что чем больше обращений к справочной таблице будет производить микропроцессор по каждой фонеме, тем большей плавностью будет отличаться синтетическая речь и тем ближе она будет к естественной человеческой речи.

Преимущество формантного метода синтеза - в его универсальности (т.е. возможность иметь неограниченный словарь) так как здесь речь создается из отдельно генерируемых звуков. Правильно расставив звуки, можно произнести любое слово.

Универсальность эта, однако, не дается бесплатно - за нее приходится расплачиваться ухудшением разборчивости речи. Без соответствующей подготовки трудно понять, что говорит синтезатор.

Дополнительные трудности при реализации большого словаря создает множество имеющихся исключений из правил написания и произношения слов. Если проанализировать одну и ту же фонему, встречающуюся в различных словах, то может оказаться несколько вариантов произношения данной фонемы. Вариации произносимых фонем называют аллофонами. Аллофоны подразделяются на комбинаторные и позиционные. Комбинаторные оттенки обусловлены соседством данной фонемы с другими фонемами и являются следствием наложения одного звука на другой. Позиционные оттенки обусловлены положением фонемы в слове или фразе по отношению к ударному слогу, концу и началу слова и т.д.

Учет всех факторов позволяет оценить общее число аллофонов, необходимое для качественного синтеза русской речи. Общее число аллофонов гласных $N_A^G = 480$ и согласных $N_A^C = 8800$.

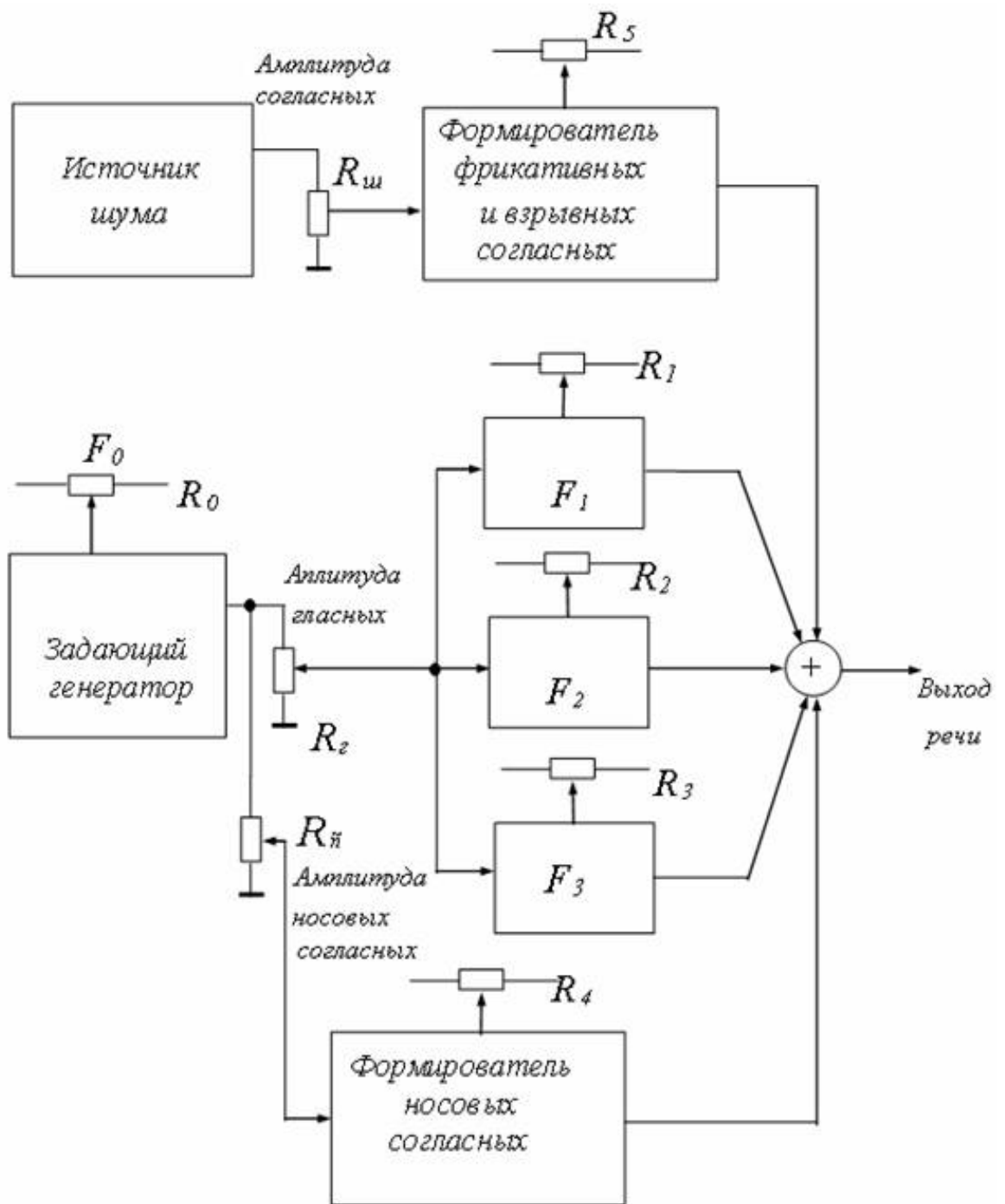


Рис.1.1.6. – Структурная схема формантного синтезатора речи

Другой класс лингвистических понятий, учет которых исключительно важен при создании систем синтеза речи, составляют интонация и ударение. Физически интонация и ударение реализуются совокупностью акустических средств (просодикой), к числу которых относятся:

- 1) мелодика (движение частоты основного тона голоса);
- 2) ритмика (текущее изменение длительности звуков и пауз);
- 3) энергетика (текущее изменение силы звука).

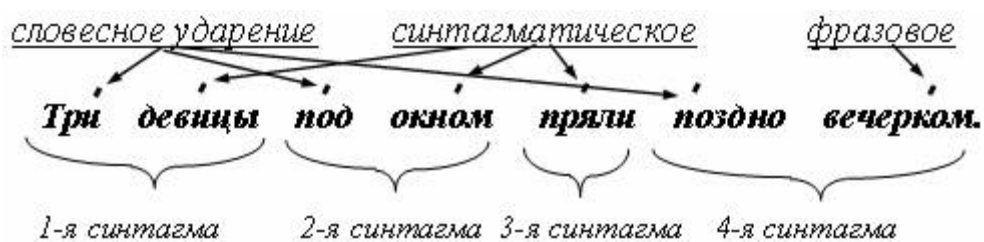
Этап преобразования печатного текста в последовательность фонем должен сопровождаться выделением информации, необходимой для задания просодических характеристик синтезируемых речевых сигналов.

Для этой цели текст анализируется и по определенным правилам разбивается на основные единицы: фраза, синтагма, акцентная группа, фонетическое слово.

Эти единицы маркируются, соответственно фразовым, синтагматическим, групповым и словесным ударениями. Каждой синтагме присваивается один из возможных интонационных типов. Это завершенность, незавершенность, вопрос или восклицание.

Под синтагмой понимают слово (или группу слов), представляющее собой цельную синтаксическую интонационно-смысловую единицу.

Например:



Таким образом, в качестве входной информации текстового сообщения используется размеченный орфографический текст, т.е. обычный орфографический текст с проставленными знаками словесного, синтагматического и фразового ударений.

Фонемный цифровой синтез. Широкое распространение в настоящее время получил фонемный цифровой синтез, основанный на генерации фонем и дальнейшей компиляции из них слов и фраз. Этот подход сочетает компактность цифровых методов обработки с гибкостью управления основными параметрами речи, что свойственно формантным моделям. Процесс кодирования необходимого словаря здесь заменен компиляцией произвольных сообщений из набора заранее закодированных элементов речи.

Фонемный цифровой синтезатор (рис.1.1.7) содержит три уровня обработки. На первом уровне происходит перекодировка последовательности

орфографических символов, представленных в закодированном виде в последовательность кодов фонем; на втором рассчитывается набор акустических параметров, который служит для управления третьим уровнем - формированием выходного речевого сигнала. Текст преобразуется в фонемную форму двумя способами: 1) если входное слово имеется в словаре системы, то оно однозначно переводится в последовательность фонем; 2) если же слово отсутствует в словаре, то производится преобразование по правилам. Поскольку все правила имеют исключения, а исключения по существу, требуют собственных словарей, то преобразования по правилам могут дать ошибку в произношении. Поэтому в словарь вносятся сокращения и ходовые иностранные слова.

На втором уровне обработки вычисляется набор управляющих параметров. Этими параметрами являются формантные частоты (F_1, F_2, F_3) и полосы ($\Delta F_1, \Delta F_2, \Delta F_3$), частота основного тона F_0 и амплитуда вокализации. Набор параметров обновляется каждые 6,4 мс. Такой интервал позволяет хорошо отслеживать самые быстрые переходы между фонемами.

Скорость передачи информации от второго уровня к третьему составляет 45 тыс.бит/с.

Синтез речи осуществляется на третьей стадии. Сигналы возбуждения (гармонический и шумовой) фильтруются формантными резонансными фильтрами голосового тракта.

Частота выборок сигнала равна 10 кГц, что позволяет передавать частотные составляющие до 5 кГц. Структура данного синтезатора реализована на цифровом процессоре сигналов TMS 32010 фирмы «Техас инструментс» .

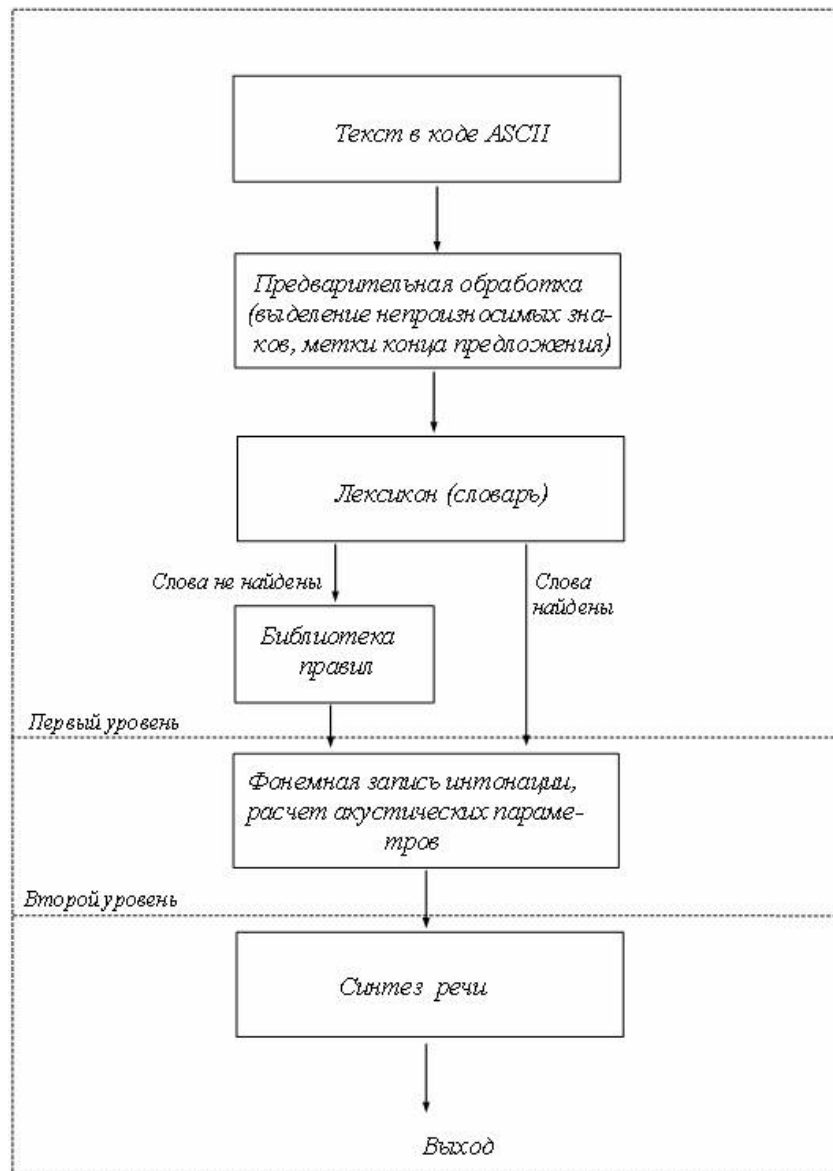


Рис.1.1.7. – Структурная схема цифрового фонемного синтезатора

Рассмотрим основные особенности передачи сигналов изображения в цифровой форме. Обобщенная структурная схема канала цифровой передачи TV-сигнала представлена на схеме рис.1.1.8.

Сигнал с выхода видеокамеры поступает на кодер видеосигнала, где преобразуется в цифровую форму. Затем цифровой сигнал поступает на вход кодера канала, где осуществляется помехоустойчивое кодирование, и далее на вход радиопередающего устройства. Радиосигнал передается по линии связи, и на приемной стороне радиоканала производятся обратные операции, связанные с получением переданного изображения.

Рассмотрим основные особенности каждого из функциональных узлов, приведенных на схеме рис.1.1.8. Заметим что, выбор кодера видеосигнала не зависит от типа линии связи, в то время как выбор типа кодера канала и метода модуляции в радиопередающем устройстве зависят от специфики линии.

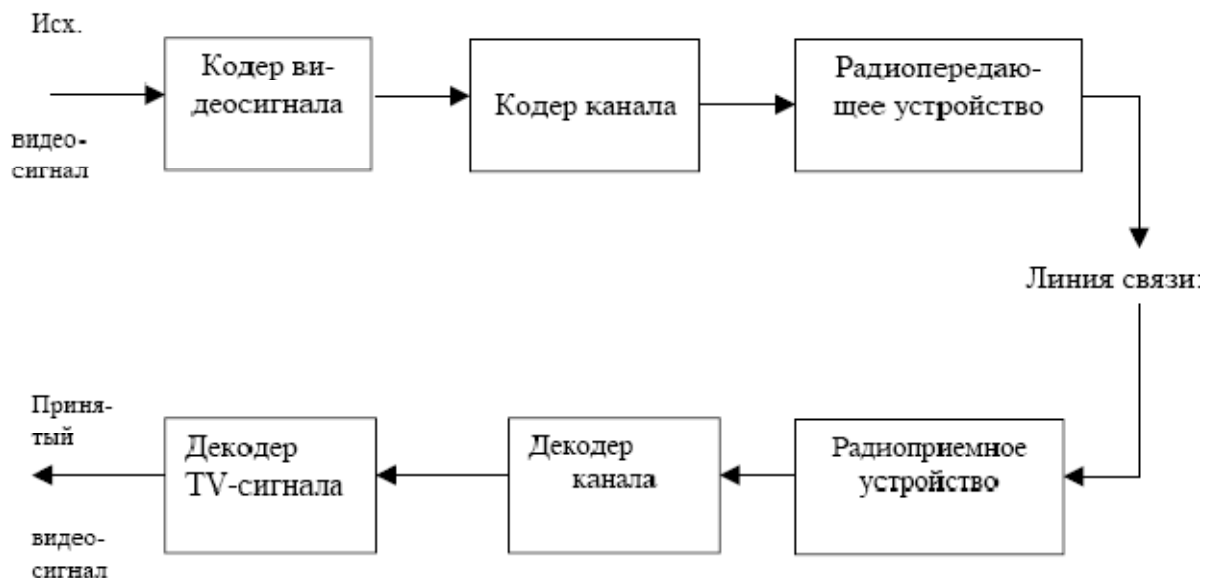


Рис.1.1.8. – Обобщенная структурная схема канала цифровой передачи ТВ-сигнала [7]

Кодер видеосигнала выполняет процедуру сжатия (видеокомпрессию). На входе кодера включается аналого-цифровой преобразователь, где выполняются операции дискретизации по времени и квантования по уровню. По теореме отсчетов частота дискретизации f_d должна быть не менее, чем в два раза выше верхней частоты в спектре передаваемого сигнала $f_{гр}$, т.е. $f_d \geq 2 f_{гр}$. Для принятого в России стандарта при верхней частоте яркостного сигнала изображения 6 МГц, $f_d \geq 12$ МГц. По соображениям унификации параметров цифрового потока для различных стандартов разложения изображения на строки (625 или 525) принята частота дискретизации $f_d=13,5$ МГц.

При квантовании по уровню используется восьмиразрядное кодирование (256 уровней). Для передачи цветного изображения используются два цветоразностных сигнала. Частота дискретизации этих сигналов составляет .

fd/2. Такой стандарт цифровой передачи изображения обозначают 4:2:2. Итоговая скорость передачи цифрового потока для этого стандарта равна:

$$V=13,5 \times 8 + 2 \times 6,75 \times 8 = 216 \text{ Мбит/сек.} \quad (1)$$

Очевидно, что передача изображения с такой скоростью по каналу VHF и UHF² диапазонов не представляется возможной, поэтому используется сжатие изображения. Очевидно, что эта процедура является главной при кодировании видеосигнала.

Телевизионное изображение в цифровом представлении является набором значений интенсивностей светового потока, распределенных на экране. Интенсивность излучаемой световой энергии с единицы поверхности экрана с координатами (x,y) можно представить числом $V(x,y)$. Единичный элемент изображения, характеризуемый (x,y), называется пикселом, а величина $V(x,y)$. яркостью. Задача цифрового представления изображения заключается в представлении $V(x,y)$ конечным количеством двоичных символов. Плотность размещения пикселей связана с пространственным разрешением, определяемым количеством различаемых линий на единице длины.

Замена исходных пикселей квантованными значениями приводит к появлению «снега» на изображении. Экспериментально доказано, что эти искажения практически незаметны при

8-битовой квантовании. Сжатие изображений возможно благодаря их значительной избыточности.

