МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН
ФЕРГАНСКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра «Телекоммуникационный инжиниринг»

ТЕОРИЯ ИНФОРМАЦИИ И КОДИРОВАНИЯ

КОНСПЕКТ ЛЕКЦИЙ

Для студентов бакалавриатуры

Фергана 2016

Цель методического указания – методическое обеспечение процесса проведения лекционных работ по дисциплине «Теория информации и кодирования». Конспект лекций рассчитан на использование в учебном процессе при подготовке бакалавров по направлению «Телекоммуникации». Продолжительность лекций, приведенных в данном конспекте лекций, составляет 36 академических часов.

Рассмотрен на заседании учебно-методического совета факультета «Телекоммуникационные технологии и профессиональное образование» (протокол № от 2016 г.)

Председатель совета: доцент Кулдашев О.Х.

Рассмотрен на заседании кафедры «Телекоммуникационный инжиниринг» (протокол № от 2016 г.)

Зав.кафедры: доцент Жураев Н.М.

Составитель: ст.преп. Егиталиев З.М.

Рецензенты:

Доцент ФФ ТУИТ:Умаралиев Н.

Доцент ФерПИ: Мамасодиков Ю.

Лекция №1. Системы связи и теорема информации.
Системы передачи сообщений: информация, сообщения, сигнал

Объектом передачи в любой системе передачи информации является сообщение, несущее какую-либо информацию. Каждый из нас неоднократно употреблял выражение "масса информации", однако немногие знают, что можно измерять информацию количественно. Прежде чем вводить систему формул и чисел, рассмотрим пример. Пусть 10 июня мы услышали сообщение бюро прогнозов: "Осадков в виде снега завтра в Москве не будет". За последние 100 лет 10 июля в Москве снега, вероятно, ни разу не было; поэтому услышанное нам и сообщение содержит в себе очень мало нового — мало информации. Если бы, однако, мы, зная, что работа бюро прогнозов надежна, услышали, что "завтра будут осадки в виде снега", то в этом сообщении для нас содержалось бы гораздо больше информации, чем в предыдущем. Таким образом, сообщение о том, что произойдет событие, которое должно произойти почти наверняка, содержит в себе очень мало информации. Напротив, сообщение о том, что произойдет событие, которое почти наверняка произойти не должно, содержит много информации. Сообщение о некотором событии содержит тем больше информации, чем больше изменяется вероятность этого события после приема сообщения о нем, по сравнению с вероятностью того же события до того, как было принято соответствующее сообщение.

В общем случае мерой количества информации в сообщениях должна служить величина, измеряющая изменение вероятности события под действием сообщения. Любое сообщение может быть непрерывным (речь, музы ка) или дискретным письменный текст, цифровые данные).

Источником информации является отправитель сообщения, а потребителем — ее получатель. В одних системах передачи информации источником и потребителем информации может быть человек, а в других — различного рода автоматические устройства, ЭВМ и т. д.

Поступающее от источника сообщение u(t) в передатчике обрабатывается определенным образом, и формируется сигнал s(t), удобный для передачи по линии связи.

В телефонии, например, эта операция сводится просто к преобразованию звукового давления в пропорционально из меняющийся электрический ток микрофона. В телеграфии производится кодирование, в результате которого последовательность элементов сообщения (букв, цифр) преобразовывается в последовательность кодовых символов (0, 1, точка, тире).

Линией связи называется среда, используемая для передачи сигналов от передатчика к приемнику. В системах электрической связи — это пара проводов, кабель или волновод; в системах радиосвязи — область пространства, в которой распространяются электромагнитные волны от передатчика к приемнику; в системах оптической связи — оптическое волокно (ВОЛС).

При передаче сигнал может искажаться, и на него могут воздействовать помехи w(t). Приемник обрабатывает принятый сигнал x(t), искаженный помехой, и восстанавливает по нему переданное сообщение u(t). Обычно в приемнике выполняются операции, обратные тем, которые были осуществлены в передатчике.

Каналом связи принято называть совокупность технических средств, служащих для передачи сообщения от источника к потребителю. Этими средствами являются передатчик, линия связи и приемник.

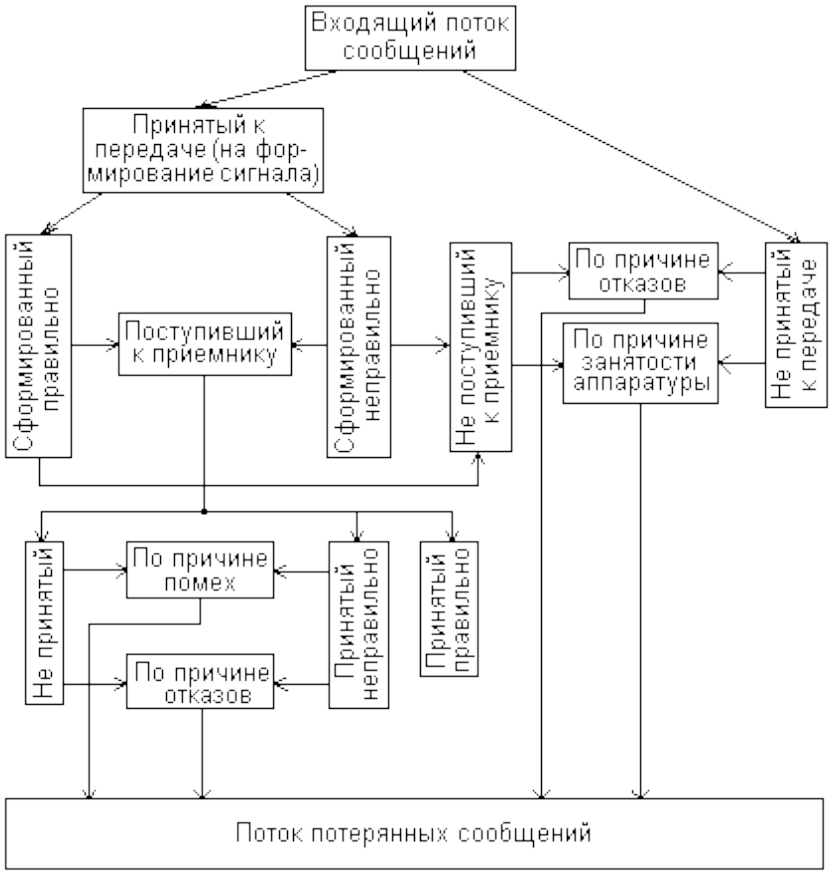
Канал связи вместе с источником и потребителем образуют систему передачи и обработки информации. Различают системы передачи дискретных сообщений (например, система телеграфной связи, система передачи цифровых данных) и системы передачи непрерывных сообщений (системы радиовещания, телевидения и т. д.).

Система передачи информации называется многоканальной, если она обеспечивает взаимно независимую передачу не скольких сообщений по одному общему каналу связи.

Под системой связи будем понимать совокупность технических средств, обеспечивающих передачу информации с заданными свойствами от различных источников различным получателям. Целенаправленная разработка системы связи может осуществляться при условии наличия критериев эффективности ее функционирования. Основной задачей системы связи является обеспечение максимальной скорости передачи при высоком качестве функционирования и экономичности системы.

Под качеством функционирования при этом понимается минимизация потерь информации, что в конечном итоге трансформируется в обеспечение высокой верности передачи.