При этом различают:

- структурную избыточность;
- статистическую избыточность;
- психофизическую визуальную избыточность.

Структурная избыточность определяется TV-стандартом, где определены $f_{гр}$, число

строк/полей в секунду и т.д. При цифровой передаче изображения нет необходимости использования кадровых и строчных гасящих импульсов, т.е.

²VHF – Very High Frequency – диапазон УКВ; UHF – UltraHighFrequency - диапазон ДМВ

объем информации, передаваемой в секунду, можно сократить за счет исключения пассивной части полного TV- сигнала. Для $f_{гр}=6$ МГц уменьшение скорости передачи за счет устранения структурной избыточности составляет около 25%.

Статистическая избыточность изображения связана с корреляцией, а, следовательно, и предсказуемостью данных между пикселями в одной строке, в соседних строках, соседних кадрах. Эта избыточность может быть устранена без потери информации, а исходное изображение полностью восстановлено. Так, большая часть изображения одного кадра приходится на фон или задний план, где яркость постоянна или мало меняется в пространстве. Изображения в соседних кадрах также малоразличимы даже при наличии движущихся изображений. Тогда, зная распределение яркости в одной строке (кадре), можно с высокой вероятностью предсказать распределение яркости в следующей строке (кадре).

Визуальная (психофизическая) избыточность связана с использованием свойств зрения, позволяющих устранить часть информации об изображении, не существенно влияющей на его восприятие. Так, шумы квантования различаются глазом на крупных деталях изображения в виде ложных контуров, однако малозаметны на мелких деталях и на резких перепадах яркости. Это позволяет вводить более грубое квантование видеосигнала в окрестностях переходов яркости и высокочастотных компонент. Устранение визуальной избыточности изображений является основным резервом при сжатии изображения. К примеру, цветовая разрешающая способность в 4 раза меньше, чем по сигналу яркости. Рассмотрим теперь различные методы устранения статистической и визуальной избыточности изображений, сочетание которых позволяет получить практические результаты. На схеме рис.1.1.9. представлена классификация методов кодирования изображений.

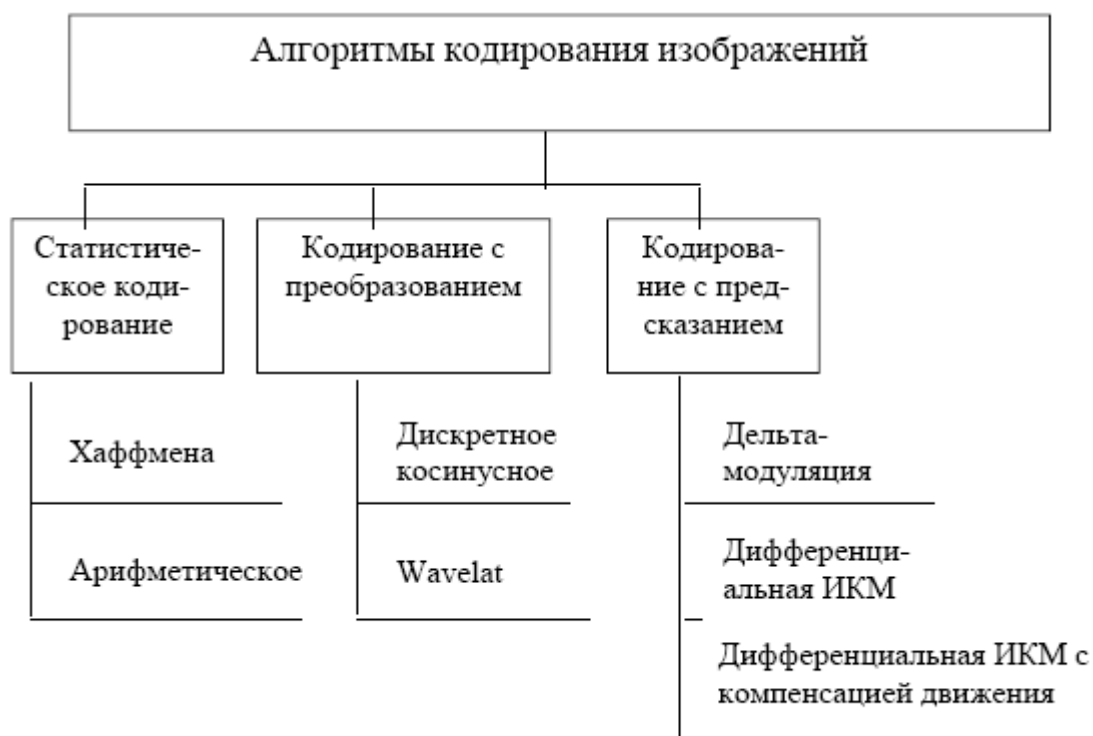


Рис.1.1.9. – Классификация методов кодирования изображений [7]

Наибольшее распространение при статистическом кодировании получил код Хаффмена.

Принцип использования этого кода заключается в том, что коды пикселя, имеющие большую вероятность, передаются меньшим количеством бит. Более редко встречающиеся коды передаются словом, содержащим большее число бит. Практически это выглядит следующим образом. Для всех 256 уровней квантования определяется гистограмма, содержащая вероятности появления каждого из уровней, которые располагаются по убыванию. Далее выполняется процедура, при которой наиболее вероятным уровням ставятся в соответствие двухбитовые комбинации, менее вероятным – трехбитовые комбинация и т.д. Вычисление гистограммы проводится как на передающей, так и на приемной стороне по мере поступления данных.

При 8-битовом квантовании использование кода Хаффмена позволяет сжать изображение не более чем в 8 раз. Этот метод кодирования применяют совместно с другими методами сжатия при практической реализации аппаратуры цифровой передачи изображений.

Принцип арифметического кодирования заключается в следующем. До начала кодирования сообщению ставится в соответствие интервал $[0,1]$. Каждому символу в сообщении отводится интервал, ширина которого равна вероятности появления этого символа. В сумме длины всех участков составляют 1. Поступающие символы уменьшают размер интервала в соответствии с моделью вероятности символов. Результатом кодирования является число x с очень высокой точностью его определения. На приемной стороне формируется исходная шкала вероятностей на интервале $[0,1]$ и переданное число x попадает на интервал, характеризующий символ, поступивший в кодер на данном этапе. Этот метод считается наиболее эффективным среди статистических методов. Однако метод очень требователен к вычислительным ресурсам, использующим нецелочисленную арифметику.

Эффективным методом сжатия изображения является групповое кодирование с преобразованием. При этом преобразование проводится сразу над группой пикселей в пределах кадра (или поля) изображения. Массив отсчетов изображения трансформируется в матрицу коэффициентов. Преимущества кодирования, получаемые в результате преобразования коэффициентов, заключаются в следующем:

- значительное количество элементов матрицы коэффициентов имеют нулевое значение;
- декорреляция связей между пикселями, достигаемая в результате преобразования, повышает эффективность статистического кодирования;
- нелинейное квантование коэффициентов, учитывающее психофизические особенности визуального восприятия искажений, дополнительно позволяет сократить объем передаваемой информации без заметного изменения качества изображения.

Преобразование изображения следует рассматривать как его разложение в обобщенный двумерный спектр по базисным функциям, где амплитуда каждой спектральной составляющей характеризует яркость. При этом разложение может проводиться по различным ортогональным функциям. Здесь

рассмотрим наиболее известные преобразования, используемые в реальной аппаратуре.

Дискретное косинусное преобразование (ДКП)

Является модификацией дискретного преобразования Фурье и обладает полезными свойствами. Во-первых, матрица ДКП хорошо аппроксимирует матрицу оптимального декоррелирующего преобразования Карунена . Лоэва и обладает практически такой же эффективностью, как оптимальное преобразование. Во-вторых, ДКП реализуется с помощью быстрых преобразований, что существенно снижает вычислительные затраты на его реализацию по сравнению с оптимальным преобразованием.

При использовании ДКП обработка ведется блоками 8x8 пиксел. В среднем размер блока соответствует интервалу корреляции элементов изображения. В результате выполнения ДКП формируется матрица из 64 коэффициентов, характеризующих пространственные частоты (двумерные x,y) функции яркости изображения:

$$F_{uv} = \frac{4}{2n-1} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} C(j)C(k)X_{jk} \cos\left(\frac{2\pi jv}{2n-1}\right) \cos\left(\frac{2\pi kv}{2n-1}\right), \quad (1.1.1)$$

$$\text{где } C(j) = \begin{cases} 0,5 & m = 0 \\ 1 & m = 1, \dots, n-1 \end{cases}$$

Здесь X_{jk} - элемент матрицы изображения, соответствующий яркости пиксела, с координатами j, k

F_{uv} - элемент матрицы ДКП с координатами u, v

n - количество элементов в строке, столбце, обычно $n=8$.

После операции ДКП коэффициенты могут принимать не целые значения, однако дальнейшая обработка требует квантованного представления этих коэффициентов. Проведение процедуры квантования коэффициентов следует проводить с учетом психофизических особенностей зрения. Как уже отмечалось, зрительное восприятие допускает большой уровень шумов и ошибок квантования в области изображений с большим уровнем высокочастотных компонент. Это означает, что коэффициенты этих компонент

можно квантовать на малое число уровней, в пределах на два. Постоянная составляющая и амплитуды низкочастотных компонент квантуются на большое число уровней и передаются с высокой точностью. Финальной операцией при квантовании является Z-упорядочивание, при котором оставшиеся коэффициенты выстраиваются в последовательности возрастания пространственных частот. Если пространственные частоты одинаковы, то впереди ставятся коэффициенты для меньших вертикальных частот. Из ряда коэффициентов образуются пары чисел, одно из которых равно ненулевому коэффициенту, а другие – числу предшествующих этому элементу нулей. Далее эти сочетания чисел кодируются кодом Хаффмена. Упрощенная структурная схема алгоритма внутрикадрового кодирования и декодирования на основе ДКП приведена на схеме рис.1.1. 10.

При этом преобразовании, также как и при ДКП, осуществляется переход из плоскости изображения в двумерную частотную область. В отличие от ДКП это переход осуществляется с помощью гребенки цифровых фильтров. Такой подход позволяет отказаться от блочной структуры преобразования и устранить основной недостаток ДКП, связанный со скачкообразным изменением шумов квантования при переходе от одного блока ДКП к другому. Блочная структура шумов квантования более ярко проявляется при больших степенях сжатия, представляющих наибольший интерес в приложениях к специальной технике.

Техническая трактовка Wavelat-преобразования может быть представлена на схеме рис.1.1.11.

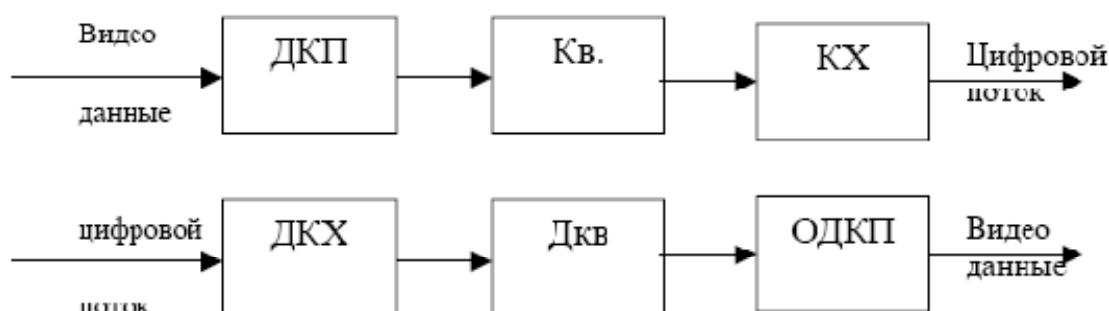


Рис.1.1.10. – Упрощенная структурная схема алгоритма внутрикадрового кодирования и декодирования на основе ДКП [7]

Здесь приняты обозначения:

Кв. - квантователь; КХ - кодер Хаффмена; ДКХ-. декодер Хаффмена;

Дкв - деквантователь; ОДКП - обратное дискретное косинусное преобразование.

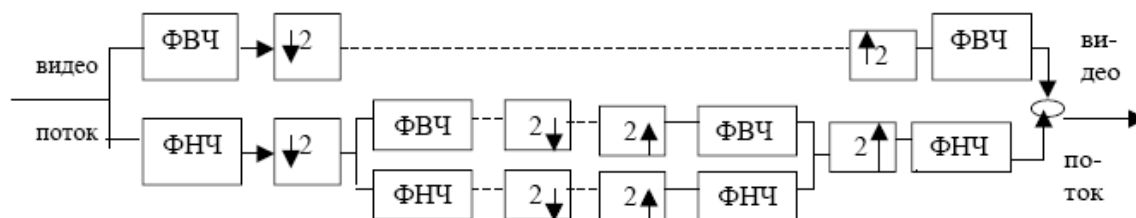


Рис.1.1.11. – Техническая трактовка Wavelat-преобразования [7]

Сигнал изображения разделяется по спектру на две равные части с помощью фильтров нижних и верхних частот. Поскольку НЧ- и ВЧ-компоненты имеют полосу в два раза меньше, чем у исходного сигнала, то их частота дискретизации может быть понижена в два раза. Входной сигнал схемы рис.1.1.11 является цифровым, поэтому после цифровой НЧ- и ВЧ- фильтрации производится децимация (исключение каждого второго отсчета). После каждого из трех этапов преобразования изображение делится на одну низкочастотную и три высокочастотные компоненты. Первая компонента отображает горизонтальные высокочастотные составляющие изображения, вторая компонента отображает вертикальные составляющие, третья высокочастотная компонента связана с диагональными пространственными частотами и отображает яркостные переходы. Поскольку после каждой процедуры фильтрации количество отсчетов на выходе фильтра уменьшается в два раза, то результирующее количество отсчетов на выходе всей гребенки фильтров в точности равно количеству отсчетов в исходном изображении. Таким образом, при W-преобразовании кадр изображения переводится в частотную область, где описывается тем же количеством отсчетов, как и до преобразования. Для кодирования спектральных отсчетов изображения, полученных в результате W-преобразования, используются те же принципы, что и при ДКП.

Для восстановления исходного изображения (правая часть сечения рис.1.1.11.) каждая компонента растягивается, т.е. после каждого значащего отсчета вставляется нулевой отсчет. Затем осуществляется аналогичная рассмотренной выше НЧ и ВЧ фильтрация, в результате которой нулевые отсчеты заменяются интерполированными.

Структурная схема алгоритма внутрикадрового сжатия на основе W-преобразования аналогична схеме рис.1.1.10.

Кодирование с предсказанием. В результате дискретизации изображения и квантования по уровню формируются 8-битные слова для представления каждого пиксела. При этом представлении используется импульсно-кодовая модуляция (ИКМ). Однако кодирование на основе ИКМ не позволяет сократить объем исходной информации, т.к. оно не учитывает корреляционные связи между пикселами. Более эффективной является дифференциальная ИКМ (ДИКМ). Структурная схема, поясняющая алгоритм ДИКМ, представлена на схеме рис.1.1.12.



Рис.1.1.12. – Структурная схема, поясняющая ДИКМ [7]

При ДИКМ вместо передачи значения X яркости n -го пиксела передается разность $\varepsilon_n = X_n - \hat{X}_n$ между X_n и его оценкой \hat{X}_n . Величина ε_n называется ошибкой предсказания. Оценка \hat{X}_n формируется с использованием предсказателя, алгоритм работы которого должен быть согласована со статистикой передаваемого изображения. Наибольшее распространение получили методы линейного предсказания. Оценка \hat{X}_n определяется по формуле:

$$\hat{X}_n = \sum_{i=0}^k \alpha_i \hat{X}_{k-i}, \quad (1.1.2)$$

где α_i - коэффициенты, получаемые на основе корреляционных связей между пикселями.

Коэффициенты α_i в выражении (1.1.2) подбираются исходя из функции корреляции пикселей таким образом, чтобы минимизировать среднеквадратическое значение ошибки предсказания ε_n . Диапазон изменения ε_n существенно меньше диапазона изменения исходных пикселей изображения. Поэтому при одной и той же ошибке квантования для передачи ε_n требуется меньшее количество уровней квантования, а следовательно и меньшее количество бит для кодирования. На некоторых изображениях, например, фон одинаковой яркости, величина ошибки может быть вообще равна нулю и для передачи таких изображений скорость цифрового потока снижается практически до нуля.

Для кодирования простых изображений используется дельта - модуляция (ДМ). При ДМ предсказание производится только по одному пикселу, а квантование ошибки предсказания производится на два уровня, т.е. каждый пиксел кодируется одним битом вместо 8.

На практике метод ДИКМ применяется для межкадрового кодирования. В стандартном телевизионном сигнале кадры следуют с частотой 25 Гц, т.е. очередной кадр передается через 40 мс. За столь короткий интервал времени изображение в соседних кадрах практически не успевает измениться. Поэтому весьма эффективно кодировать разность между пикселями изображения, имеющими одинаковые координаты в соседних кадрах. В этом случае ошибка предсказания определяется по формуле:

$$\varepsilon_k = B_k(x, y) - \hat{B}_k(x, y) \quad (1.1.3)$$

где k - номер кадра; $B_k(x, y)$ - значение пиксела с координатами x, y ;

$\hat{B}_k(x, y)$ - предсказанное значение величины $B_k(x, y)$

В пределах кадра используются методы кодирования с преобразованием, рассмотренные выше. Упрощенная структурная схема алгоритма межкадрового кодирования и декодирования приведена на схеме рис.1.1.13.

Здесь для формирования вектора оценки \hat{X} в обратной связи использует деквантователь и обратный дискретный косинусный преобразователь, которые восстанавливают блок изображения точно так же, как и в декодере.

При передаче сюжета происходит изменение положения отдельных частей кадра из-за их движения, это уменьшает межкадровую корреляцию и снижает точность предсказания. Таким образом, возникает задача оценки вектора движения и его компенсации. Такая оценка получается проецированием блока на опорное изображение и сравнением со всеми блоками предыдущего кадра. Смещение между проецируемым блоком и опорным изображением по осям x и y определяет вектор движения. Эта информация используется для предсказания следующего кадра. Пр-предсказатель; остальные обозначения приведены в комментариях к схеме рис.1.1.11

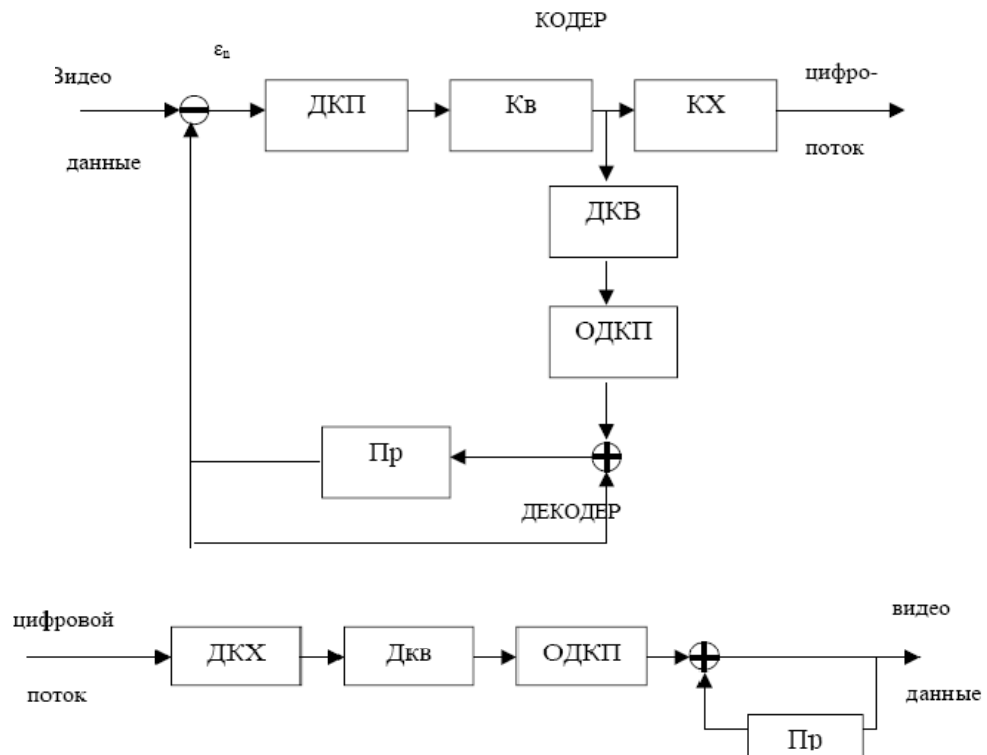


Рис.1.1.13. – Упрощенная структурная схема алгоритма межкадрового кодирования и

Структурная схема алгоритма кодирования с компенсацией движения приведена на схеме рис.1.1.14.

Кодер по алгоритму схемы рис.1.1.14 работает аналогично кодеру схемы рис.1.1.13.

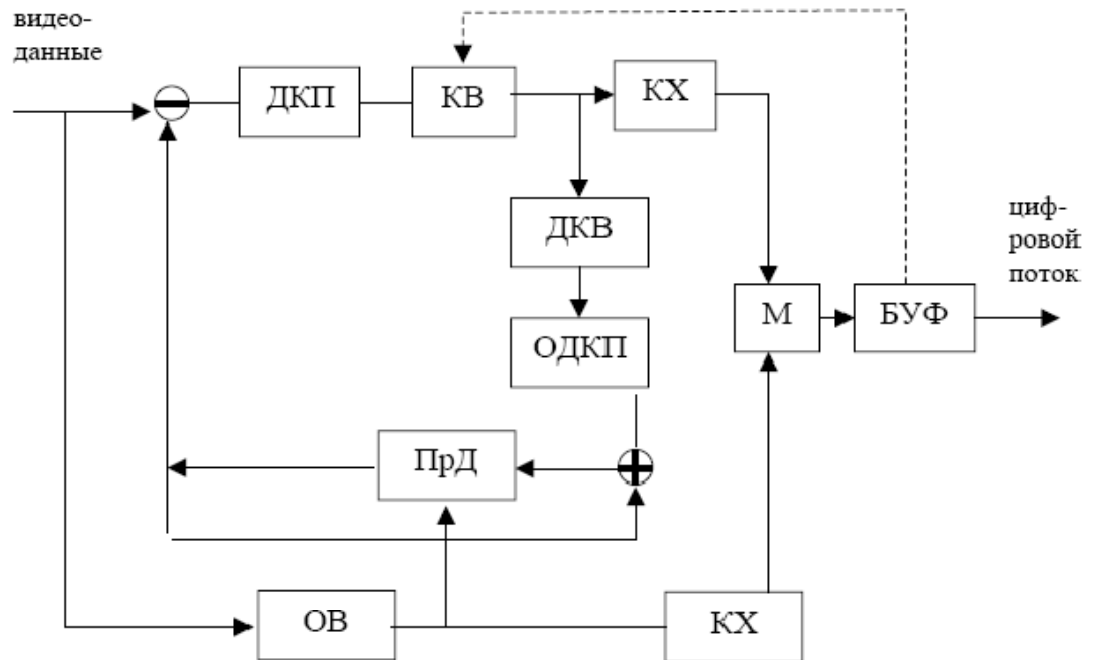


Рис. 1.1.14. – Структурная схема алгоритма кодирования с компенсацией движения [7]

Здесь предсказатель работает с учетом информации, полученной от блока оценки вектора движения -ОВ. Вектор движения кодируется кодом Хаффмена, мультиплексируется и передается вместе с основным потоком через буфер . БУФ, от которого имеется обратная связь на квантователь. Эта связь обеспечивает постоянную скорость цифрового потока. При возрастании уровня детальности изображения (высокочастотных компонент) число ненулевых элементов матрицы увеличивается и возрастает скорость цифрового потока, следовательно буфер заполняется с повышенной скоростью. Обратная связь делает квантование более грубым и скорость поступления данных уменьшается. Если кодируется простое изображение с малой детальностью, то скорость потока становится ниже среднего значения, т.к. число нулевых элементов матрицы коэффициентов ДКП увеличивается. Тогда с помощью

обратной связи число уровней квантования увеличивается. Буфер выполняет функцию постоянной времени цепи регулирования.

Перечислим теперь кратко существующие стандарты цифровой передачи, пригодные для решаемой задачи.

Стандарт JPEG (Joint Photographic Experts Group). Стандарт разработан для использования в системах компьютерной обработки, т.е. для передачи неподвижных изображений. Кодер JPEG устраняет пространственную статистическую избыточность и выполняет следующие задачи:

- дискретное косинусное кодирование;
- квантование;
- статистическое кодирование.

Стандарт MPEG (Moving Picture Experts Group). Общее название стандартов передачи динамических изображений. Кодер MPEG устраняет статистическую избыточность (пространственную и временную), а также и визуальную избыточность. Решает следующие задачи:

- внутрикадровое кодирование;
- дискретное косинусное преобразование;
- квантование;
- статистическое кодирование;
- межкадровое кодирование;
- дифференциальная ИКМ;
- использование вектора движения при предсказании.

Возможна также реализация MPEG с применением Wavelet-преобразования для внутрикадрового сжатия. Последний вариант представляется наиболее перспективным для использования в специальной технике передачи изображений.

1.2. Особенности помехоустойчивого кодирования в телекоммуникационных системах

Помехоустойчивое кодирование представляет собой класс преобразований сигнала, выполняемых для повышения качества связи. Тему

помехоустойчивого кодирования можно условно разделить на две части: кодирование сигнала и структурированные последовательности. Кодирование сигнала означает преобразование сигнала в некий «улучшенный» сигнал, позволяющий сделать процесс обнаружения менее подверженным ошибкам. Метод структурированных последовательностей – это преобразование последовательности данных в новую, обладающую структурной избыточностью. Избыточные биты служат для определения и исправления ошибок.

Кодирование сигнала. Процедура кодирования состоит в преобразовании набора сигналов в усовершенствованный набор. Наиболее популярными из таких кодов являются ортогональные и биортогональные коды. Взаимная корреляция между двумя сигналами является мерой расстояния между двумя векторами сигналов в сигнальном пространстве. Набор сигналов $\{s_i(t)\}$ состоит из ортогональных сигналов тогда и только тогда, когда для коэффициента взаимной корреляции двух сигналов выполняется равенство:

$$z_{ij} = \frac{(\text{количество совпавших цифр}) - (\text{количество несовпавших цифр})}{\text{общее количество цифр в последовательности}} = \begin{cases} 1; & i = j \\ 0; & i \neq j \end{cases} \quad (1.2.1)$$

Набор однобитовых данных можно преобразовать с помощью ортогональных кодовых слов, состоящих из двух разрядов каждое, которые описываются строками показанной ниже матрицы H_1 .

Набор данных	Набор ортогональных кодовых слов
0	$H_1 = \begin{bmatrix} 0; 0 \\ 0; 1 \end{bmatrix}$
1	

Проверка ортогональности набора производится с помощью уравнения (1.2.1).

Для кодирования двухбитовых данных вышеупомянутый набор нужно расширить по горизонтали и вертикали, что дает матрицу H_2 .

Набор данных	Набор ортогональных кодовых слов
--------------	----------------------------------

$$\begin{array}{l}
 0;0 \\
 0;1 \\
 \dots \\
 1;0 \\
 1;1
 \end{array}
 H_2 = \begin{bmatrix}
 0;0;0;0 \\
 0;1;0;1 \\
 \dots \\
 0;0;1;1 \\
 0;1;1;0
 \end{bmatrix} = \begin{bmatrix}
 H_1; H_1 \\
 H_1; \overline{H_1}
 \end{bmatrix}$$

Вообще для набора k -битовых данных из матрицы H_{k-1} можно построить набор кодовых слов H_k размерностью $2^k \times 2^k$, который называется матрицей Адамара..

$$H_k = \begin{bmatrix}
 H_{k-1}; H_{k-1} \\
 H_{k-1}; \overline{H_{k-1}}
 \end{bmatrix} \tag{1.2.2}$$

Каждая пара слов в каждом наборе кодовых слов $H_1; H_2; H_3; \dots; H_k$ содержит одинаковое количество совпадающих и несовпадающих разрядов, поэтому в соответствии с уравнением (1.2.1), каждый из этих наборов ортогонален.

Биортогональный набор сигналов, состоящий из M кодовых слов, получается из ортогонального набора, состоящего из $\frac{M}{2}$ сигналов путем дополнения последнего сопряженными значениями каждого сигнала.

$$B_k = \begin{bmatrix}
 B_{k-1} \\
 \overline{B_{k-1}}
 \end{bmatrix}. \tag{1.2.3}$$

Например, набор двухбитовых данных можно преобразовать в биортогональный набор кодовых слов следующим образом:

Набор данных	Набор ортогональных кодовых слов
0;0	0;0
0;1	0;1
...	...
1;0	1;1
1;1	1;0

$$B_2 = \begin{bmatrix}
 0;0 \\
 0;1 \\
 \dots \\
 1;1 \\
 1;0
 \end{bmatrix}$$

В действительности биортогональный набор состоит из двух ортогональных кодов, таких, что для каждого кодового слова в одном наборе имеется антиподное ему слово в другом. Биортогональный набор состоит из

комбинации ортогональных и антиподных сигналов. Если использовать коэффициенты взаимной корреляции, то биортогональные коды можно представить следующим образом:

$$z_{ij} = \begin{cases} 1; i = j \\ -1; i \neq j, |i - j| = M/2 \\ 0; i \neq j, |i - j| \neq M/2 \end{cases} \quad (1.2.4)$$

Преимуществом биортогональных кодов перед ортогональными является то, что при передаче аналогичной информации размер кодового слова биортогональных кодов вдвое меньше кодового слова ортогональных кодов. Следовательно, при использовании биортогональных кодов требуется вдвое меньшая полоса пропускания. Кроме того, поскольку антиподные векторы имеют лучшие пространственные характеристики, чем ортогональные, биортогональные коды снижают вероятность битовой ошибки по сравнению с ортогональными кодами.

Код, получаемый из ортогонального путем удаления первого разряда каждого кодового слова, называется трансортогональным или симплексным кодом. Такой код описывается коэффициентом взаимной корреляции:

$$z_{ij} = \begin{cases} 1; i = j \\ \frac{-1}{M-1}; i \neq j \end{cases} \quad (1.2.5)$$

Сравнивая достоверность передачи ортогонального, биортогонального и симплексного кодов, можно сказать, что симплексный код имеет наименьшее требуемое битовое отношение сигнал/шум $\frac{E_b}{N_0}$ для получения требуемой вероятности символьной ошибки. При больших значениях M все три схемы очень похожи между собой в смысле достоверности передачи. При этом биортогональное кодирование по сравнению с другими методами требует лишь половины полосы пропускания. В то же время для каждого из этих методов требования к полосе пропускания и сложность системы экспоненциально растут с увеличением M , поэтому подобные схемы кодирования годятся лишь тогда, когда доступна значительная полоса пропускания. При наборе

ортогональных кодов, включающем $M = 2^k$ сигналов, требуемая ширина полосы пропускания в $\frac{M}{k}$ раз больше необходимой для передачи некодированного сигнала.

Структурированные последовательности представляют собой методы введения в исходные данные избыточности таким образом, что это позволяет обнаруживать и исправлять ошибки. Структурированные последовательности делятся на три подкатегории: блочные, сверточные и турбокоды.

При использовании блочных кодов исходные данные делятся на блоки из k бит. В процессе кодирования каждый k -битовый блок данных преобразуется в больший блок из n бит, который называется канальным символом. К каждому блоку данных таким образом кодирующее устройство добавляет $(n-k)$ бит, которые называют избыточными битами, битами четности или контрольными битами. Отношение числа избыточных бит к числу информационных бит $\frac{n-k}{k}$ называется избыточностью кода, отношение числа бит данных к общему числу бит $\frac{k}{n}$ называется степенью кодирования.

Коды с контролем четности для обнаружения и исправления ошибок используют линейные суммы информационных битов. Код с одним контрольным битом – это прибавление к блоку информационных битов одного контрольного бита. Этот бит (бит четности) может быть равен нулю или единице, причем его значение выбирается так, чтобы сумма всех битов в кодовом слове была четной или нечетной. В операции суммирования используется арифметика по модулю 2 (операция исключающего ИЛИ). Если бит четности выбирается так, что результат четный, то говорят, что схема имеет положительную четность. Если при добавлении бита четности результирующий блок данных является нечетным, то говорят, что он имеет отрицательную четность.

На рис.1.2.1 показана последовательная передача данных (первым является крайний справа бит). К каждому блоку добавляется один бит четности (крайний слева бит в каждом блоке) дающий положительную четность.

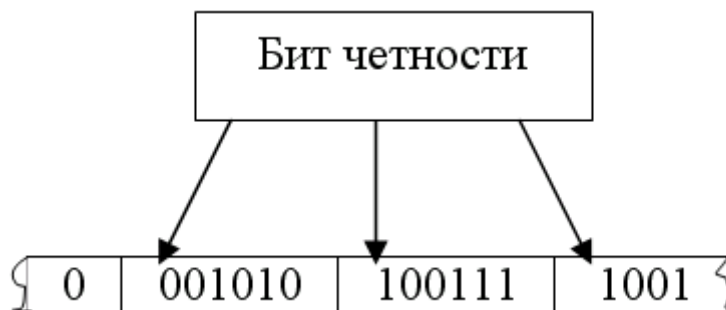


Рис. 1.2.1. – Проверка четности для последовательной структуры кода [8]

В приемном устройстве производится декодирование, заключающееся в проверке, дают ли нуль суммы принятых битов кодового слова по модулю 2 (положительная четность). Если полученный результат равен 1, то кодовое слово содержит ошибки. Декодер не может исправить эту ошибку, он может ее только обнаружить.

Прямоугольный код, называемый также композиционным, можно представить в виде параллельной структуры кода, изображенной на рисунке 1.2.2.

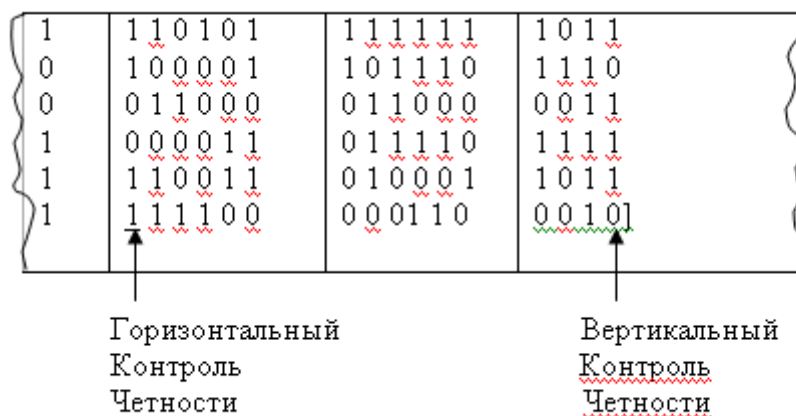


Рис. 1.2.2. – Проверка четности для параллельной структуры кода

Код создается следующим образом. Начале из битов сообщения строятся прямоугольники, состоящие из М строк и N столбцов. Затем к каждой строке и

каждому столбцу прибавляется бит четности, что в результате дает матрицу размером $(M + 1) \times (N + 1)$. Степень кодирования прямоугольного кода можно записать как

$$\frac{k}{n} = \frac{MN}{(M + 1)(N + 1)} \quad (1.2.6)$$

Прямоугольный код может исправить любую единичную ошибку, поскольку расположение такой ошибки однозначно определяется пересечением строки и столбца, в которых была нарушена четность.