Рассмотрим основные причины, приводящие к возможным потерям информации в системе связи. Они иллюстрируются схемой, представленной на рисунке:



На вход системы связи поступает поток сообщений, который далее может быть либо принят для передачи, либо не принят в связи с занятостью запоминающих или входных устройств системы связи. Поток сообщений, принятый для передачи, преобразуется в поток сигналов, предназначенных для передачи по каналу (будим полагать используемые в системе связи каналы дискретными и в качестве сигналов рассматривать последовательности символов кода). При этом преобразовании также могут возникать определенные потери информации, вызванные ненадежностью в основном кодирующих устройств и каналообразующей аппаратуры. Поток символов, поступивший из канала к приемнику может быть принят и не принят по причине неисправности аппаратуры или по причине ее занятости приемом других информационных потоков. Однако даже если поток был принят приемником, под действием помех в канале связи могут возникать такие ошибки, которые делают невозможным достоверное выявление информации. Последнее имеет место, если введенной в информацию избыточности оказалось недостаточно для исправления ошибок, возникших под действием помех в канале связи.

Таким образом, из потока сообщений, поступающих на входы системы связи, формируется некоторый поток потерянных сообщений. Независимо от места возникновения потерь информации основными

причинами потерь являются помехи в каналах связи, неисправность аппаратуры и перегрузка обслуживающих или запоминающих устройств.

Количественная оценка каждого из этих явлений может быть осуществлена с помощью теории вероятностей. Данное обстоятельство и позволяет сформировать единый информационный подход (т.е. подход с позиций теории информации) к оценке качества функционирования системы связи.

Для выяснения существа этого подхода рассмотрим подробнее все перечисленные составляющие потерь и их взаимодействие. Способность системы обеспечивать передачу информации с заданной верностью при воздействии помех в канале связи называют *помехоустойчивостью*.

Мешающее действие помех в дискретных каналах обычно оценивается некоторой моделью ошибок (в простейшем случае симметрично канала без памяти - вероятностью ошибки $P_{ош}$). Эффективным средством борьбы с этим фактором является введение избыточности в передаваемый сигнал, которую называют *информационной избыточностью*. Существует два типа избыточности: кодовая избыточность и избыточность повторения. *Под кодовой информационной избыточностью понимают* наличие дополнительных, не несущих информации о существе передаваемого сообщения, разрядов в кодовой комбинации.

Для повышения качества функционирования технических средств можно использовать многократное повторение однотипных блоков, сигналов и т.п., для получения требуемой помехоустойчивости или надежности. Такую избыточность *называют избыточностью повторения*.

Информационная избыточность повторения предполагает повторение информации в канале связи во времени, или повторение по временным или частотнымканалам.

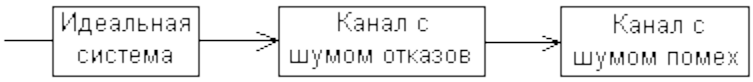
Наличие неисправности в аппаратуре приводит к ее неработоспособности. *Под надежностью* любой информационной системы понимают свойство системы выполнять свои функции "сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах". При оценке качества системы связи необходимо учитывать возможность возникновения сбоев и отказов. Под сбоем обычно понимается самоустраняющийся отказ, приводящий к кратковременному нарушению работоспособности. Под отказом понимают нарушение работоспособности аппаратуры.

Проблема надежности отличается от проблемы помехоустойчивости тем, что в случае отказа повторение одной и той же операции во времени не позволит обнаружить и исправить ошибку. Вместе с тем в системах с

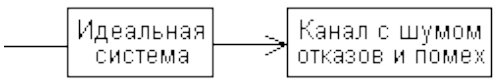
последовательными кодами одиночный отказ элемента может привести к неодиночной ошибке в выходном сигнале. Однако при соответствующем проектировании информационных систем основные методы обеспечения помехоустойчивой передачи информации могут быть применены к задаче конструирования надежных технических устройств. Также как и для повышения помехоустойчивости, для увеличения надежности необходимо вводить избыточность. В частности, с небольшими изменениями можно использовать большинство результатов теории кодирования при введении аппаратурной кодовой избыточности.

Проблема помехоустойчивости в определенной степени является противоречивой по отношению к проблеме надежности. Если для увеличения помехоустойчивости необходимо увеличивать избыточность передаваемой информации, то это приводит к усложнению системы и если вводимая избыточность не рассчитывалась на исправление ошибок, возникших из-за неисправности аппаратуры, то снижается надежность. Только оценивая помехоустойчивость и надежность единым критерием, можно оценить общую эффективность построения системы связи. Свойство системы в отношении помехоустойчивости и надежности можно связать с количеством информации, проходящим через систему.

Представим обобщенно мешающие воздействия в виде некоторых условных канала с шумом отказов аппаратуры и канала с шумом помех в линии связи (рисунок).



Это соответствует выделению идеальной системы без потерь информации и последовательно соединенных с ней каналов с шумом отказов и помех. Объединим оба последних канала в единый канал с шумом (рисунок).



Количество информации, передаваемое через такой канал

$$I(Z,U)=H(Z)-H(Z/U),$$

где H(Z)-энтропия на выходе канала, H(Z/U)-условная энтропия приема ансамбля сообщений Z при условии наличия на входе канала ансамбля сообщений U.

Величину H(Z/U) можно определить, если с учетом имеющих место в

канале помех и конкретной конфигурации аппаратной реализации системы определить вероятность потерь P_0, вызванных ошибками выявления сообщения, возникающей из-за помех и отказов.

Однако для получения общей вероятности потерь в системе необходимо учесть еще потери, вызванные отказом в обслуживании.

Расчет вероятности потерь по причине отказа в обслуживании можно вести исходя из известных в теории массового обслуживания соотношений.

Снижение этих потерь также может быть осуществлено путем введения избыточности в обслуживании. Она позволяет уменьшать время обслуживания информационных потоков, что в ряде случаев очень важно, т.к. в задачах, связанных с оперативным управлением, регулированием или контролем, существенную роль играет старение информации, поэтому задержки могут оказаться эквивалентными потере информации.

Значительную сложность представляет выбор между избыточностью в информации и избыточностью в обслуживании. Информационная избыточность приводит к увеличению времени обслуживания каждого сообщения. Поэтому для компенсации потерь, связанных с отказом в обслуживании, вносится избыточность в обслуживании (увеличивают объем памяти буферных запоминающих устройств; число декодеров и т.д). Это усложняет аппаратуру и снижает надежность.

Поэтому серьезной задачей является определение оптимальных соотношений по всем видам избыточности. Комплексный информационный подход к оценке потерь информации с учетом всех сторон функционирования технических средств позволяет добиться наивысшей эффективности работы системы связи.

Сообщение и сигнал. Канал связи

Под информацией понимают сведения о каком - либо явлении, событии, объекте. Информация, выраженная в определенной форме, представляет собой сообщение, иначе говоря, сообщение — это то, что подлежит передаче. Сигнал является материальным носителем сообщения.

В качестве сигнала можно использовать любой физический процесс, изменяющийся в соответствии с передаваемым сообщением. Существенно то, что сигналом является не сам физический процесс, а изменение отдельных параметров этого процесса. Указанные изменения определяются тем сообщением, которое несет данный сигнал. Правила этих изменений — код — обычно задаются заранее. В системах передачи и обработки информации сигнал предназначен для передачи информации от

отправителя к получателю. Код полностью известен как на передающей, так и на приемной сторонах — он устанавливается заранее.

Сообщения и соответствующие им сигналы бывают дискретными и непрерывными. Дискретное сообщение представляет собой последовательность отдельных элементов. Сигнал также представляет собой дискретную последовательность отдельных элементов, соответствующих элементам передаваемого сообщения. С такими сигналами мы имеем дело в вы числительной технике, в телеграфии. Так, при передаче телеграммы сообщением является текст телеграммы, элементами сообщения — буквы, сигналами — кодовые комбинации, соответствующие этим буквам.