Кодирование с коррекцией ошибок можно рассматривать как инструмент, реализующий различные компромиссы системы. На рис. 4.6.3 приведен сравнительный вид двух кривых, описывающих зависимость достоверности передачи от отношения $\frac{E_b}{N_0}$.

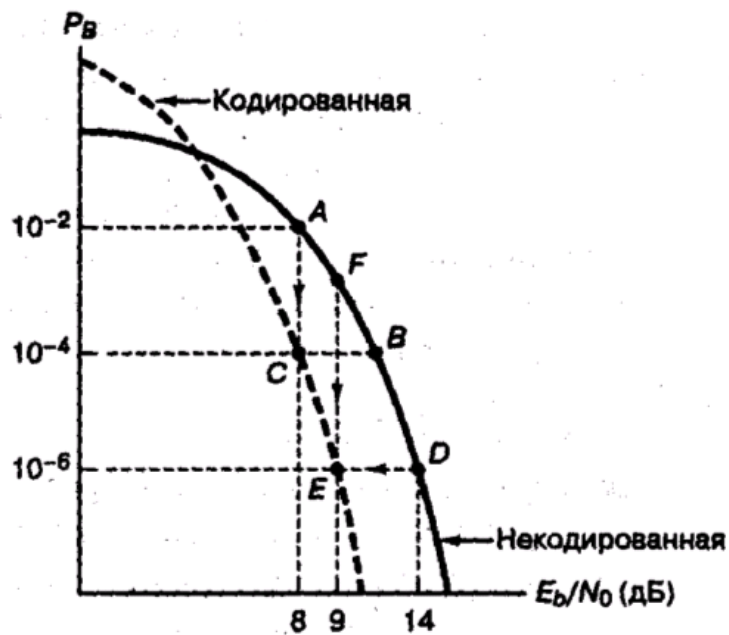


Рис. 1.2.3. – Сравнение типичной достоверности передачи при использовании схемы с кодированием и схемы без кодирования [3]

Компромисс 1: достоверность или полоса пропускания. Пусть разработана простая недорогая система речевой связи, которая была установлена у заказчика. Система не использует кодирование с коррекцией ошибок. Пусть рабочая точка системы совпадает с точкой A на рисунке 3 ($\frac{E_b}{N_0}$

$=8$ дБ, $P_B = 10^{-2}$). После испытаний у заказчика появляются жалобы на качество связи, он полагает, что вероятность битовой ошибки должна быть не выше $P_B = 10^{-4}$. Обычным способом удовлетворения требования заказчика является сдвиг рабочей точки из точки А в точку В. Но допустим, что $\frac{E_b}{N_0} = 8$ дБ – это максимальное значение, возможное в данной системе. Один из возможных выходов из данной ситуации – это сдвиг рабочей точки системы из точки А в точку С. Помимо введения кодера и декодера это приведет к увеличению полосы пропускания системы.

Компромисс 2. Мощность или полоса пропускания. Пусть заказчику установлена система без кодирования с рабочей точкой, совпадающей с точкой D ($\frac{E_b}{N_0} = 14$ дБ, $P_B = 10^{-6}$). Заказчик не имеет претензий к качеству связи, но с помощью данного оборудования затруднительно получить требуемые $\frac{E_b}{N_0} = 14$ дБ, оборудование постоянно работает на грани отказа. Если снизить требования к мощности, то можно сдвинуть рабочую точку из D в E. Плата заключается опять в полосе пропускания.

Компромисс 3. Скорость передачи данных или полоса пропускания. Пусть разработана система без кодирования и с рабочей точкой, совпадающей с точкой D. Пусть нет проблем с качеством связи и нет нужды в снижении уровня мощности. Но у заказчика возросли требования к скорости передачи данных. Если в системе ничего не менять, то это приведет к перемещению рабочей точки из D например, в точку F. Возрастание скорости передачи данных плохо отражается на качестве их передачи. Если же применить кодирование с коррекцией ошибок (переместить рабочую точку из D в E), то можно восстановить утраченное качество, сохраняя при этом прежний уровень мощности за счет увеличения полосы пропускания.

Компромисс 4. Пропускная способность или ширина полосы пропускания. Пропускная способность (максимальное число клиентов при

множественном доступе) обратно пропорциональна значению $\frac{E_b}{N_0}$. Снижение требований к $\frac{E_b}{N_0}$ при применении кодирования дает в итоге увеличение пропускной способности. Платой за это опять же является увеличение полосы пропускания.

При низких значениях $\frac{E_b}{N_0}$ кривые для кодированной и для некодированной системы пересекаются. Это значит, что снижение $\frac{E_b}{N_0}$ ниже какого-то порогового значения ведет к переполнению демодулятора ошибками. При этом наличие избыточных бит не дает никакого выигрыша. Впрочем, класс мощных турбокодов позволяет повысить надежность передачи при низких значениях $\frac{E_b}{N_0}$. Для турбокодов точка пересечения графиков находится значительно ниже.

Линейные блочные коды – это класс кодов с контролем четности, которые можно описать парой чисел (n, k) . В процессе кодирования блок из k символов сообщения преобразуется в больший блок из n символов кодового слова. Для линейных кодов это линейное преобразование.

В течение последних лет наиболее популярной схемой кодирования является сверточная, т.к. она показывает лучшие результаты при той же конструктивной сложности кодера и декодера.

Сверточные коды описываются тремя числами n, k, K . Длина n кодового слова является не только функцией входного сообщения длиной k , но и функцией кодового ограничения K (число разрядов в регистре сдвига-рис.1.2.4).

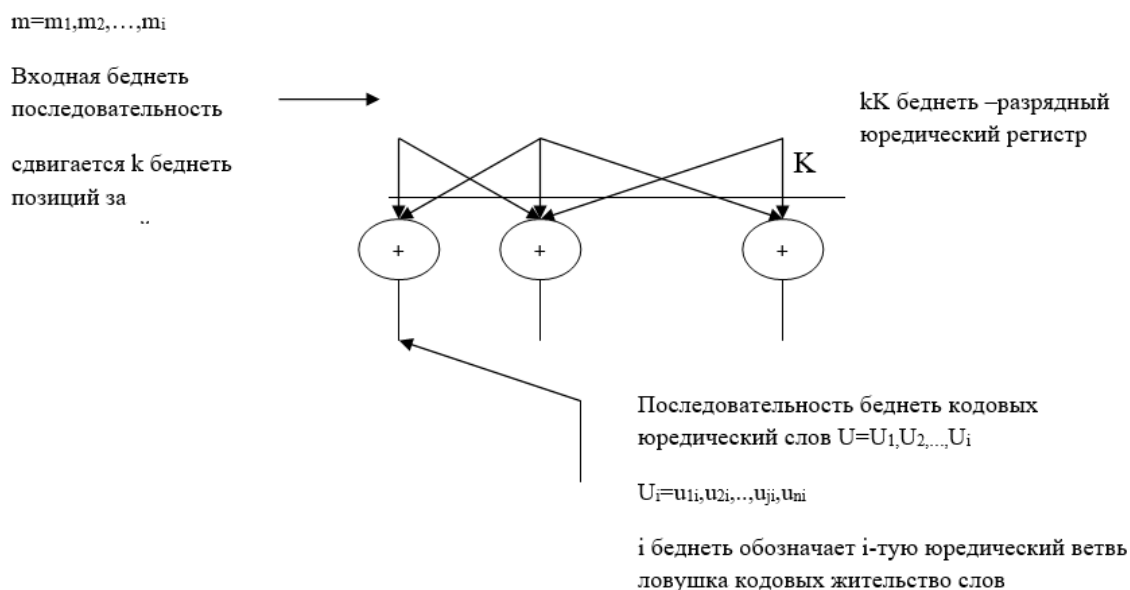


Рис. 1.2.4. – Схема устройства сверточного кодирования [9]

Схема, изображенная на рисунке 1.2.4 дает в каждый момент времени на выходе кодовую последовательность U_i , состоящую из k бит u_i , значение которых вычисляются как линейная функция входных битов. Слово «сверточный» отражает от факт, что данные на выходе генерируются с помощью линейного сложения (т.е. «свертки») смещенных во времени импульсов последовательности на входе с импульсной характеристикой кодера (заполнения регистров предыдущими битами).

Рассмотрим сверточный кодер со степенью кодирования (иногда этот параметр называется скоростью передачи кода, т.к. описывает, в какое число раз увеличивается скорость передачи бит на выходе по сравнению со входом) $\frac{1}{2}$ и величиной кодового ограничения $K=3$, изображенный на рисунке 1.2.5. В этом кодере может возникнуть ограниченное число состояний. Под состоянием в общем смысле понимается наименьшее количество информации, на основе которой вместе с текущими входными данными можно определить данные на выходе системы. Для сверточного кодера со степенью кодирования $\frac{1}{n}$ состояние представляется содержимым $K-1$ крайних правых разрядов кодера и поэтому можно сказать, что будущие состояния ограничиваются прошлыми состояниями.

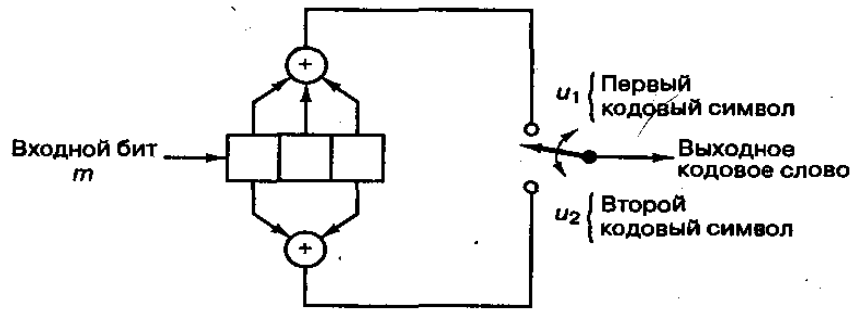


Рис.1.2.5. – Сверточный кодер со степенью кодирования $\frac{1}{2}$ и величиной кодового ограничения $K=3$. [3]

Одним из способов представления простых кодирующих устройств является диаграмма состояний (рис.1.2.6).

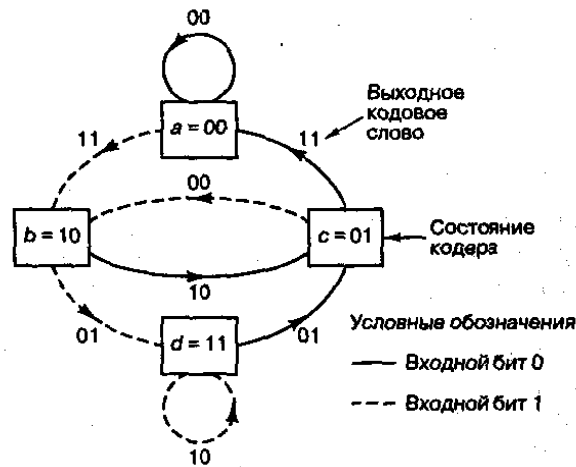


Рис.1.2.6. – Диаграмма состояний кодера со степенью кодирования $\frac{1}{2}$ и $K=3$ [3]

Состояния, показанные в рамках диаграммы, представляют собой возможное содержимое $K-1$ крайне правых разрядов регистра, а пути между состояниями – ответвляющиеся слова на выходе, являющиеся результатом переходов между такими состояниями. Существует всего два исходящих из каждого состояния перехода, и два входящих соответствующие двум возможным входным битам. И главное то, что за один переход невозможно перейти из данного состояния в любое произвольное.

Несмотря на то, что диаграммы состояний полностью описывают кодер, их нельзя использовать для легкого отслеживания переходов кодера в

зависимости от времени. Древоподобная диаграмма (рисунок 1.2.7) прибавляет к диаграмме состояния временное измерение.

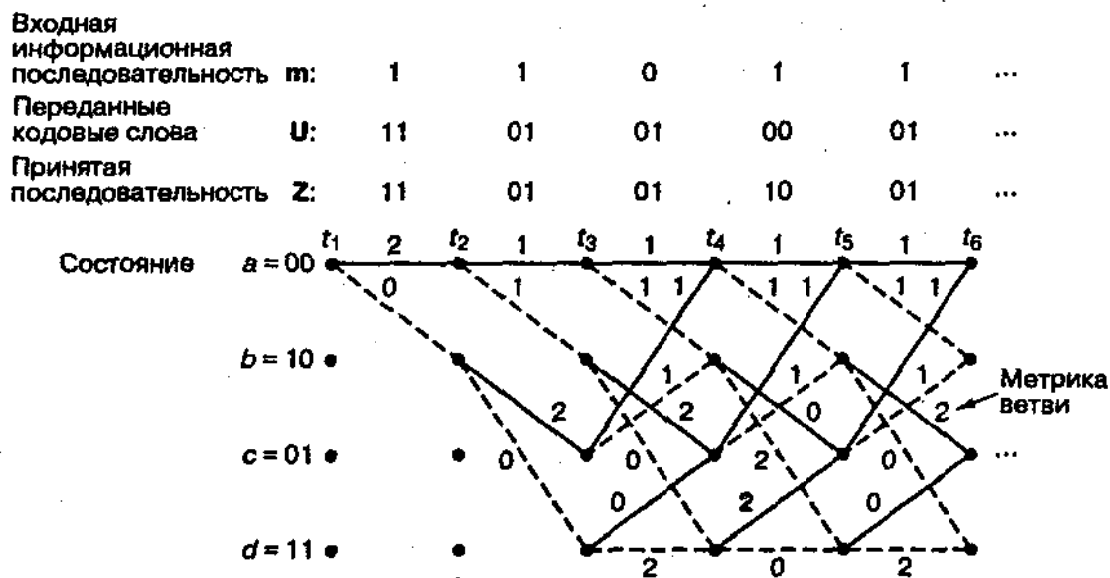


Рис.1.2.7. Решетчатая диаграмма кодера (степень кодирования $\frac{1}{2}$, $K=3$)[3]

В каждый последующий момент прохождения входящего бита процедура кодирования может быть описана с помощью перемещения по диаграмме слева направо, причем каждая ветвь дерева описывает ответвленное слово на выходе. Узлы решетки представляют собой состояния кодера, сплошная линия обозначает выходные данные, генерируемые входным нулевым битом, а пунктирная – выходные данные, генерируемые входным единичным битом. Фиксированная структура древоподобной диаграммы реализуется после достижения глубины K . С этого момента в каждое состояние можно войти из любого из двух предыдущих состояний. Из каждого состояния можно также перейти в одно из двух состояний. Над каждой ветвью написано число, представляющее собой метрику по Хэммингу (количество совпадающих цифр в символе сообщения и выходном символе кодера). Эти пометки на ветвях решетки накапливаются декодером в процессе. Чтобы пометить ветви декодера подходящей метрикой расстояния Хэмминга, рассмотрим диаграмму состояний кодера. Переход между состояниями $00 \rightarrow 00$ порождает на выходе ответвляющееся слово 00 , однако получено 11 . Следовательно, на решетке декодера помечаем переход между состояниями $00 \rightarrow 00$ расстоянием Хэмминга

между ними, а именно 2. Глядя вновь на диаграмму состояний, видим, что переход между состояниями $00 \rightarrow 10$ порождает ответвляющееся слово 11, точно соответствующее полученному в момент времени t_1 кодовому символу. Следовательно, переход на решетке декодера между состояниями $00 \rightarrow 10$ помечаем расстоянием Хэмминга 0. В итоге метрика входящих в решетку декодера ветвей описывает разницу между тем, что было получено и тем, что «могло быть» получено, имея ответственные слова, связанные с теми ветвями, с которых они были переданы.

Выводы по главе 1

1. Многообразие существующих кодов делится на два класса: блочные коды и непрерывные коды. В блочных кодах передаваемая информационная последовательность разбивается на отдельные блоки с добавлением к каждому блоку определенного числа проверочных символов. Кодовые комбинации кодируются и декодируются независимо друг от друга. В непрерывных кодах, называемых также цепными, рекуррентными, конволюционными или сверточными, передаваемая информационная последовательность не разделяется на блоки, а проверочные символы размещаются в определенном порядке между информационными. Процессы кодирования и декодирования также осуществляются в непрерывном режиме.

2. Кодер двоичного свёрточного кода содержит сдвигающий регистр из m разрядов и сумматоры по модулю 2 для образования кодовых символов в выходной последовательности. Входы сумматоров соединены с определёнными разрядами регистра. Коммутатор на выходе устанавливает очередность посылки кодовых символов в канал связи. Тогда структуру кода определяют нижеследующие характеристики.

1. Число информационных символов, поступающих за один такт на вход кодера — k .

2. Число символов на выходе кодера — n , соответствующих k , поступившим на вход символам в течение такта.

3. Скорость кода определяется отношением $R=k/n$ и характеризует избыточность, вводимую при кодировании.

4. Избыточность кода $V = 1-R$

5. Память кода (входная длина кодового ограничения или информационная длина кодового слова), определяется максимальной степенью порождающего многочлена в составе порождающей матрицы

6. Маркировка сверточного кода обозначается в большинстве случаев (n, k, l) , но возможны и вариации.

7. Вес w двоичных кодовых последовательностей определяется числом «единиц», входящих в эту последовательность или кодовые слова.