Непрерывное сообщение — это некоторая физическая величина (звуковое давление, температура и т. п.), принимающая любые значения в заданном интервале. Сообщение с помощью датчиков преобразовывается в непрерывно изменяющуюся электрическую величину u(t) — видеосигнал или аналоговый сигнал. В большинстве случаев видеосигнал является низкочастотным колебанием, которое отображает передаваемое сообщение. Для удобства анализа видеосигнал часто условно рассматривают как сообщение, которое необходимо передать по каналу связи.

Для передачи на большое расстояние видеосигнал преобразовывается в высокочастотный сигнал (радиосигнал).

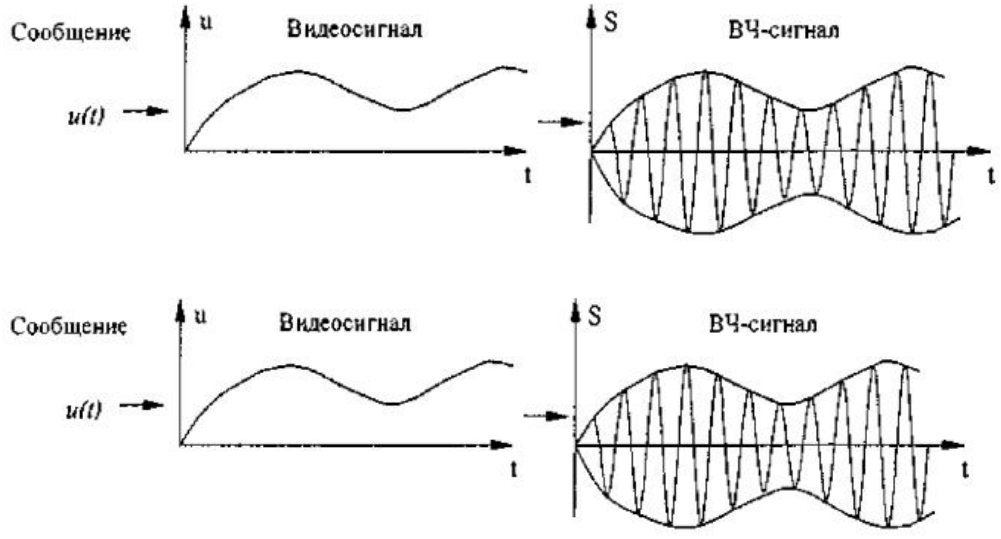


Рис.: Преобразование непрерывного сообщения в сигнал

Во многих случаях сигнал отображает временные процессы, происходящие в некоторой системе.

Поэтому описанием конкретного сигнала может быть некоторая функция времени. Определив, так или иначе, эту функцию, мы определяем

и сигнал. Однако такое полное описание сигнала требуется не всегда. Для решения ряда вопросов достаточно более общего описания в виде нескольких обобщенных параметров, характеризующих основные свойства сигнала, подобно тому, как это делается в системах транспортирования. Указывая габариты и вес, мы характеризуем основные свойства предмета с точки зрения условий его транспортирования; другие свойства (например, цвет) с этой точки зрения являются несущественными.

Сигнал есть также объект транспортирования, а техника передачи информации есть, по существу, техника транспортирования (передачи) сигналов по каналам связи. Поэтому целесообразно определить параметры сигнала, которые являются основными с точки зрения его передачи. Такими параметрами являются длительность сигнала, динамический диапазон и ширина спектра.

Всякий сигнал, рассматриваемый как временной процесс, имеет начало и конец. Поэтому длительность сигнала T является естественным его параметром, определяющим интервал времени, в пределах которого сигнал существует.

Характеристиками сигнала внутри интервала его существования являются динамический диапазон и скорость изменения сигнала.

Динамический диапазон определяется как отношение наибольшей мгновенной мощности сигнала к наименьшей:

$$Д = 10lg\frac{P_c\,max}{P_c\,min}\ (дБ).$$ осигнал

Динамический диапазон речи диктора равен 25 ÷ 30 дБ, вокального ансамбля — 45 ÷ 55 дБ, симфонического оркестра — 65 ÷ 75 дБ.

В реальных условия х всегда имеют место помехи. Для удовлетворительной передачи требуется, чтобы наименьшая мощность сигнала превышала мощность помех. Отношение сигнала к помехе характеризует относительный уровень сигнала. Обычно определяется логарифм этого отношения, который называется превышением сигнала над помехой. Это превышение и принимается в качестве второго параметра сигнала. Третьим параметром является ширина спектра сигнала F. Эта величина дает представление о скорости изменения сигнала внутри интервала его существования. Спектр сигнала может простираться в пределах очень большой полосы частот. Однако для большинства сигналов можно указать полосу частот, в пределах которой сосредоточена его основная энергия. Этой полосой и определяется ширина спектра сигнала.

Канал связи можно охарактеризовать так же, как и сигнал, тремя параметрами: временем, в течение которого по каналу ведется передача, динамическим диапазоном и полосой пропускания канала.

Общими признаками различных каналов являются следующие. Во-первых, большинство каналов можно считать линейными. В таких каналах выходной сигнал представляет собой просто сумму входных сигналов (принцип суперпозиции). Во-вторых, на выходе канала, даже при отсутствии полезного сигнала, всегда имеются помехи. В-третьих, сигнал при передаче по каналу претерпевает задержку по времени и затухание по уровню. И, наконец, в реальных каналах всегда имеют место искажения сигнала, обусловленные несовершенством канала.

Сигнал на выходе канала можно записать в следующем виде:

$$x(t) = \mu \cdot s(t - \tau) + w(t)$$

где s(t) — сигнал на входе канала; w(t) — помеха; и τ — величины, характеризующие затухание и время задержки сигнала.

Контрольные вопросы:

1. Какие основные причины, приводящие к возможным потерям информации в системе связи?
2. Ктоявляетсяисточникоминформации
3. Опишите процесс преобразования непрерывного сообщения в сигнал
4. Что является характеристиками сигнала внутри интервала его существования?
5. Какие общие признаки у различных каналов связи?
6. В каком виде можно записать сигнал на выходе канала?

Лекция №2. Информационные характеристики дискретного источника сообщений. Классификация дискретных источников. Количество информации. Энтропия.

Математической моделью множества возможных реализаций источника была дискретная или непрерывная случайная величина.

Для построения модели дискретных сообщений необходимо знать объем алфавита знаков (z1, z2, …, zl), из которых источником формируется сообщения, и вероятности создания им отдельных знаков с учетом возможной взаимосвязи между ними.

При доказательстве основных положений теории информации Шенноном использовалась модель, называемая эргодическим источником сообщений. Предполагается, что создаваемые им сообщения математически можно представить в виде эргодической случайной последовательности. Такая последовательность, как известно, удовлетворяет условиям стационарности и эргодичности. Первое означает, что вероятности отдельных знаков и их сочетаний не зависят от расположения последних по длине сообщения. Из второго следует, что статистические закономерности, полученные при исследовании одного достаточно длинного сообщения с вероятностью, близкой к единице, справедливы для всех сообщений, создаваемых источником. Из статистических характеристик в данном случае нас интересует средняя неопределенность в расчете на один знак последовательности.

Стационарный источниксообщений, выбирающий каждый знак формируемой последовательности независимо от других знаков, всегда является эргодическим. Его также называют источником без памяти.

На практике, однако, чаще встречаются источники, у которых вероятности выбора одного знака сообщения зависит от того, какие знаки были до этого (источники с памятью). Поскольку такая связь, как правило, распространяется на ограниченное число предыдущих знаков, для описания функционирования источника целесообразно использовать цепи Маркова.

В Марковском эргодическом источнике вероятность передачи того или иного сообщения однозначно определяется состоянием источника. После передачи сообщения источник переходит в новое состояние, которое зависит от предыдущего состояния и переданного сообщения.