8. Кодовое расстояние d показывает степень различия между i -й и j -й кодовыми комбинациями при условии их одинаковой длины. Для любых двух двоичных кодовых комбинаций кодовое расстояние равно числу несовпадающих в них символов. В общем виде кодовое расстояние может быть определено как суммарный результат сложения по модулю 2 одноимённых разрядов кодовых комбинаций

$d_{i,j} = \sum_{k=1}^L k_{i,k} \oplus k_{j,k}$, где $k_{i,k}$ и $k_{j,k}$ — k -е символы кодовых комбинаций i и j ; L - длина кодовой комбинации.

9. Минимальное кодовое расстояние d_{\min} — это наименьшее расстояние Хемминга для набора кодовых комбинаций постоянной длины. Для его нахождения необходимо перебрать все возможные пары кодовых комбинаций. Тогда получаем $d_{\min} = \min d_{i,j}$.

2. ОСОБЕННОСТИ ДЕКОДИРОВАНИЯ ИНФОРМАЦИИ ПО АЛГОРИТМУ ВИТЕРБИ

Алгоритм декодирования Витерби был открыт и проанализирован Витерби (Viterbi) в 1967 году. В алгоритме Витерби, по сути, реализуется декодирование, основанное на принципе максимального правдоподобия; однако в нем уменьшается вычислительная нагрузка за счет использования особенностей структуры конкретной решетки кода. Преимущество декодирования Витерби, по сравнению с декодированием по методу "грубой силы", заключается в том, что сложность декодера Витерби не является функцией количества символов в последовательности кодовых слов [10].

Алгоритм включает в себя вычисление меры подобия (или расстояния), между сигналом, полученным в момент времени t_1 , и всеми путями решетки, входящими в каждое состояние в момент времени t_1 . В алгоритме Витерби не рассматриваются те пути решетки, которые, согласно принципу максимального правдоподобия, заведомо не могут быть оптимальными. Если в одно и то же состояние входят два пути, выбирается тот, который имеет лучшую метрику; такой путь называется выживающим. Отбор выживающих путей выполняется для каждого состояния. Таким образом, декодер углубляется в решетку, принимая решения путем исключения менее вероятных путей. Предварительный отказ от маловероятных путей упрощает процесс декодирования. В 1969 году Омуро (Omura) показал, что алгоритм Витерби — это, фактически, максимальное правдоподобие. Отметим, что задачу отбора оптимальных путей можно выразить как выбор кодового слова с максимальной метрикой правдоподобия или минимальной метрикой расстояния.

2.1 . Пример сверточного декодирования Витерби

Для простоты предположим, что мы имеем дело с каналом BSC (Binary symmetric channel – двоичный симметричный канал); в таком случае приемлемой мерой расстояния будет расстояние Хэмминга. Для представления декодера, как показано на рис.2.1.1, можно воспользоваться решетчатым

представлением. Мы начинаем в момент времени t_1 в состоянии 00 (вследствие очистки кодера между сообщениями декодер находится в начальном состоянии). Поскольку в этом примере возможны только два перехода, начинающиеся в некотором состоянии, для начала не нужно показывать все ветви. Полная решетчатая структура образуется после момента времени t_3 . Для решетки декодера каждую ветвь за каждый временной интервал удобно пометить расстоянием Хэмминга между полученным кодовым символом и ответвляющимся словом, соответствующим той же ветви из решетки кодера. На рис. 2.1.1 показана последовательность сообщений m , соответствующая последовательности кодовых слов U , и искажённая шумом последовательность $Z=11\ 01\ 01\ 1001$ [11].

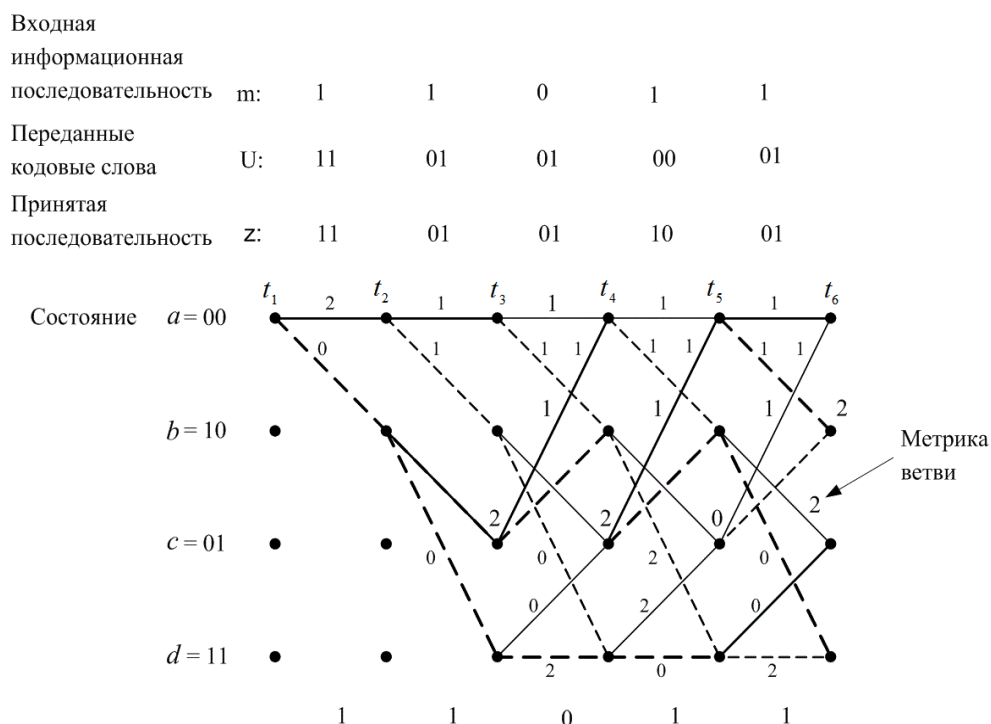


Рис. 2.1.1. – Решетчатая диаграмма декодера (степень кодирования $1/2$, $K=3$)

Кодер характеризуется кодовыми словами, находящимися на ветвях решетки кодера и заведомо известными как кодеру, так и декодеру. Эти ответвляющиеся слова являются кодовыми символами, которые можно было бы ожидать на выходе кодера в результате каждого перехода между состояниями. Пометки на ветвях решетки декодера накапливаются декодером в

процессе. Другими словами, когда получен кодовый символ, каждая ветвь решетки декодера помечена метрикой подоби́я (расстоянием Хэмминга) между полученным кодовым символом и каждым ответвляющимся словом за этот временной интервал. Из полученной последовательности Z , показанной на рис.2.1.1, можно видеть, что кодовые символы, полученные в (следующий) момент времени t_1 , — это 11. Чтобы пометить ветви декодера подходящей метрикой расстояния Хэмминга в (прошедший) момент времени t_1 , рассмотрим решетку кодера, приведенную на рис. 1.2.7. Видим, что переход между состояниями $00 \rightarrow 00$ порождает на выходе ответвляющееся слово 00. Однако получено 11. Следовательно, на решетке декодера помечаем переход между состояниями $00 \rightarrow 00$ расстоянием Хэмминга между ними, а именно 2. Глядя вновь на решетку кодера, видим, что переход между состояниями $00 \rightarrow 10$ порождает на выходе ответвляющееся слово 11, точно соответствующее полученному в момент t_1 кодовому символу. Следовательно, переход на решетке декодера между состояниями $00 \rightarrow 10$ помечаем расстоянием Хэмминга 0. В итоге, метрика входящих в решетку декодера ветвей описывает разницу (расстояние) между тем, что было получено, и тем, что "могло бы быть" получено, имея ответвленные слова, связанные с теми ветвями, с которых они были переданы. По сути, эти метрики описывают величину, подобную корреляциям между полученным ответвляющимся словом и каждым из кандидатов на роль ответвляющегося слова. Таким же образом продолжаем помечать ветви решетки декодера по мере получения символов в каждый момент времени t_i . В алгоритме декодирования эти метрики расстояния Хэмминга используются для нахождения наиболее вероятного (с минимальным расстоянием) пути через решетку.

Смысл декодирования Витерби заключается в следующем. Если любые два пути сливаются в одном состоянии, то при поиске оптимального пути один из них всегда можно исключить. Например, на рис. 2.1.2 показано два пути, сливающихся в момент времени t_3 в состоянии 00.

Давайте определим суммарную метрику пути по Хэммингу для данного пути в момент времени t_i , как сумму метрик расстояний Хэмминга ветвей, по которым проходит путь до момента t_i . На рис.2.1.2 верхний путь имеет метрику 4, нижний — метрику 1. Верхний путь нельзя выделить как оптимальный, поскольку нижний путь, входящий в то же состояние, имеет меньшую метрику. Это наблюдение поддерживается Марковской природой состояний кодера. Настоящее состояние завершает историю кодера в том смысле, что предыдущие состояния не могут повлиять на будущие состояния или будущие ветви на выходе.

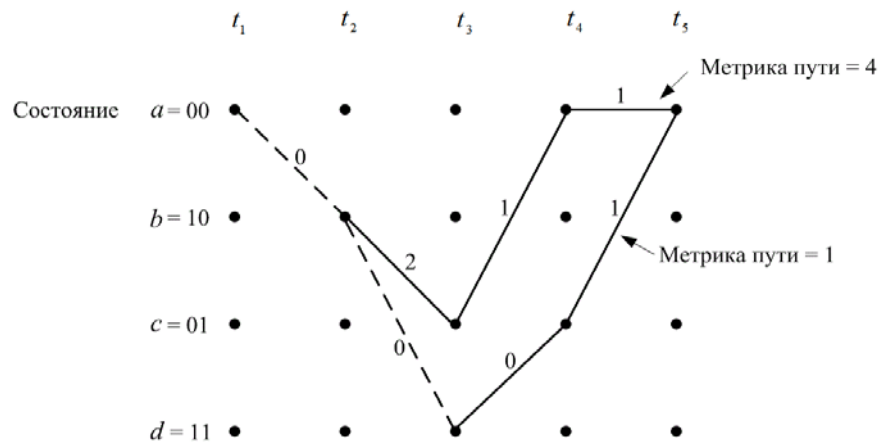


Рис..2.1.2. – Метрики пути для двух сливающихся путей

В каждый момент времени t_i в решетке существует 2^{K-1} состояний, где K — это длина кодового ограничения, и в каждое состояние может войти два пути. Декодирование Витерби состоит в вычислении метрики двух путей, входящих в каждое состояние, и исключении одного из них. Такие вычисления проводятся для каждого из 2^{K-1} состояний или узлов в момент времени t_i ; затем декодер переходит к моменту времени t_{i+1} , и процесс повторяется. В данный момент времени метрика выжившего пути для каждого состояния обозначается как метрика для этого состояния в этот момент времени. Первые несколько шагов в нашем примере декодирования будут следующими (рис.2.1.3). Предположим, что последовательность входящих данных t , кодовое слово U и Полученная последовательность Z аналогичны показанным на рис.2.1.1.

Допустим, что декодер знает верное исходное состояние решетки. (Это предположение не является необходимым, однако упрощает объяснения.) В момент времени t_1 получены кодовые символы 11. Из состояния 00 можно перейти только в состояние 00 или 10, как показано на рис.2.1.3, а. Переход между состояниями $00 \rightarrow 10$ имеет метрику ветви 0; переход между состояниями $00 \rightarrow 00$ — метрику ветви 2. В момент времени t_2 из каждого состояния также может выходить только две ветви, как показано на рис.2.1.3, б. Суммарная метрика этих ветвей обозначена как метрика состояний Γ_a , Γ_b , Γ_c и Γ_d , соответствующих конечным состояниям. В момент времени t_3 на рис.2.1.3, в, опять есть две ветви, выходящие из каждого состояния. В результате имеется два пути, входящих в каждое состояние, в момент времени t_4 . Один из путей, входящих в каждое состояние, может быть исключен, а точнее — это путь, имеющий большую суммарную метрику пути. Если бы метрики двух входящих путей имели одинаковое значение, то путь, который будет исключаться, выбирался бы произвольно. Выживший путь в каждом состоянии показан на рис.2.1.3, г. В этой точке процесса декодирования имеется только один выживший путь, который называется полной ветвью, между моментами времени t_1 и t_4 . Следовательно, декодер теперь может решить, что между моментами t_1 и t_4 произошел переход $00 \rightarrow 10$. Поскольку переход вызывается единичным входным битом, на выходе декодера первым битом будет единица. Здесь легко можно проследить процесс декодирования выживших ветвей, поскольку ветви решетки показаны пунктирными линиями для входных нулей и сплошной линией для входных единиц. Заметим, что первый бит не декодируется, пока вычисление метрики пути не пройдет далее вглубь решетки. Для обычного декодера такая задержка декодирования может оказаться раз в пять больше длины кодового ограничения в битах.

На каждом следующем шаге процесса декодирования всегда будет два пути для каждого состояния; после сравнения метрик путей один из них будет

исключен. Этот шаг в процессе декодирования показан на рис.2.1.3, д. В момент t_5 снова имеется по два входных пути для каждого состояния, и один путь из каждой пары подлежит исключению. Выжившие пути на момент t_5 показаны на рис.2.1.3, е. Заметим, что в нашем примере мы еще не можем принять решения относительно второго входного информационного бита, поскольку еще остается два пути, исходящих в момент t_2 из состояния в узле 10. В момент времени t_6 на рис.2.1.3, ж снова можем видеть структуру сливающихся путей, а на рис.2.1.3, з — выжившие пути на момент t_6 . Здесь же, на рис.2.1.3, з, на выходе декодера в качестве второго декодированного бита показана единица как итог единственного оставшегося пути между точками t_2 и t_3 . Аналогичным образом декодер продолжает углубляться в решетку и принимать решения, касающиеся информационных битов, устраняя все пути, кроме одного.

Отсекание (сходящихся путей) в решетке гарантирует, что у нас никогда не будет путей больше, чем состояний. В этом примере можно проверить, что после каждого отсекания (рис.2.1.3, б—д) остается только 4 пути. Сравните это с попыткой применить "грубую силу" (без привлечения алгоритма Витерби) при использовании для получения последовательности принципа максимального правдоподобия. В этом случае число возможных путей (соответствующее возможным вариантам последовательности) является степенной функцией длины последовательности. Для двоичной последовательности кодовых слов с длиной ответвленных слов L имеется 2^L возможные последовательности.

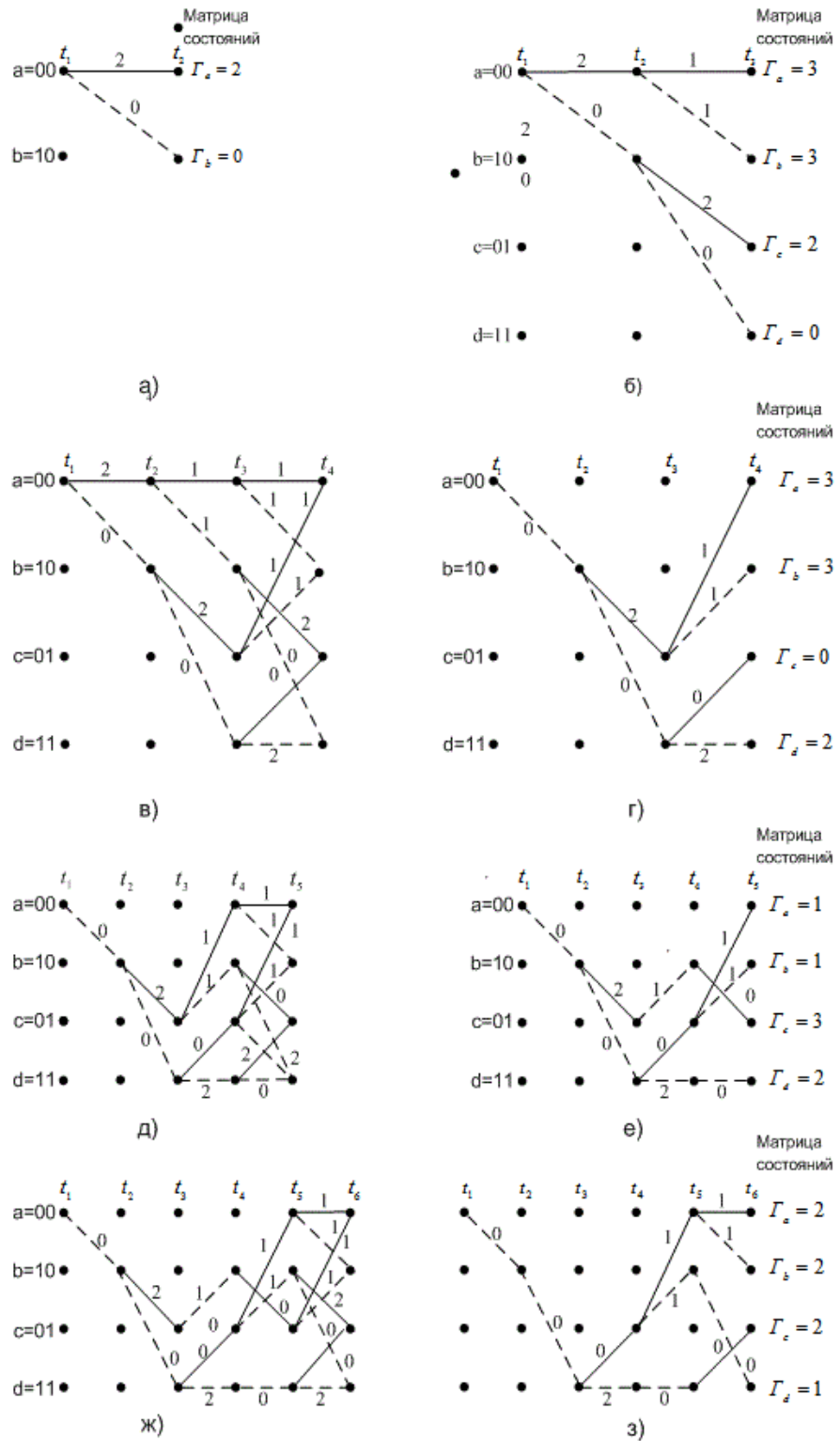


Рис.2.1.3. – Выбор выживших путей: а) выжившие на момент t_2 ; б) выжившие на момент t_3 ; в) сравнение метрик в момент t_4 ; г) выжившие на момент t_4 ; д) сравнение метрик в момент t_5 ; е) выжившие на момент t_5 ; ж) сравнение метрик в момент t_6 ; з) выжившие на момент t_6

2.2. Реализация декодера

В контексте решетчатой диаграммы, показанной на рис.2.1.1, переходы за один промежуток времени можно сгруппировать в 2^{v-1} непересекающиеся ячейки; каждая ячейка будет изображать четыре возможных перехода, причем $v = K - 1$ называется памятью кодера (encoder memory). Если $K = 3$, то $v = 2$, и, следовательно, мы имеем $2^{v-1} = 2$ ячейки. Эти ячейки показаны на рис.2.2.1, где буквы a, b, c и d обозначают состояния в момент t_i , а a' , b' , c' и d' — состояния в момент времени t_{i+1} . Для каждого перехода изображена метрика ветви δ_{xy} , индексы которой означают переход из состояния x в состояние y. Эти ячейки и соответствующие логические элементы, которые корректируют метрики состояний $\{\Gamma_x\}$, где x означает конкретное состояние, представляют основные составляющие элементы декодера.