Цепь Маркова порядка n характеризует последовательность событий, вероятности которых зависят от того, какие n событий предшествовали

данному. Эти n конкретных событий определяют состояние источника, в котором он находится при выдаче очередного знака.
Когда корреляционные связи наблюдаются только между двумя знаками (простая цепь Маркова), максимальное число различных состояний источника равно
При наличии корреляционной связи между тремя знаками состояния источника определяется двумя предшествующими знаками и т.д.
Аналитически можно получить выражения для энтропии источника сообщений при любой протяженности корреляционной связи.

Свойства эргодических последовательностей знаков

Характер последовательностей, формируемых реальным источником сообщений, зависит от существующих ограничений на выбор знаков. Они выражаются в том, что вероятности реализации знаков различны и между ними существуют корреляционные связи. Эти ограничения приводят к тому, что вероятности формируемых последовательностей существенно различаются.
Пусть, например, эргодический источник без памяти последовательно выдает знаки z1, z2, z3в соответствии с вероятностями 0,1; 0,3; 0,6. Тогда в образованной им достаточно длинной последовательности знаков мы ожидаем встретить в среднем на один знак z1 три знака z2 и шесть знаков z3. Однако при ограниченном числе знаков в последовательности существуют вероятности того, что она будет содержать;

- только знаки z1 (либо z2, либо z3);
- только знаки z1 и один знак z2 или z3;
- только знаки z2 и один знак z1 или z3;
- только знаки z3 и один знак z1 или z2;
- только знаки z1 и два знака z2 или z3 и т.д.

С увеличением числа знаков вероятности появления таких последовательностей уменьшается.
Фундаментальные свойства длинных последовательностей знаков, создаваемых эргодическим источником сообщений, отражает следующая теорема: как бы ни малы были два числа $\delta>0$ и $\mu>0$ при достаточно большом N, все последовательности могут быть разбиты на две группы.
Одну группу составляет подавляющее большинство последовательностей, каждая из которых имеет настолько ничтожную вероятность, что даже суммарная вероятность таких последовательностей очень мала и при достаточно большом Nбудет меньше сколь угодно малого числа δ. Эти последовательности называют нетипичными.

Вторая группа включает типичные последовательности, которые при достаточно большом N отличаются тем, что вероятности их появления практически одинаковы, причем вероятность p любой такой последовательности удовлетворяет неравенству

$$\left|\log(1/p)/N - H(Z)\right| < \mu \quad (1)$$

где H(Z) – энтропия источника сообщений.

Соотношение (4.1) называют также свойством асимптотической равномерности длинных последовательностей. Рассмотрим его подробнее.

Поскольку при N→∞ источник сообщений с вероятностью, сколь угодно близкой к единице, выдает только типичные последовательности, принимаемое во внимание число последовательностей равно 1/p. Неопределенность создания каждой такой последовательности с учетом их равновероятности составляет log(1/p). Тогда величина log(1/p)/N представляет собой неопределенность, приходящуюся в среднем на один знак. Конечно, эта величина практически не должна отличатся от энтропии источника, что и констатируется соотношением (4.1).

Приведем доказательство теоремы для простейшего случая эргодического источника без памяти. Оно непосредственно вытекает из закона больших чисел, в соответствии с которым в длинной последовательности из N элементов l(z1, z2, …, zl), имеющих вероятности появления p1, p2, …, pl содержится Np1 элементов z1, Np2элементов z2 и т.д.

Тогда вероятность p реализации любой типичной последовательности близка к величине

$$p = p_1^{p_1 N} p_2^{p_2 N} \ldots p_l^{p_l N}. \quad (2)$$

Логарифмируя правую и левую части выражения (4.2), получаем

$$\log p = N \sum_{i=1}^{N} p_i \log p_i, \quad (3)$$

откуда (при очень больших N)

$$\log(1/p)/N = H(Z). \quad (4)$$

Для общего случая теорема доказывается с привлечением цепей Маркова.

Избыточность источника

Избыточность определяет насколько хорошо в источнике сообщений используются возможные элементы сообщения. Наиболее экономнымявляется алфавит, использующий некоррелированные равновероятные символы. При наличии корреляционных связеймежду буквами (знаками) алфавита часть информации не являетсядля получателя непредвиденной. Эту информацию можно не передавать по каналу связи, она может быть восстановлена на приемном конце на основании статистических характеристик алфавита.
Мерой избыточности служит величина D, показывающая, насколько хорошо используются знаки данного алфавита источника:

$$D = \left[H_{max}(Z) - H(Z)\right] / \left[H_{max}(Z)\right], \quad (5)$$

где Hmax(Z) – максимально возможная энтропия, равная logl; H(Z) – энтропия источника.
Если избыточность источника равна нулю, то формируемые им сообщения оптимальны в смысле наибольшего количества переносимой информации. Для передачи определенного количества информации I при отсутствии помех в этом случае необходимо k1=I/[Hmax(Z)] знаков.
Поскольку энтропия сообщений, формируемых реальным источником, обладающим избыточностью, меньше максимальной, то для передачи того же количества информации I знаков требуется больше, а именно: k2=I/[H(Z)]>k1. Поэтому говорят также об избыточности знаков в сообщении или просто об избыточности сообщения, характеризуя ее тем же параметром D:

$$D = (k_2 - k_1)/k_2 = \left[H_{max}(Z) - H(Z)\right] / \left[H_{max}(Z)\right]. \quad (6)$$

Избыточность нельзя рассматривать как признак несовершенства источника сообщений. Обычно она является следствием его физических свойств. Ограничения, существующие в любом естественном языке, связаны, например, с особенностями артикуляции, не позволяющими формировать слова, состоящие из произвольных сочетаний букв.

Последствия от наличия избыточности сообщений неоднозначны. С одной стороны, избыточные сообщения требуют дополнительных затрат на передачу, например, увеличения длительности передач или расширения практической ширины спектра канала связи, что нежелательно. С другой стороны, при использовании сообщений, подчиняющихся априорно известным ограничениям, появляется возможность обнаружения и исправления ошибок, которые приводят к нарушению этих ограничений. Следовательно, наличие избыточности способствует повышению помехоустойчивости сообщений. Высокая избыточность большинства естественных языков обеспечивает, например, надежное общение людей даже при наличии у них акцентов и дефектов речи.

Однако при обмене информацией в автоматических системах естественная избыточность подлежит устранению. Это объясняется тем, что алгоритмы обнаружения и исправления ошибок, базирующихся на статистических закономерностях функционирования источника, оказываются слишком сложными для реализации их техническими средствами. В случае необходимости для повышения помехоустойчивости затем вводится «рациональная» избыточность, позволяющая обеспечить обнаружение и исправление наиболее вероятных ошибок простыми техническими средствами. При низком уровне помех в канале связи устранение избыточности приводит к увеличению скорости передачи информации и может дать значительный экономический эффект.

Пример 4.3. Определить возможный эффект от устранения избыточности припередачи текста на русском языке.

Максимальная энтропия текста на русском языке (с учетом пренебрежения при передаче различиями в буквах е и ё, ъ и ь) установлена ранее (см. пример 3.1) и равна 5 дв. ед. Там же определена энтропия с учетом неравномерного распределения вероятностей появления отдельных букв (4.35 дв. ед). Имея сведения о переходных вероятностях и исходя из предположения, что текст представляет собой простую цепь Маркова, можно установить, что энтропия уменьшается до 3.52 дв. ед. Учет всех ограничений в языке, включая связи между словами, позволяет оценить минимальную величину энтропии значением 1,5 дв. ед. Таким образом, избыточность русского языка составляет

$$D = \left[H_{max}(Z) - H(Z)\right] / \left[H_{max}(Z)\right] = (5 - 1{,}5)/5 = 0{,}7. \quad (7)$$

Это означает, что каналы связи, построенные без учета ограничений, существующих в языке, и способных передавать равновероятные буквы,

следующие друг за другом в любых сочетаниях, при передаче информации без помех текстом на русском языке используется всего на 30%. Полное устранение избыточности позволило бы повысить эффективности их использования более чем в 3 раза!

Производительность источника дискретных сообщений

Под производительностью источника сообщений подразумевают количество информации, вырабатываемое источником в единицу времени. Эту характеристику источника называют также скоростью создания сообщений или потоком входной информации. Поскольку возможное воздействие помех на источник сообщений принято учитывать эквивалентным изменением характеристик модели канала связи, то производительность источника сообщений равна энтропии источника, приходящейся на единицу времени.

Длительность выдачи знаков источником в каждом из состояний в общем случае может быть различной. Обозначим длительность выдачи знака zi, формируемого источником в состоянии Sq, через τqzi. Тогда средняя длительность выдачи источником одного знака

$$\tau_u = \sum_{q=1}^{R} p(S_q) \sum_{i=1}^{I} p_q(z_i) \tau_{qz_i}. \quad (8)$$

Производительность источника Ru(z) теперь можно выразить формулой

$$R_u(Z) = H(Z)/\tau_u. \quad (9)$$

Как следует из (4.5), повышение производительности источника возможно не только за счет увеличения энтропии, но и за счет снижения средней длительности формирования знака. Длительность знаков желательно выбирать обратно пропорциональными вероятностям их появления.

Если длительность выдачи знака не зависит от состояния источника, для всех знаков одинакова и равна τ, то τи=τ. Выражение для Ru(z)принимает вид

$$R_u(Z) = H(Z)/\tau. \quad (10)$$

Структурное (комбинаторное) определение количества информации (по Хартли).

Данное определение количества информации применимо лишь к дискретным сообщениям, причем таким, у которых символы равновероятны и взаимно независимы. Количество информации, содержащееся втакого рода сообщениях определяют из следующих соображений.

Пусть дан источник дискретных сообщений $A=(a_1, a_2,..., a_m)$, объем алфавита, которого равен *m*. Предположим, что каждое сообщение включает в себя *n* символов, при этом сообщения различаются либо набором символов, либо их размещением. Число различных сообщений N_0, состоящих из *n* символов, будет $N_0 = m^n$. Предположим, что все сообщения равновероятны и одинакова ценность этих сообщений.

Тогда легко подсчитать количество информации, которое несет каждое сообщение.

Вероятность появления каждого такого сообщения (P_n) может быть легко найдена:

$$P_n = \frac{1}{N_0} = m^{-n}. \quad (11)$$

И, следовательно, количество информации в одном сообщении (I_n), равно:

$$I_n = -\log_2 m^{-n} = n \cdot \log_2 m \text{ (бит)}. \quad (12)$$

Эту формулу предложил Р.Хартли в 1928 г., и она носит его имя. Разделив I_n на количество символов в сообщении (*n*), получим значение среднего количества информации (I_1), приходящееся на один символ сообщения:

$$I_1 = \log_2 m = -\log_2 P_m \text{ (бит / символ)}, \quad (13)$$

где P_m - вероятность появления одного символа сообщения.

Из соотношений (12) и (13) вытекают важные свойства дискретных сообщений, символы которых равновероятны и взаимно независимы.

1. Количество информации в сообщении пропорционально полному числу символов в нем – *n* и логарифму объема алфавита- *m*.
2. Среднее количество информации, приходящееся на один символ, зависит только от *m* – объема алфавита.

В реальных дискретных сообщениях символы часто появляются с различными вероятностями и, более того, часто существуют статистическая связь между символами, характеризующаяся условной вероятностью $P(a_i / a_j)$, которая равна вероятности появления символа a_i после символа a_j. Например, в тексте на русском языке вероятность появления различных

символов (букв) различна. В среднем, в тексте из 1000 букв буква О появляется 110 раз, Е – 87, А – 75, Т – 65, Н – 65, С – 55, кроме того, существуют статистические связи между буквами, скажем, после гласных букв не может появиться Ь или Ъ.

Исходя из этого, применение формулы вычисления количества информации по Хартли (12) и (13) не всегда корректно.

Статистическое определение количества информации (по Шеннону)

Этот подход к определению количества информации в сообщениях, учитывающий не равновероятное появление символов сообщения и их статистическую связь, был предложен К.Шенноном в 1946 г.

Рассмотрение этого метода удобно начать с определения количества информации в дискретных сообщениях, символы которых появляются не равновероятно, однако статистическая связь между символами отсутствует.

Пусть, как и ранее, дан источник дискретных сообщений $A = (a_1, a_2, ...a_m)$ с объемом алфавита равным *m,* который генерирует сообщение, состоящее из *n* символов. Допустим, что в этом сообщении символ a_1 встречается n_1 раз, символ $a_2 - n_2$ раз и так далее вплоть до символа a_m, который встречается n_m раз, причем очевидно, что

$$n_1 + n_2 + ... + n_m = n$$

При приеме одного символа a_1, как следует из (11), получаем количество информации I^0_{a1}:

$I^0_{a1} = -\log_2 P_{a1}$,

где P_{a1} - априорная вероятность появления символа a_1.

А количество информации I_{a1}, содержащееся в n_1 взаимно независимых символах a_1, будет равно:

$I_{a1} = -n_1 \cdot \log_2 P_{a1}$.

Аналогично, в n_2 символах a_2 содержится количество информации I_{a2}:

$I_{a2} = -n_2 \cdot \log_2 P_{a2}$,

и так далее вплоть до

$I_{am} = -n_m \cdot \log_2 P_{am}$.

Очевидно, что полное количество информации (I_n), содержащееся в сообщении из *n* символов, равно сумме количеств информации содержащихся во всех *m*символах алфавита.

$$I_n = I_{a1} + I_{a2} + ... + I_{am} = -(n_1 \cdot \log_2 P_{a1} + n_2 \cdot \log_2 P_{a2} + ... + n_m \cdot \log_2 P_{am}) = -\sum_{i=1}^{m} n_i \cdot \log_2 P_{ai}$$

(бит).

Разделив и умножив это выражение на n ($n \neq 0$),приведем это выражение к виду:

$$I_n = -n \cdot \sum_{i=1}^{m} \frac{n_i}{n} \cdot \log_2 P_{ai} \quad \text{(бит)}$$

Ясно, что отношение $\frac{n_i}{n}$ – это априорная вероятность появления i-го символа. Таким образом, при достаточно большомn, имеем: $\frac{n_i}{n} = P_{ai}$, причем $\sum_{i=1}^{m} P_{ai} = 1$, как сумма вероятностей полной группы событий.

Окончательно получим:

$$I_n = -n \cdot \sum_{i=1}^{m} P_{ai} \cdot \log_2 P_{ai} \quad \text{(бит)} \quad (14)$$

При этом среднее количество информации, приходящееся на один символ (H), будет равно:

$$H = \frac{I_n}{n} = -\sum_{i=1}^{m} P_{ai} \cdot \log_2 P_{ai} \left(\frac{\textit{бит}}{\textit{символ}} \right). \quad (15)$$

Определенная таким образом величина H называется *энтропией,* а формула (17) известна как формула Шеннона для энтропии источника дискретных сообщений. *Энтропия* определяет среднее количество информации, приходящееся на один символ дискретного сообщения.