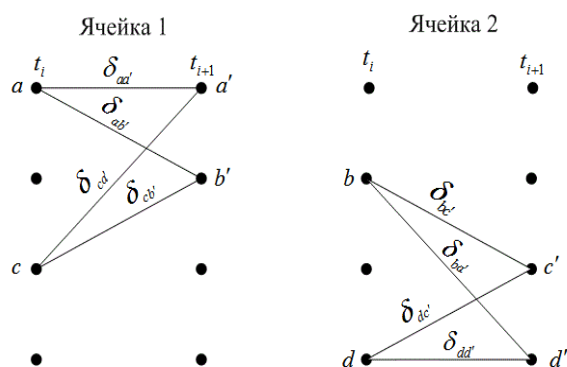


Рис.2.2.1. – Примеры ячеек декодера

2.2.1. Процедура сложения, сравнения и выбора

Вернемся к примеру двух ячеек с $K = 3$. На рис. 2.2.2 показан логический блок, соответствующий ячейке 1. Логическая схема осуществляет специальную операцию, которая называется сложение, сравнение и выбор (add-compare-select — ACS). Метрика состояния $\Gamma_{a'}$ вычисляется путем прибавления метрики предыдущего состояния a, Γ_a , к метрике ветви $\delta_{aa'}$ и метрики предыдущего состояния c, Γ_c , к метрике ветви $\delta_{ca'}$. Это даст в результате две метрики путей в

качестве кандидатов для новой метрики состояния $\Gamma_{a'}$. Оба кандидата сравниваются в логическом блоке, показанном на рис. 2.2.2. Наиболее правдоподобная из двух метрик путей (с наименьшим расстоянием) запоминается как новая метрика состояния $\Gamma_{a'}$ для состояния a' . Также сохраняется новая история путей $\hat{m}_{a'}$ для состояния a' , где $\hat{m}_{a'}$ — история пути информации для данного состояния, дополненная сведениями о выжившем пути.

На рис. 2.2.2 также показана логическая схема ACS для ячейки 1, которая дает новую метрику состояния $\Gamma_{b'}$ и новую историю состояния $\hat{m}_{b'}$. Операция ACS аналогичным образом осуществляется и для путей в других ячейках. Выход декодера составляют последние биты на путях с наименьшими метриками состояний.

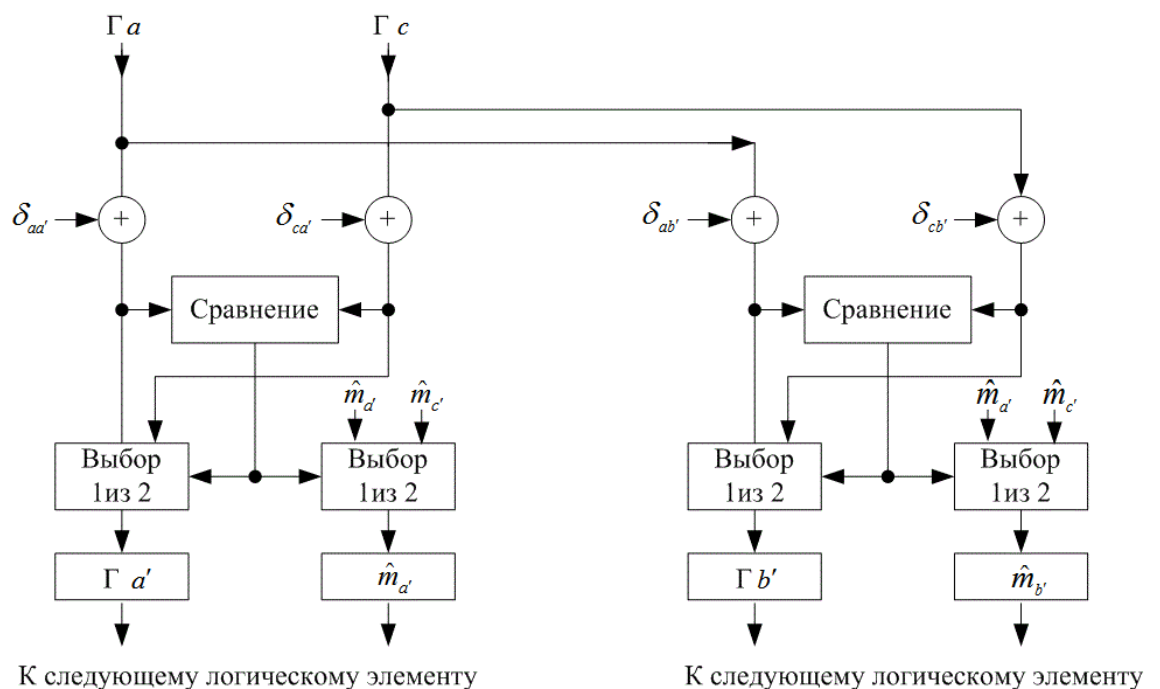


Рис.2.2.2. – Логический блок, предназначенный для осуществления операции сложения, сравнения и выбора

2.2.2. Вид процедуры сложения, сравнения и выбора на решетке

Рассмотрим тот же пример, которым мы воспользовались в разделе 2.2.1 для описания декодирования на основе алгоритма Витерби. Последовательность сообщений имела вид $m = 11011$, последовательность

кодовых слов — $U=11\ 01\ 01\ 00\ 01$, а принятая последовательность — $Z=11\ 01\ 01\ 10\ 01$.

Решетчатая диаграмма декодирования, аналогичная показанной на рис. 2.1.1, изображена на рис.2.2.3.

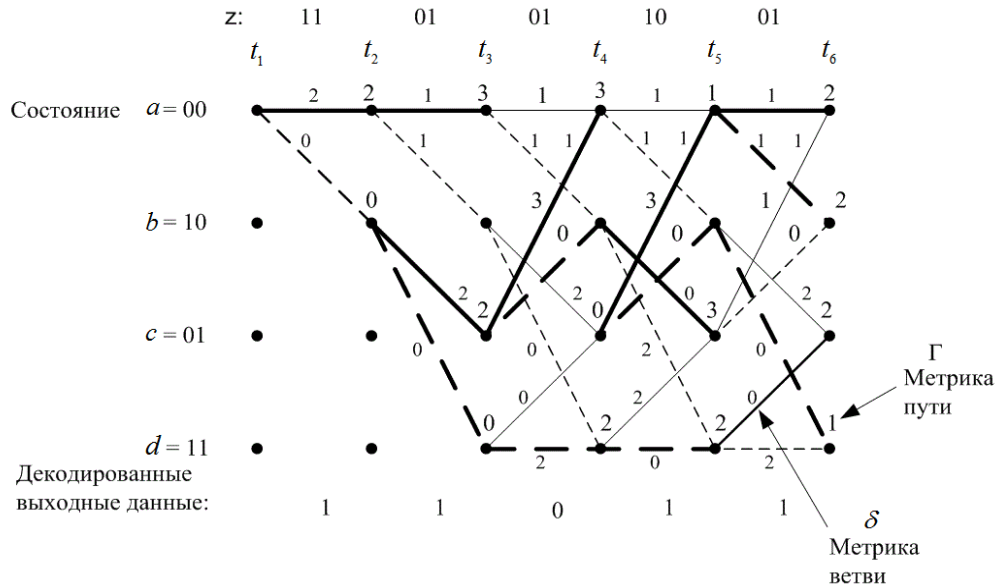


Рис.2.2.3. – Операция сложения, сравнения и выбора при декодировании по алгоритму Витерби

Метрика ветви, которая описывает каждую ветвь, — это расстояние Хэмминга между принятым кодовым символом и соответствующим ответвленным словом из решетки кодера. Еще на решетке (рис.2.2.3) показаны значения каждого состояния x в каждый момент $t_2 - t_6$, метрика состояния которых обозначена Γ_x . Операция ACS выполняется после появления двух переходов, входящих в состояние, т.е. для момента t_4 и более поздних. Например, в момент времени t_4 значение метрики состояния для состояния a вычисляется суммированием метрики состояния $\Gamma_a = 3$ в момент t_3 и метрики ветви $\delta_a = 1$, что в итоге дает значение 4. В то же время к метрике состояния $\Gamma_c = 2$ в момент времени t_3 прибавляется метрика ветви $\delta_c = 1$, что дает значение 3. В ходе процедуры ACS происходит отбор наиболее правдоподобной метрики (с минимальным расстоянием), т.е. новой метрики состояния; поэтому для состояния a в момент t_4 новой метрикой состояния

будет $T_{\text{дл}}=3$. Отобранный путь изображен жирной линией, а путь, который был отброшен, показан светлой линией. На рис. 7.15 на решетке слева направо показаны все метрики состояний. Убедимся, что в любой момент времени значение каждой метрики состояния получается суммированием метрики состояния, соединенного с предыдущим состоянием вдоль отобранного пути (жирная линия), и метрики ветви, соединяющей эти состояния. В определенной точке решетки (после временного интервала, равного 4 или 5 длинам кодового ограничения) будут декодированы самые ранние биты. Чтобы показать это, посмотрим на рис. 2.2.3 в момент t_5 . Видим, что значение метрики состояния, соответствующей минимальному расстоянию, равно 1. Отобранный путь можно проследить из состояния d обратно, к моменту t_1 , и убедиться, что декодированное сообщение совпадает с исходным. Напомним, что пунктирные и сплошные линии соответствуют двоичным единице и нулю.

2.2.3. Память путей и синхронизация

Требования к памяти декодера, работающего согласно алгоритму Витерби, растут с увеличением длины кодового ограничения как степенная функция. Для кода со степенью кодирования $1/n$ после каждого шага декодирования декодер держит в памяти набор из 2^{n-1} путей.

С высокой степенью вероятности при превышении существующей глубины декодирования эти пути не будут взаимно непересекающимися [12]. Все 2^{n-1} пути ведут к полной ветви, которая в конце концов разветвляется на разные состояния. Поэтому, если декодер сохраняет историю 2^{n-1} путей, самые первые биты на всех путях будут одинаковы. Следовательно, простой декодер имеет фиксированный объем истории путей и выдает самые ранние биты произвольного пути каждый раз, когда продвигается на один уровень вглубь решетки. Требуемый объем сохраняемых путей будет равен следующему [12].


$$n = h2^{k-1} \quad (2.2.1)$$

Здесь h - длина истории пути информационного бита на состояние. При уточнении, которое проводится для минимизации h , вместо самых ранних

битов произвольных путей на выходе декодера используются самые ранние биты наиболее вероятных путей. Было показано [12], что значения h , равного 4 или 5 длинам кодового ограничения, достаточно, чтобы характеристики декодера были близки к оптимальным. Необходимый объем памяти и является основным ограничением при разработке декодеров, работающих согласно алгоритму Витерби. В серийно выпускаемых декодерах длина кодового ограничения равна величине порядка $K = 10$. Попытка повысить эффективность кодирования за счет увеличения длины кодового ограничения вызывает степенной рост требований к памяти (и сложности), как это следует из уравнения (2.2.1).

Синхронизация ответвляющихся слов — это процесс определения начала ответвляющегося слова в принятой последовательности. Такую синхронизацию можно осуществить, не прибавляя новую информацию к потоку передаваемых символов, поскольку можно видеть, что, пока принятые данные не синхронизированы, у них непомерно высокая частота появления ошибок. Следовательно, синхронизацию можно осуществить просто: нужно проводить сопутствующее наблюдение за уровнем частоты появления ошибок, т.е. нас должна интересовать частота, при которой увеличиваются метрики состояний, или частота, при которой сливаются выжившие пути на решетке. Параметр, за которым следят, сравнивается с пороговым значением, после чего соответствующим образом осуществляется синхронизация.

2.3. Модели каналов: мягкое или жесткое принятие решений

Последовательность кодовых слов , определяемую ответвленными словами, каждое из которых состоит из n кодовых символов, можно рассматривать как бесконечный поток, в отличие от блочного кода, где исходные данные и их кодовые слова делятся на блоки строго определенного размера. Последовательность кодовых слов выдается сверточным кодером и подается на модулятор, где кодовые символы преобразуются в сигналы. Модуляция может быть узкополосной (например, модуляция импульсными сигналами) или полосовой (например, модуляция PSK или FSK). Вообще, за

такт в сигнал $s_i(t)$ преобразуется l символов, где l — целое, причем $i = 1, 2, \dots$, а $M = 2^l$. Если $l = 1$, модулятор преобразует каждый кодовый символ в двоичный сигнал. Предполагается, что канал, по которому передается сигнал, искажает сигнал гауссовым шумом. После того как искаженный сигнал принят, он сначала обрабатывается демодулятором, а затем подается на декодер.

Рассмотрим ситуацию, когда двоичный сигнал передается за отрезок времени $(0, T)$, причем двоичная единица представляется сигналом $s_1(t)$, а двоичный нуль — сигналом $s_2(t)$. Принятый сигнал имеет вид $r(t) = s_i(t) + n(t)$, где $n(t)$ представляет собой вклад гауссового шума с нулевым средним. В главе 3 мы описывали обнаружение $r(t)$ в два основных этапа. На первом этапе принятый сигнал переводится в число $z(t) = a_i + n_0$, где a_i — это компонент сигнала $z(T)$, а n_0 — компонент шума. Компонент шума n_0 — это случайная переменная, значения которой имеют гауссово распределение с нулевым средним. Следовательно, $z(T)$ также будет случайной гауссовой величиной со средним a_1 или a_2 , в зависимости от того, какая величина была отправлена — двоичная единица или двоичный нуль. На втором этапе процесса обнаружения принимается решение о том, какой сигнал был передан. Это решение принимается на основе сравнения $z(T)$ с порогом. Условные вероятности $p(z|s_1)$ и $p(z|s_2)$, показанные на рис.2.3.1, обозначены как правдоподобие s_1 и s_2 . Демодулятор преобразует упорядоченный по времени набор случайных переменных $\{z(t)\}$ в кодовую последовательность Z и подает ее на декодер. Выход демодулятора можно настроить по-разному. Можно реализовать его в виде жесткой схемы принятия решений относительно того, представляет ли $z(T)$ единицу или нуль. В этом случае выход демодулятора квантуется на два уровня, нулевой и единичный, и соединяется с декодером. Поскольку декодер работает в режиме жесткой схемы принятия решений, принятых демодулятором, такое декодирование называется жестким.