В общем случае, символы, входящие в сообщения, могут появляться не только с различной вероятностью, но и быть статистически зависимыми. Статистическая зависимость может быть выражена условной вероятностью появления одного символа после другого.

Чтобы учесть статистические связи между символами, входящими в сообщение, вводят понятие *условной энтропии.*

Условная энтропия (H_k) определяется выражением

$$H_k = -\sum_{i=1}^{m} P_{ai} \cdot \sum_{j=1}^{m} P\left(a_j / a_i \right) \cdot \log_2 P\left(a_j / a_i \right) \left(\frac{\textit{бит}}{\textit{символ}} \right), \quad (16)$$

где $P\left(a_j / a_i \right)$ – условная вероятность появления символа a_j после символа a_i. Количество информации (I_k), содержащееся втакого рода сообщении длиной nсимволов, равно:

$$I_k = n \cdot H_k = -n \sum_{i=1}^{m} P_{ai} \cdot \sum_{j=1}^{m} P\left(a_j / a_i \right) \cdot \log_2 P\left(a_j / a_i \right) \textit{(бит)} \quad (17)$$

Свойства функции энтропии источника дискретных сообщений.

Свойства функции энтропии можно наглядно продемонстрировать на

примере источника дискретных сообщений *A*=(*a1, a2*) с объемом алфавита *m*равного 2, т.е. *m*=2. В этом случае справедливо $P_{a1} + P_{a2} = 1$, и выражение (1.8) может быть записано в виде:

$$H = -P_{a1} \cdot \log_2 P_{a1} - P_{a2} \cdot \log_2 P_{a2} = -P_{a1} \cdot \log_2 P_{a1} - (1 - P_{a1}) \cdot \log_2 (1 - P_{a1})$$ (бит/символ).

График этой функции имеет вид, представленный на рис. 1.1.

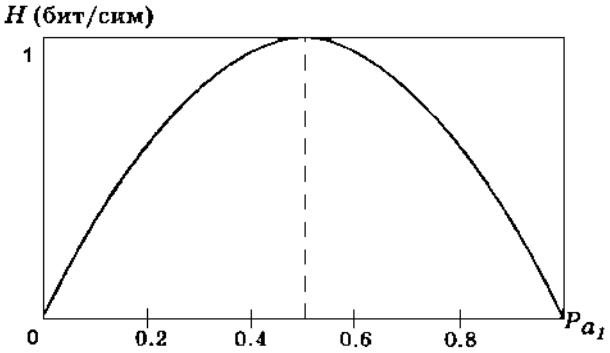


Рис.1.1. График функции энтропии.

Из графика видно, что обращение вероятности появления одного из возможных символов в 0 или 1 вносит полную определенность, энтропия обращается в 0 и сообщение о приёме такого символа не содержит в себе никакой информации.

При $P_{a1} = P_{a2} = 0{,}5$ получение конкретного символа наиболее неопределено и количество информации, содержащееся в поступившем символе, максимально.

Анализ формулы (14) и графика (Рис. 1.1) позволяет сформулировать основные свойства функции энтропии.

1. Энтропия источника дискретных сообщений есть величина вещественная, ограниченная и положительная.
2. Энтропия равна 0, если с вероятностью, равной единице, всегда выбирается один и тот же символ.
3. Энтропия максимальна, если все символы источника сообщений появляются независимо и равновероятно.

Интересно отметить, что сравнение выражений (12), (14) и (17) показывает, что формула Хартли является частным случаем формулы Шеннона при условии независимости и равновероятности появления символов в сообщении, а формула Шеннона, в свою очередь, является частным случаем условной энтропии при условии, что символы сообщения независимы. Действительно, из (17) следует, что количество информации (I_k),содержащееся в сообщении, состоящем из *n*неравновероятныхи взаимно зависимых символов определяется выражением:

$$I_k = -n\sum_{i=1}^{m} P_{ai} \cdot \sum_{j=1}^{m} P\left(a_j / a_i\right) \cdot \log_2 P\left(a_j / a_i\right).$$

Если символы сообщения взаимно независимы, то $P\left(a_j / a_i\right) = P(a_j)$ и $\sum_{j=1}^{m} P(a_j) = 1$, следовательно, это выражение преобразуется к виду:

$$I_k = -n\sum_{i=1}^{m} P_{ai} \cdot \sum_{j=1}^{m} P\left(a_j / a_i\right) \cdot \log_2 P\left(a_j / a_i\right) = -n\sum_{j=1}^{m} P(a_j) \cdot \log_2 P(a_j) = I_n$$

Последнее выражение соответствует формуле Шеннона.

В случае равновероятного появления символов сообщения $P(a_j) = \frac{1}{m}$, при j= 1, 2,…, m. (m– объём алфавита) и выше приведённое выражение для формулы Шеннона после соответствующего преобразования примет вид:

$$I_n = -n\sum_{j=1}^{m} P(a_j) \cdot \log_2 P(a_j) = -n\sum_{j=1}^{m} \frac{1}{m} \cdot \log_2 \frac{1}{m} = -n \cdot \log_2 \frac{1}{m} = n \cdot \log_2 m,$$

который соответствует формуле Хартли.

Таким образом, количество информации, определяемое по Хартли, т.е. при допущении полной независимости и равной вероятности появления отдельных символов сообщения, определяет максимально возможное количество информации в сообщении заданной длины (n).

При неравной вероятности появления символов (формула Шеннона) количество информации, содержащееся в сообщении заданной длины (n), снижается. Другим фактором, снижающим энтропию, а, следовательно, и количество информации в сообщении заданной длины (n), является наличие статистической зависимости между символами – корреляции.

Из-за корреляционных связей между символами и неравновероятного их появления количество информации в реальных сообщениях падает. Количественно эти потери информации характеризуются коэффициентом избыточности (R)

$$R = \frac{H_{\max} - H}{H_{\max}} = 1 - \frac{H}{H_{\max}} = 1 - \frac{H}{\log_2 m}, \qquad (18)$$

где $H_{\max}$ — максимальное количество информации, которое может содержать один символ сообщения, определяемое по формуле (1.6);

H — среднее количество информации, которое переносит один символ в реальных сообщениях;

m— число символов в алфавите источника сообщений (объём алфавита).

Избыточность $(R \neq 0)$ говорит о том, что число символов в сообщении больше, чем это требовалось бы при полном ихиспользовании, т.е. при условии, что символы появляются равновероятно и взаимно независимо.

Интересно отметить, что, для европейских языков избыточность составляет не менее 0.5.

Контрольные вопросы:

1. Опишитесвойстваэргодическихпоследовательностейзнаков
2. Чемопределяетсяизбыточностьисточника?
3. Чему равна максимальная энтропия текста на русском языке?
4. Опишите структурное (комбинаторное) определение количества информации (по Хартли)
5. Чтоназываетсяэнтропие?
6. Каковы основные свойства функции энтропии?

Лекция №3. Взаимная и условная информация и их характеристики.

С помощью взаимной информации мы можем определить количество информации о системе X, ведя наблюдение непосредственно за системой X. На практике часто бывает, что система X для наблюдения недоступна, и тогда ведут наблюдение за другой системой Y, как-то связанной с системой X. Например, вместо непосредственного наблюдения за космическим кораблем ведется наблюдение за системой сигналов, передаваемых его аппаратурой. Или наблюдение за футбольным матчем по телевизору.

Между системой X и Y имеются различия, которые могут быть двух видов:

1. Различия за счет того, что некоторые состояния системы X не находят отражения в системе Y (Y менее подробна, чем система X).
2. Различия за счет ошибок: неточностей измерения параметров системы X и ошибок при передаче сообщений.

Например, в черно-белом телевидении теряется цвет; влияние помех, которые вносят искажения.

То есть система Y отличается от системы X. Возникает вопрос: какое количество информации о системе X дает наблюдение системы Y? Данную информацию определяют как уменьшение энтропии системы X в результате получения сведений о системе Y.

$$I_{Y \to X} = H(X) - H(X/Y), \quad (1)$$

где H(X) - априорная энтропия до наблюдения, H(X/Y) - остаточная энтропия, после получения сведений, $I_{Y->X}$ - полная или средняя информация о системе X, содержащаяся в системе Y.

В общем случае, при наличии двух систем, каждая содержит относительно другой системы одну и ту же полную информацию. Покажем это:

$$H(X,Y) = H(X) + H(Y/X)$$
$$H(X,Y) = H(Y) + H(X/Y)$$

откуда

$$H(X) + H(Y/X) = H(Y) + H(X/Y)$$
$$H(X) - H(X/Y) = H(Y) - H(Y/X)$$
$$I_{Y->X} = I_{X->Y} = I_{X<->Y}$$

$I_{X<->Y}$ - называется полной взаимной информацией содержащейся в системах X и Y.

Посмотрим, во что обращается полная взаимная информация в крайних случаях полной независимости и полной зависимости систем. Если X и Y независимые системы, то H(Y/X)=H(Y), и * =0. Это и понятно, так как нельзя получить сведений о системе, наблюдая вместо нее другую систему, никак с нею не связанную.

Другой крайний случай X и Y полностью определяют друг друга, то есть совпадают. Тогда

H(X)=H(Y), H(X/Y)=H(Y/X)=0

и $I_{X<->Y} = I_X = I_Y = H(X) = H(Y).$

Рассмотрим случай, когда между X и Y имеется жесткая зависимость, но односторонняя: состояние одной из систем полностью определяет состояние другой, но не наоборот. По состоянию подчиненной системы вообще нельзя однозначно определить состояние другой. Очевидно, энтропия подчиненной системы меньше чем та, которой она подчиняется, так как она менее подробна. Тогда полная взаимная информация, содержащаяся в системах, из которых одна является подчиненной, равна энтропии подчиненной системы.

Пусть из двух систем X и Y подчиненной является Y. Тогда H(Y/X)=0, и $I_{X<->Y} = H(Y).$

Таким образом, полная взаимная информация, содержащаяся в системах, из которых одна является подчиненной, равна энтропии подчиненной системы.

Выведем выражение для информации $I_{X<->Y}$ не через условную энтропию, а через энтропию объединенной системы $H(X,Y)$ и энтропию отдельных ее частей H(X), H(Y).

$$I_{Y->X} = H(X) - H(X/Y),\ H(X,Y) = H(Y) + H(X/Y)$$
$$I_{Y->X} = H(X) + H(Y) - H(X,Y)$$

Выразим полную взаимную информацию через вероятности состояний системы. Для этого запишем значения энтропии отдельных систем через математическое ожидание:

H(X)=M[-log P(X)], H(Y)=M[-log P(Y)], H(X,Y)=M[-log P(X,Y)]

Тогда $I_{Y->X} = M[-\log P(X) - \log P(Y) + \log P(X,Y)]$

$$I_{Y->X} = M[\log \frac{P(X,Y)}{P(X)P(Y)}]$$ идалее

$$I_{Y<->X} = \sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i,y_j)\log \frac{p(x_i,y_j)}{p(x_i)p(y_j)} \quad (2)$$

Таким образом, получили полную взаимную информацию об одной системе X с помощью другой системы Y. (Информация “от системы к системе”). Кроме полной взаимной информации существуют и частные виды взаимной информации. Выражение (2) представим в виде

$$I_{Y<->X} = \sum_{j=1}^{m} p(y_j)\sum_{i=1}^{n} p(x_i/y_j)\log \frac{p(x_i/y_j)}{p(x_i)}$$

Тогда, вторая сумма и будет представлять частную информацию о системе X, получаемую с помощью отдельного события системы Y.

$$I_{y_j->X} = \sum_{i=1}^{n} p(x_i/y_j)\log \frac{p(x_i/y_j)}{p(x_i)} \quad (3)$$

Выражение (3) представляет частную информацию “от события к системе”. Далее можно определить частную информацию о событии xi, содержащуюся в событии yj (информация “от события к событию”):

$$I_{y_j->x_i} = \log \frac{p(x_i/y_j)}{p(x_i)} = \log \frac{p(x_i,y_j)}{p(x_i)p(y_j)} \quad (4)$$

Информация $I_{y_j \to x_i}$ симметрична $I_{y_j \to x_i} = I_{x_i \to y_j} = I_{y_j <-> x_i}$

При оценке неопределенности выбора часто необходимо учитывать статистические связи, которые в большинстве случаев имеют место как между состояниями двух или нескольких источников, объединенных в рамках одной системы, так и между состояниями, последовательно выбираемыми одним источником.

Определим энтропию объединения двух статистически связанных ансамблей *U* и *V*. *Объединение ансамблей* характеризуется матрицей *p(UV)* вероятностей $p(u_i v_i)$ всех возможных комбинации состояний$u_i (1<i<N)$ ансамбля *U* и состоянии $v_j (1<j<k)$ ансамбля *V:*

$$p(U,V) = \begin{Vmatrix} p(u_1 v_1) \dots p(u_i v_1) \dots p(u_N v_1) \\ p(u_1 v_j) \dots p(u_i v_j) \dots p(u_N v_j) \\ p(u_1 v_k) \dots p(u_i v_k) \dots p(u_N v_k) \end{Vmatrix} \quad (4)$$

Суммируя столбцы и строки матрицы (4), получим информацию об ансамблях *U и V* исходных источников *и* н *v.*

$$U = \begin{vmatrix} u_1 \dots u_i \dots u_N \\ p(u_1) \dots p(u_i) \dots p(u_N) \end{vmatrix}, V = \begin{vmatrix} v_1 \dots v_i \dots v_N \\ p(v_1) \dots p(v_i) \dots p(v_N) \end{vmatrix}$$

Вероятности $p(u_{ivj})$ совместной реализации взаимозависимых состоянии u_i и v_j, можно выразить через условные вероятности $p(u_i / v_j)$ или $p(v_j / u_i)$ в соответствии с тем, какие состояния принять *за* причину, а какие - за следствие:

$p(u_{ivj}) = p(u_i)\, p(vj / ui) = p(vj)\, p(ui / vj)$ (1.15)

где $p(u_i / v_j)$ - вероятность реализации состоянии u_i ансамбля *U* при условии, что реализовалось состояние v_j ансамбля *V;* $p(v_j / u_i)$ - вероятность реализации состояния v_j, ансамбля *V* при условии, что реализовалось состояние u_i, ансамбля *U.* Тогда выражение (3) для энтропий объединения принимает вид

$$H(U,V) = -\sum_{i=1}^{N}\sum_{j=1}^{k} p(u_i) p\left(v_j / u_i\right) \log\left[p(u_i) p\left(v_j / u_i\right)\right] = -\sum_{i=1}^{N} p(u_i) \log p(u_i) - \sum_{i=1}^{N} p(u_i) \sum_{j=1}^{k} p\left(v_j / u_i\right) \log\left(v_j / u_i\right) \quad (6)$$

Сумма $-\sum_{j=1}^{k} p\left(v_j / u_i\right) \log p\left(v_j / u_i\right)$ представляет собой случайную величину, характеризующую неопределенность, приходящуюся на одно состояние ансамбля *V* при условии, что реализовалось конкретное состояние u_i ансамбля *U.*