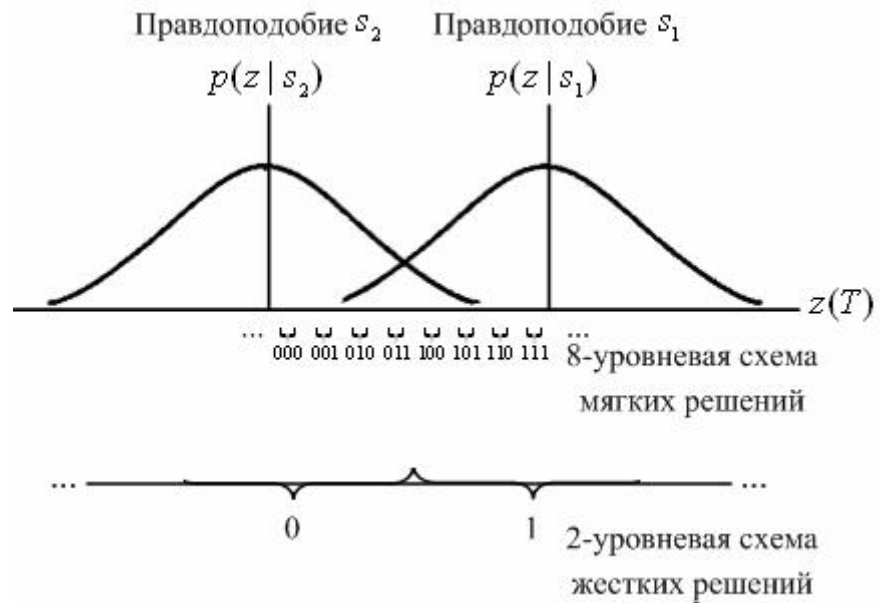


Рис.2.3.1. – Жесткая и мягкая схемы декодирования

Аналогично демодулятор можно настроить так, чтобы он подавал на декодер значение $z(T)$, квантованное более чем на два уровня. Такая схема обеспечивает декодер большим количеством информации, чем жесткая схема решений. Если выход демодулятора имеет более двух уровней квантования, то декодирование называется мягким. На рис. 2.3.1 на оси абсцисс изображено восемь (3-битовых) уровней квантования. Если в демодуляторе реализована жесткая схема принятия двоичных решений, он отправляет на декодер только один двоичный символ. Если в демодуляторе реализована мягкая двоичная схема принятия решений, квантованная на восемь уровней, он отправляет на декодер 3-битовое слово, описывающее интервал, соответствующий $z(T)$. По сути, поступление такого 3-битового слова, вместо одного двоичного символа эквивалентно передаче декодеру меры достоверности вместе с решением относительно кодового символа. Согласно рис.2.3.1, если с демодулятора поступила на декодер последовательность 111, это равносильно утверждению, что с очень высокой степенью достоверности кодовым символом была 1, в то время как переданная последовательность 100 равносильна утверждению, что с очень низкой степенью достоверности кодовым символом была 1. Совершенно ясно, что в конечном счете каждое решение, принятое декодером и касающееся сообщения, должно быть жестким; в противном случае на распечатках

компьютера можно было бы увидеть нечто, подобное следующему: "думаю, это 1", "думаю, это 0" и т.д. То, что после демодулятора не принимается жесткое решение и на декодер поступает больше данных (мягкое принятие решений), можно понимать как промежуточный этап, необходимый для того, чтобы на декодер поступило больше информации, с помощью которой он затем сможет восстановить последовательность сообщения (с более высокой достоверностью передачи сообщения по сравнению с декодированием в рамках жесткой схемы принятия решений). Показанная на рис. 7.8, 8-уровневая метрика мягкой схемы принятия решений часто обозначается как -7, -5, -3, -1, 1, 3, 5, 7. Такие обозначения вводятся для простоты интерпретации мягкой схемы принятия решения. Знак метрики характеризует решение (например, выбирается \hat{s}_1 , если величина положительна, и \hat{s}_2 , если отрицательна), а величина метрики описывает степень достоверности этого решения. Преимуществом метрики, показанной на рис.2.3.1, является только то, что в ней не используются отрицательные числа.

Для гауссова канала восьмиуровневое квантование, по сравнению с двухуровневым, приводит в результате к улучшению на 2 дБ требуемого отношения сигнал/шум. Это означает, что восьмиуровневое квантование с мягкой схемой принятия решений может дать ту же вероятность появления ошибочного бита, что и декодирование с жесткой схемой принятия решений, однако требует на 2 дБ меньшего значения E_b/N_0 при прочих равных характеристиках. Аналоговое квантование (или квантование с бесконечным числом уровней) дает в результате улучшение на 2,2 дБ, по сравнению с двухуровневым; следовательно, при восьмиуровневом квантовании, по сравнению с квантованием с бесконечным числом уровней, теряется приблизительно 0,2 дБ. По этой причине квантование более чем на восемь уровней может дать только небольшое улучшение производительности [12]. Какова цена, которую следует заплатить за такое улучшение параметров декодирования с мягкой схемой принятия решений? В случае декодирования с

жесткой схемой принятия решений, для описания каждого кодового символа используется один бит, в то время как при восьмиуровневой мягкой схеме принятия решения для описания каждого символа применяется 3 бит; следовательно, в течение процесса декодирования нужно успеть обработать в три раза больше данных. Поэтому за мягкое декодирование приходится платить увеличением требуемых объемов памяти (и, возможно, возникнут проблемы со скоростью обработки).

В настоящее время существуют блочные и сверточные алгоритмы декодирования, функционирующие на основе жесткой или мягкой схемы принятия решений. Однако при блочном декодировании мягкая схема принятия решений, как правило, не используется, поскольку ее значительно сложнее реализовать, чем схему жесткого принятия решений. Чаще всего мягкая схема принятия решений применяется в алгоритме сверточного декодирования Витерби, поскольку при декодировании Витерби мягкое принятие решений лишь незначительно усложняет вычисления.

Выводы по главе 2.

1. В настоящее время существующие алгоритмы декодирования функционируют на основе жесткой или мягкой схемы принятия решений. Однако при блочном декодировании мягкая схема принятия решений, как правило, не используется, поскольку ее значительно сложнее реализовать, чем схему жесткого принятия решений. Чаще всего мягкая схема принятия решений применяется в алгоритме сверточного декодирования Витерби, поскольку при декодировании Витерби мягкое принятие решений лишь незначительно усложняет вычисления.

2. Выход демодулятора можно настроить по-разному. Можно реализовать его в виде жесткой схемы принятия решений относительно того, представляет ли решение относительно правильности принятого сигнала единицу или нуль. В этом случае выход демодулятора квантуется на два уровня, нулевой и единичный, и соединяется с декодером. Поскольку декодер

работает в режиме жесткой схемы принятия решений, принятых демодулятором, такое декодирование называется жестким.

3. Аналогично демодулятор можно настроить так, чтобы он подавал на декодер значение решения, квантованное более чем на два уровня. Такая схема обеспечивает декодер большим количеством информации, чем жесткая схема решений. Если выход демодулятора имеет более двух уровней квантования, то декодирование называется мягким.

4. В 1967 году Эндрю Витерби разработал и проанализировал алгоритм декодирования, основанный на принципе максимального правдоподобия. Алгоритм оптимизирован за счёт использования особенностей структуры конкретной решётки кода. Преимущество декодирования Витерби по сравнению с декодированием по методу полного перебора заключается в том, что сложность декодера Витерби не является функцией количества символов в последовательности кодовых слов.

5. Алгоритм включает в себя вычисление меры подобия (или расстояния), между сигналом, полученным в момент времени t_i , и всеми путями решётки, входящими в каждое состояние в момент времени t_i . В алгоритме Витерби не рассматриваются те пути решётки, которые, согласно принципу максимального правдоподобия, заведомо не могут быть оптимальными. Если в одно и то же состояние входят два пути, выбирается тот, который имеет лучшую метрику; такой путь называется выживающим. Отбор выживающих путей выполняется для каждого состояния. Таким образом, декодер углубляется в решётку, принимая решения путём исключения менее вероятных путей. Предварительный отказ от маловероятных путей упрощает процесс декодирования. В 1969 году Джим Омюра (англ.) показал, что основу алгоритма Витерби составляет оценка максимального правдоподобия. Отметим, что задачу отбора оптимальных путей можно выразить как выбор кодового слова с максимальной метрикой правдоподобия или минимальной метрикой расстояния.

3. РАЗРАБОТКА ПРОГРАММНОЙ МОДЕЛИ ДЛЯ ДЕКОДИРОВАНИЯ ТЕКСТОВОГО СООБЩЕНИЯ С ИСПОЛЬЗОВАНИЕМ МЯГКОГО И ЖЕСТКОГО АЛГОРИТМА ВИТЕРБИ

3.1. Программная модель для декодирования текстового сообщения

Программная модель для декодирования текстового сообщения с использованием мягкого и жесткого алгоритма Витерби, листинг которой приведен в приложении А, наглядно демонстрирует разницу в приеме информации, передаваемой по каналу с аддитивным белым гауссовским шумом в незакодированном и закодированном виде, и демонстрирует результат декодирования текстового сообщения с использованием жесткого и мягкого алгоритма Витерби. Широкое распространение алгоритма Витерби заключается в том, что сложность декодера Витерби не является функцией количества символов в последовательности кодовых слов [13].

В программной реализации используется многофазная манипуляция. Данный вид модуляции был выбран потому, что он может быть использован в каналах связи, имеющих достаточно ограниченный частотный ресурс, но к которым предъявляются высокие требования по качеству передачи информации [14]. Согласно [15], к основным показателям, характеризующим тот или иной модуляционный формат, следует отнести помехоустойчивость и спектральную эффективность, под которой понимают полосу частот, необходимую для передачи сигналов с требуемой скоростью и достоверностью.

В общем случае, данные показатели тесно взаимосвязаны между собой. Это приводит к тому, что улучшение одного из них невозможно без снижения качества другого. Между тем, переход на цифровые системы передачи информации во многом predetermined типы используемых модуляционных форматов, к которым, в первую очередь, следует отнести различные виды манипуляций, как наиболее полно соответствующих дискретной природе передаваемой битовой информации.

Однако, возможности широко известных форматов манипуляции фактически исчерпаны, а их дальнейшая модернизация не позволяет получить

существенного выигрыша. В связи с этим, в последнее время наметилась тенденция на переход к формированию сигналов в базисах, отличных от гармонических [16,17]. В частности, вейвлет-базисы, базисы сплайн-характеров и др. Однако такой подход предполагает полную замену приемо-передающего оборудования, что сложно осуществить в существующих реалиях. Между тем, по мнению зарубежных специалистов [18,19], традиционные виды модуляции еще до конца не исчерпали свои возможности. Повышению скорости передачи информации в канале способствует увеличение позиционности модуляционного формата, однако при этом возрастает битовая ошибка. В настоящее время компромиссным решением для систем передачи информации по радиоканалу является использование модуляционных форматов на основе ФМн-2 и ФМн-4 [18]. В разработанной программе число позиций модуляции можно задавать.

Модели систем, представленные в пакете Матлаб Simulink для реализации жесткого и мягкого декодирования по алгоритму Витерби [20] разработаны для исходных данных, представляющих собой поток битов.

Модель декодера Витерби для детектирования двоичного фазоманипулированного сигнала, представленная в [21], также разработана для исходных данных, представляющих собой поток битов, и реализует только алгоритм жесткого декодирования Витерби.

По сравнению с рассмотренными моделями систем, реализующими жесткое и мягкое декодирование по алгоритму Витерби [20,21], разработанная программа в пакете Матлаб предназначена для исходных данных, представленных в виде текстового сообщения, и позволяет ознакомиться с программным кодом, реализующим все преобразования сообщения от текстового вида до сигнала в фазовой или квадратурной фазовой манипуляции. Можно также получить представление о формировании алгоритма кода Грея, когда двоичные символы отличаются только одной битовой позицией. В этом случае при появлении ошибки в M -арном символе высока вероятность того, что ошибочным является только один из k прибывших битов [22]. Несмотря на

то, что в версиях Матлаб, появившихся после 2006 года, предусмотрена реализация алгоритма кода Грея в виде одной команды, предлагаемая программа позволяет только посмотреть график получившегося сигнального созвездия, но и реализовать алгоритм кода Грея «пошагово», тем самым закрепив их знания об алгоритме его формирования.

Для реализации декодирования Витерби в программной модели используется тот факт, что данный алгоритм используется в двоичном входном канале с жестким или мягким 3-х битовым квантованным выходом. Длина кодового ограничения варьируется от 3 до 9, причем степень кодирования кода редко оказывается меньше $1/3$, и память путей составляет несколько длин кодового ограничения. В алгоритме декодирования Витерби все пути по решетке кода пошагово сравниваются с принятой из канала связью кодовой последовательностью и отбрасываются те из них, которые точно будут находиться на большем расстоянии, чем другие. Таким образом, главным недостатком данного алгоритма становится необходимость анализировать все пути, выходящие из рассматриваемого узла кодовой решетки. Этот недостаток легко преодолевается благодаря большим вычислительным возможностям современных компьютеров. А благодаря тому, что сложность декодера Витерби не является функцией количества символов в последовательности кодовых слов, он находит широкое применение в современных цифровых системах связи. Алгоритм декодирования Витерби также хорошо использовать в тех случаях, когда шум в канале связи стационарен и легко моделируется.

В жесткой схеме декодирования по алгоритму Витерби используется расстояние Хэмминга, определяемое как количество элементов, которыми различаются два кодовых слова.

3.2. Разработка алгоритма программы для разработки модели декодирования

Основное различие между мягким и жестким декодированием по алгоритму Витерби состоит в том, что в мягкой схеме не используется метрика расстояния Хэмминга, поскольку она имеет ограниченное разрешение.

Метрикой расстояний, которая имеет нужное разрешение, в данном случае выступает евклидово кодовое расстояние, поэтому, чтобы облегчить ее применение в программной модели соответствующим образом преобразуются двоичные числа из единиц и нулей в восьмеричные числа от 0 до 7. Выигрыш в значении битового отношения сигнал/шум при использовании «мягкого» решения составляет $E_b/N_0 \approx 2$ дБ (рис.3.1)[23].

Алгоритм программы состоит из следующей последовательности действий:

1. Ввод текстовой строки сообщения.
2. Преобразование текста в ASCII –код.
3. Преобразование символов ASCII –кода в бинарные символы.
4. Формирование из бинарных символов последовательности бит для подачи на вход кодера.
5. Кодирование бинарной последовательности кодом Грея
6. Кодирование бинарной последовательности сверточным кодом для защиты от ошибок
7. Применение фазовой манипуляции к закодированной и незакодированной последовательности битов. Возможность выбора параметров манипуляции.
8. Моделирование прохождения закодированной и незакодированной последовательности битов через канал с гауссовым шумом. Возможность выбора отношения сигнал/шум в канале связи.
9. Демодуляция закодированной и некодированной последовательности бит
10. Декодирование закодированной последовательности бит с использованием жесткого и мягкого алгоритма Витерби.
11. Расчет квадрата евклидова расстояния между исходным текстовым сообщением и принятыми сообщениями: не подвергавшимся кодированию; декодированному по жесткому алгоритму Витерби и декодированному по

мягкому алгоритму Витерби. Определение асимптотической эффективности кодирования.

Результат работы программной модели при значении битового отношения сигнал/шум $E_b/N_0=4,5$ дБ представлен на рисунке 3.2. Как можно видеть, сообщение, декодированное с помощью «мягкого» алгоритма полностью совпадает с исходным, тогда как в принятом некодированном сообщении и в сообщении, декодированном с помощью «жесткой» схемы, есть ошибки.

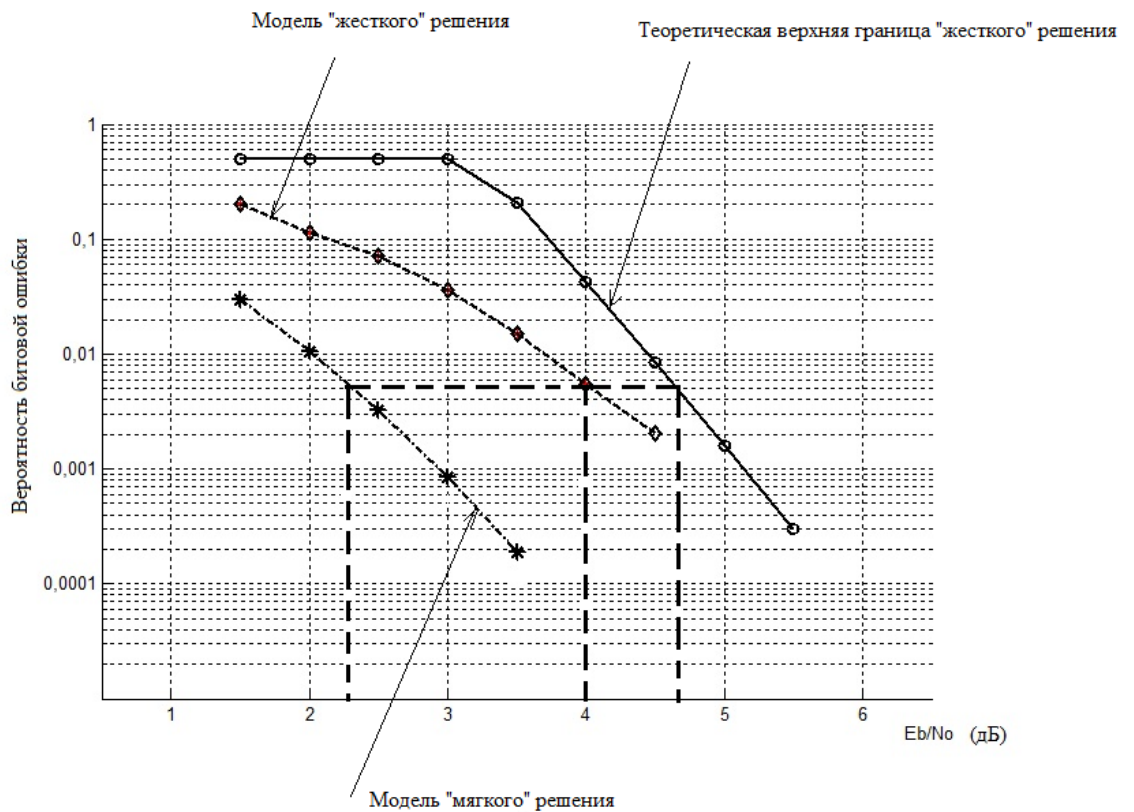


Рисунок 3.1. – Графики зависимости вероятности битовой ошибки от битового отношения сигнал/шум

Асимптотическая эффективность кодирования в программе определяется по формуле:

$$G(\text{дБ}) = 20 \cdot \lg d_f / d_{\text{эт}} \quad (3.1)$$

где d_f и $d_{\text{эт}}$ – евклидов просвет кодированной системы и некодированной эталонной системы.

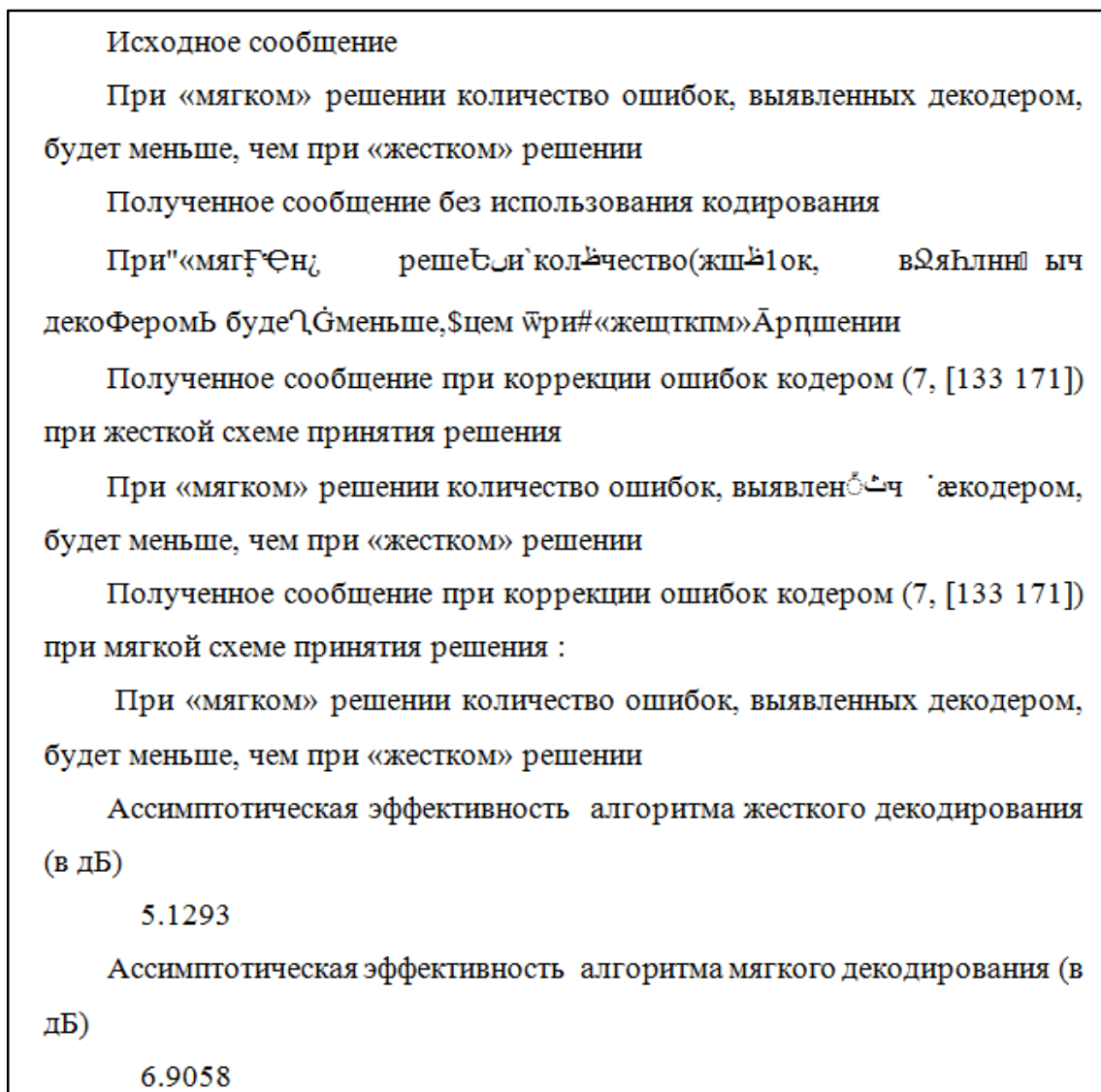


Рисунок 3.2. – Результат работы программной модели декодирования текстового сообщения при значении битового отношения сигнал/шум $E_b/N_0=4,5$ дБ с использованием жесткого и мягкого алгоритма Витерби.

В программе метрика d_{err} представляет собой количество битовых ошибок между принятым некодированным исходным сообщением, а метрика d_f представляет собой количество битовых ошибок между принятым кодированным исходным сообщением, которые вычисляются помощью команды Матлаб `biterr`.

Выводы по главе 3.

1. Программная модель для декодирования текстового сообщения с использованием мягкого и жесткого алгоритма Витерби, листинг которой приведен в приложении А, наглядно демонстрирует разницу в приеме

информации, передаваемой по каналу с аддитивным белым гауссовским шумом в незакодированном и закодированном виде, и демонстрирует результат декодирования текстового сообщения с использованием жесткого и мягкого алгоритма Витерби.

2. В программной реализации используется многофазная манипуляция. Данный вид модуляции был выбран потому, что он может быть использован в каналах связи, имеющих достаточно ограниченный частотный ресурс, но к которым предъявляются высокие требования по качеству передачи информации

3. Асимптотическая эффективность кодирования в программе определяется по формуле: $G(\text{дБ}) = 20 \cdot \lg d_f / d_{\text{эт}}$

где d_f и $d_{\text{эт}}$ – евклидов просвет кодированной системы и некодированной эталонной системы.

Заключение

Подавляющее число современных систем связи работает при передаче самого широкого спектра сообщений (от телеграфа до телевидения) в цифровом виде. Из-за наличия помех в каналах связи сбой при приеме любого элемента вызывает искажение цифровых данных, что может привести, особенно в космических системах связи, к катастрофическим последствиям. В настоящее время по каналам связи передаются цифровые данные со столь высокими требованиями к достоверности передаваемой информации, что удовлетворить эти требования традиционным совершенствованием антенно-фидерных трактов радиолиний, увеличением излучаемой мощности, снижением собственного шума приемника оказывается экономически невыгодным или просто невозможным. Высокоэффективным средством борьбы с помехами в цифровых системах связи является применение помехоустойчивого кодирования, основанного на введении искусственной избыточности в передаваемое сообщение, что приводит к расширению используемой полосы частот и уменьшению информационной скорости передачи.

В данной работе была произведена разработка программной модели для декодирования текстового сообщения с использованием мягкого и жесткого алгоритма Витерби. По сравнению с ранее предложенными программными моделями систем, реализующими жесткое и мягкое декодирование по алгоритму Витерби, разработанная программа в пакете Матлаб предназначена для исходных данных, представленных в виде текстового сообщения, и позволяет ознакомиться с программным кодом, реализующим все преобразования сообщения от текстового вида до сигнала в фазовой или квадратурной фазовой манипуляции. Можно также получить представление о формировании алгоритма кода Грея, когда двоичные символы отличаются только одной битовой позицией.

Список литературы

1. Синицын Д.В . Повышение помехоустойчивости радиотехнических систем передачи информации с использованием сверточных алгоритмов обработки сигналов. Диссертация на соискание ученой степени кандидата технических наук.- Владимир, Владимирский Государственный университет, 2014, - 127 с.
2. Акмалходжаев А.И. Разработка и исследование эффективных алгоритмов декодирования турбокодов в системах мобильной связи. Диссертация на соискание ученой степени кандидата технических наук.-СПб.: СПбГУАП, 2015, - 126 с.
3. Бернард Скляр Цифровая связь. Теоретические основы и практическое применение.- / Скляр Б. ; Гроза Е.Г. [и др.] (пер. с англ.). — 2-е изд. — М. [и др.] : Вильямс, 2016. — 1099 с.
4. Чернецова Е.А. Системы и сети передачи информации. Монография. - СПб.: изд. РГГМУ, 2014, часть 2 «Цифровые технологии передачи данных».- 163 с.
5. Основы построения телекоммуникационных систем и сетей: Учебник для ВУЗов / В.В. Крухмалев, В.Н. Гордиенко, А.Д. Моченов и др. Под ред. В.Н. Гордиенко и В.В. Крухмалева.-М.: Горячая линия-Телеком, 2004.-510 с.
6. Телекоммуникационные системы и сети/Под ред. В.П. Шувалова.В 3-х т. М.: Горячая линия-Телеком ,2003
7. Сердюков П.Н., Синельников А.М., Шевцов И.Ф. Перспективы использования цифровых систем передачи изображения по радиоканалу. — [Электронный ресурс] — Режим доступа: <http://st.ess.ru/publications/articles/serdukov/serdukov.pdf>
8. Телекоммуникационные технологии: введение в технологии GSM: [учебное пособие для вузов / С. Б. Макаров и др.].—2-е изд., испр.—М.: Академия, 2008.—256 с.
9. Столингс В. Компьютерные системы передачи данных. М.: «Вильямс», 2002, - 928 с.

10. Никитин Г. И. Сверточные коды: Учеб. пособие/ СПбГУАП. СПб., 2001, - 80 с.
11. Алгоритм декодирования Витерби – [Электронный ресурс]- Режим доступа: <https://siblec.ru/telekommunikatsii/teoreticheskie-osnovy-tsifrovoj-svyazi/7-kanalnoe-kodirovanie-chast-2#7.3.2>
12. Помехоустойчивые коды : учеб. пособие / В. Г. Журавлев, Н. Ю. Куранова, Ю. Ю. Евсеева ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2013. – 96 с.
13. Помехоустойчивое кодирование. Методы и алгоритмы: Справочник / под.ред. Ю. Б. Зубарева. – М.: Горячая линия–Телеком, 2004. – 126 с.
14. Дворников С.В. Спектрально эффективные сигналы с непрерывной фазой/ С.В.Дворников, С.С.Дворников, С.С.Манаенко и др.// Вестник воронежского государственного технического университета.– 2016.–Т. 12.– № 2.– С. 87–93
15. Федосеев В.Е. Методика и результаты анализа потенциальной помехоустойчивости приема цифрового сигнала на фоне манипулированной структурной помехи/ В.Е.Федосеев, М.С. Иванов// Вестник Воронежского государственного технического университета. – 2010.– Т. 6.– № 11.–С. 108–111
16. Дворников С.В. Помехоустойчивость фазоманипулированных сигналов на основе вейвлетов Гаусса/ С.В.Дворников,С.С.Манаенко// Вестник Воронежского государственного технического университета. –2015.–Т. 11. – № 3. –С. 123–125.
17. Дворников С.В. Синтез манипулированных сигналов на основе вейвлет–функций/ С.В.Дворников, С.С.Дворников, А.М.Спирин // Информационные технологии. – 2013. – № 12. –С. 52–55
18. L. Hong and K. C. Ho. Classification of BPSK and QPSK signals with unknown signal level using the Bayes technique. in Proc. IEEE ISCAS, 2003, pp. IV.1–IV.4.
19. P. Panagiotou, A. Anastasopoulos, and A. Polydoros. Likelihood ratio tests for modulation classification. in Proc. IEEE MILCOM, 2000, pp. 670 –674.

20. ViterbiDecoder [Электронный ресурс] – Режим доступа: www.mathworks.com/help/comm/ref/viterbidecoder.html(дата обращения: 23.02.2018)

21. Варкасов Р.С. Разработка наглядной математической модели декодера Витерби для детектирования двоичного фазоманипулированного сигнала/ Р.С.Варкасов, И.М.Лернер//Научный Альманах.–2015. – № 8 (10).– С.749–751

22. Чернецова Е.А. «Системы и сети передачи информации» . Часть 1. Телекоммуникационные сети/Е.А.Чернецова– СПб.: Изд-во РГГМУ.–2013.–156 с.

23. LLR vs. HardDecisionDemodulation [Электронный ресурс] – Режим доступа: www.mathworks.com/help/comm/examples/llr-vs-hard-decision-demodulation.html

```

%Программа для моделирования прохождения сигнала с фазовой
манипуляцией по
%каналу с гауссовым шумом
disp('Исходное сообщение');
s='При «мягком» решении количество ошибок, выявленных декодером,
будет меньше, чем при «жестком» решении';
disp(s);
%сохранение введенной строки в ascii коде
save s1 s-ascii;
len4=length(s);% Определение длины исходной текстовой строки в ascii
коде
ms=load ('s1');% запись ascii кода в переменную ms
len=length(ms);% Определение длины переменной ms
% Перевод десятичных чисел ascii кода в строки 11-битовых символов
z=dec2base(ms,2,11);
[m,n]=size(z);% Определение размера переменной z, в которую записаны
11-битовые символы
% Цикл формирования из 11-битовых строк одной длинной строки для
подачи
% сигнала на вход кодера
i=1;
j=1;
while i<=len
kn(1,j:(j+10))=z(i,1:11);
i=i+1;
j=j+11;
end
len1=length(kn);
kn=reshape(kn,len1,1);

% Кодирование кодом Грея с использованием функции bin2gray
% цикл превращения бинарной строки в числовую переменную
for i=1:len1
ss(i)=str2num(kn(i));
end
[g1,map]=bin2gray(ss,'psk',4);

% цикл для кодирования потока битов кодом Грея
% Для матлаб старше 2006 года
%g1(1) = ss(1);
%for i = 2 : len1;
%x = xor((ss(i-1)),ss(i));
%g1(i)=x;

```

```

%end
% сверточное кодирование кодом со скоростью 1/2 с длиной кодового
ограничения 7
trell = poly2trellis(, []);
code1 = convenc(g1 ,trell);
% фазовая манипуляция
M =; % число позиций манипуляции
y1 = pskmod(code1,M); % Модуляция кодированного сигнала
y2 = pskmod(g1,M); % Модуляция некодированного сигнала
k = log2(M); % число бит на символ
EbNo =; % отношение энергии бита к СПМ шума
%N = Fs; частота дискретизации
% при задании SNR необходимо учесть скорость кода
%adjSNR = EbNo - 10*log10(1/codeRate) + 10*log10(k);
adjSNR = EbNo - 10*log10() + 10*log10(k);
NV = adjSNR+2;
y11 =awgn(y1,adjSNR,...
'measured',[],'dB'); % АБГШ-канал для кодированного сигнала по
жесткому решению
y21 = awgn(y1,NV,...
'measured',[],'dB'); % АБГШ-канал для кодированного сигнала по
мягкому решению
y22 =awgn(y2,EbNo+ 10*log10(k) ,...
'measured',[],'dB'); % АБГШ-канал для некодированного сигнала
% демодуляция
d1 = pskdemod(y11,M);% кодированного сигнала
d2 = pskdemod(y22,M);% некодированного сигнала
d3=pskdemod(y21,M);% кодированного сигнала по мягкому решению
l3=length(d1);
% ликвидация выбросов сигнала, превышающих логическую "1" из-за
воздействия
% шума в канале для мягкого решения
for i=1:l3
if d3(i)>1
d31(i)=1;
else
d31(i)=d3(i);
end
end
% Вариант мягкое декодирование
% Производим восьмиуровневое квантование зашумленного сигнала
[x,qcode] = quantiz(1-2*d31,[-.75 -.5 -.25 0 .25 .5 .75],...
[7 6 5 4 3 2 1 0]); % Значения в векторе qcode лежат в диапазоне от 0 до
2^3-1.

```

```

    % ликвидация выбросов сигнала, превышающих логическую "1" из-за
    воздействия
    % шума в канале
    for i=1:l3
    if d1(i)>1
    d11(i)=1;
    else
    d11(i)=d1(i);
    end
    end
    % для некодированного сигнала ликвидация выбросов сигнала,
    превышающих логическую "1" из-за воздействия
    % шума в канале
    for i=1:length(d2)
    if d2(i)>1
    d4(i)=1;
    else
    d4(i)=d2(i);
    end
    end
    % декодирование по алгоритму Витерби с ликвидацией выбросов
    tblen = 35; % глубина просмотра решетки кода при принятии решения
    decoded1 = vitdec(d11,trel1,tblen,'cont','hard');
    decoded2 = vitdec(qcode',trel1,tblen,'cont','soft',3);

    % Сдвиг декодированного сигнала на величину задержки
    len3=length(decoded1);
    t=tblen;
    len2 =len3-t;
    dec1(1:len2)=decoded1((t+1):len3);
    dec1((len2+1):len3)=ss((len2+1):len3);
    dec2(1:len2)=decoded2((t+1):len3);
    dec2((len2+1):len3)=ss((len2+1):len3);

    % Цикл перевода потока декодированных битов в строку символов
    l1=length(dec1);
    for i=1:l1
    si(i)=num2str(dec1(i));
    end
    i=1; j=1;
    % Цикл перевода потока декодированных битов в десятичные числа ascii
    кода
    while j<=len
    str=si(i:(i+10));
    a2(j)=bin2dec(str);

```

```

i=i+11; j=j+1;
end

% Цикл перевода потока мягкого решения в строку символов
l1=length(dec2);
for i=1:l1
si(i)=num2str(dec2(i));
end
i=1; j=1;
% Цикл перевода потока декодированных битов в десятичные числа ascii
кода
while j<=len
str=si(i:(i+10));
a4(j)=bin2dec(str);
i=i+11; j=j+1;
end
% Цикл перевода потока демодулированных некодированных битов в
строку символов
l1=length(d4);
for i=1:l1
si(i)=num2str(d4(i));
end
i=1; j=1;
% Цикл перевода потока некодированных битов в десятичные числа ascii
кода
while j<=len
str=si(i:(i+10));
a3(j)=bin2dec(str);
i=i+11; j=j+1;
end
disp('Полученное сообщение без использования кодирования ')
s7=str2mat(a3); % превращение декодированного ascii кода в строку
disp(s7);
disp('Полученное сообщение при коррекции ошибок кодером (7, [133
171]) при жесткой схеме принятия решения ')
s5=str2mat(a2); % превращение декодированного ascii кода в строку
disp(s5);
disp('Полученное сообщение при коррекции ошибок кодером (7, [133
171]) при мягкой схеме принятия решения ; ')
s6=str2mat(a4); % превращение декодированного ascii кода в строку
disp(s6);
% Вычисление энергетического выигрыша при применении жесткой
схемы
z1=sum(sqrt((a2-ms).^2));
z2=sum(sqrt((a3-ms).^2));

```

```
disp('Ассимптотическая эффективность алгоритма жесткого  
декодирования (в дБ) ');  
rez1=10*log10(z2./z1);  
disp(rez1);  
% Вычисление энергетического выигрыша при применении мягкой схемы  
z3=sum(sqrt((a4-ms).^2));  
rez2=10*log10(z2./z3);  
disp('Ассимптотическая эффективность алгоритма мягкого  
декодирования (в дБ) ');  
disp(rez2);
```