Назовем ее частной условной энтропией ансамбля *V* и обозначим $H_{ui}(V)$:

$$H_{Ui}(V) = -\sum_{j=1}^{k} p(v_j / u_i)\log(v_j / u_i) \quad (7)$$

При усреднении по всем состояниям ансамбли *U* получаем среднюю неопределенность, приходящуюся на одно состояние ансамбля *V* при известных состояниях ансамбля *U:*

$$H_U(V) = \sum_{i=1}^{N} p(u_i) H_{Ui}(V) \quad (8)$$

или

$$H_U(V) = -\sum_{i=1}^{N} p(u_i) \sum_{j=1}^{k} p\left(v_j / u_i\right) \log p\left(v_j / u_i\right) \quad (9)$$

Величину $H_u(v)$ называют полной условной или просто *условной энтропией ансамбля V* по отношению к ансамблю *U.*

Подставляя (9) в (6), получаем

$$H(UV) = H(U) + H_U(V) \quad (10)$$

Выражая в (1) $p(u_i v_j)$ через другую условную вероятность в соответствии с (5), найдем

$$H(UV) = H(V) + H_V(U) \quad (11)$$

где

$$H_V(U) = \sum_{j=1}^{k} p(v_j) H_{vj}(U) \quad (12)$$

и

$$H_{vj}(U) = -\sum_{i=1}^{N} p\left(u_i / v_j\right) \log p\left(u_i / v_j\right) \quad (13)$$

Таким образом, энтропия объединения двух статистически связанных ансамблей *U* и *V* равна безусловной энтропии одного ансамбля плюс условная энтропия другого относительно первого.

Распространяя правило (9) на объединение любого числа зависимых ансамблей, получим

$$H(UVZ...W) = H(U) + H_U(V) + H_{UV}(Z) + ... + H_{UVZ}(W) \quad (14)$$

Покажем теперь, что в объединении ансамблей условная энтропия любого ансамбля всегда меньше или равна безусловной энтропии того же ансамбля. Для объединения двух ансамблей *U* и *V* данное утверждение принимает вид соотношений

$$H_U(V) \le H(V) \quad (15)$$

$$H_V(U) \le H(U) \quad (16)$$

Из (10) и (15) следует, чти объединение двух произвольных ансамблей удовлетворяет соотношению

$$H(UV) \le H(U) + H(V) \quad (17)$$

Для объединения нескольких произвольных ансамблей соответственно имеем

$$H(UVZ...W) \le H(U) + H(V) + H(Z) + ... + H(W) \quad (18)$$

Действительно, наличие сведений о результатах реализации состоянии одного ансамбли никак не может увеличить неопределенность выбора состояния из другого ансамбля. Эта неопределенность может только уменьшиться, если существует взаимосвязь в реализациях состоянии из обоих ансамблей.

В случае отсутствия статистической связи в реализациях состояний u_i *из ансамбля U* и v_j из ансамбля *V* сведения о результатах выбора состояний из одного ансамбля не снижают неопределенности выбора состояний из другого ансамбля, что находит отражение в равенствах

$$H_U(V) = H(V), \; H_V(U) = H(U) \quad (19)$$

Если имеет место однозначная связь в реализациях состояний $u_i (1<i<N)$ из ансамбля *U* и $v_{j(1<j<N)}$ из ансамбля *V,* то условная энтропия любого из ансамблей равна нулю:

$$H_U(V) = 0, \; H_V(U) = 0 \quad (20)$$

Действительно, условные вероятности $p(u_i / v_j)$ и $p(v_j / u_i)$ в этом случае принимают значения, равные нулю или единице. Поэтому все слагаемые, входящие *в* выражения (7) и (13) для частных условных энтропий, равны нулю. Тогда в соответствии с (1.18) и (1.22) условные энтропии также равны нулю.

$$I(x_2 \mid y_2) = -\log p(x_2 \mid y_2) = -\log(7/10) \cong 0{,}515 \; дв.ед.$$

Равенства (20) отражают факт отсутствия дополни тельной неопределенности при выборе событий из второго ансамбля.

Уяснению соотношений между рассмотренными энтропиями дискретных источников информации (ансамблей) способствует их графическое отображение.

Пример 1 Определить энтропии *H(U), H(V),* H_u(V), H(UV), если задана матрица вероятностей состояний системы, объединяющей источники *u* и v:

$$p(v,u)=\begin{Vmatrix} 0,4 & 0,1 & 0 \\ 0 & 0,2 & 0,1 \\ 0 & 0 & 0,2 \end{Vmatrix}$$

Вычисляем безусловные вероятности состояний каждой системы как суммы совместных вероятностей по строкам и столбцам заданной матрицы.

Пример 2. Известны энтропии двух зависимых источников:

h(u) = 5 дв. ед., *H(V)* = 10 дв. ед, Определить, в каких пределах (будет изменяться условная энтропия $H_u(V)$ при изменений $H_v(U)$ в максимально возможных пределах.

При решении удобно использовать графическое отображение связи между энтропиями. Ммаксимального значения $H_{u(v)}$ достигает при отсутствии взаимосвязи н будет равно *H(V),* т.е., 10 дв.ед. По мере увеличения взаимосвязи $H_{u(V)}$ будет уменьшаться до значения *H(V)-H(U)* = 5 дв. ед. при этом $H_{v(U)} = 0$.

Контрольные вопросы:

1. Чему равна полная взаимная информация?
2. Что необходимо учитывать при оценке неопределенности выбора?
3. Чемхарактеризуетсяобъединениеансамблей?
4. Через что можно выразить вероятности совместной реализации взаимозависимых состоянии

Лекция №4. Методы кодирования информации. Теорема Шеннона-Фано и Хаффмана. Использование методов эффективного кодирования

С точки зрения теории информации кодирование — это процесс однозначного сопоставления алфавита источника сообщения и некоторой совокупности условных символов, осуществляемое по определенному правилу, а код (кодовый алфавит) — это полная совокупность (множество) различных условных символов (символов кода), которые могут использоваться для кодирования исходного сообщения и которые возможны при данном правиле кодирования. Число же различных кодовых символов составляющих кодовый алфавит называют объемом кода или объёмом кодового алфавита. Очевидно, что объём кодового алфавита не может быть меньше объёма алфавита кодируемого исходного сообщения. Таким образом, кодирование — это преобразование исходного сообщения

в совокупность или последовательность кодовых символов, отображающих сообщение, передаваемое по каналу связи.

Кодирование может быть числовым (цифровым) и нечисловым, в зависимости от вида, в котором представлены кодовые символы: числа в какой-либо системе счисления или иные какие-то объекты или знаки соответственно.

В большинстве случаев кодовые символы представляют собой совокупность или последовательность неких простейших составляющих, например, последовательность цифр в кодовых символах числового кода, которые называются элементами кодового символа. Местоположение или порядковый номер элемента в кодовом слове определяется его позицией.

Эффективное кодирование равновероятных символов сообщений.

Эффективное кодирование используется в каналах без шума, т.е. в таких каналах, где помехи отсутствуют, либо ими можно пренебречь. Основной задачей кодирования в таком канале является обеспечение максимальной скорости передачи информации, близкой к пропускной способности канала передачи.

В случае, если все символы алфавита кодируемого сообщения независимы и их появление равновероятно, построение оптимального эффективного кода не представляет трудностей. Действительно, пусть *H(x)* — энтропия исходного сообщения. Будем считать, что символы сообщения *(x_i)* равновероятны и объем алфавита исходного источника сообщений равен *m*. Следовательно, вероятность появления любого *i*-го символа данного сообщения *($P(x_i)$)* будет одинакова, т.е.

$P(x_i) = \frac{1}{m}$, $i=1,.., m,$

а энтропия сообщения равна *(H(x))*:

$$H(x) = -\sum_{i=1}^{m} P(x_i) \cdot \log_2 P(x_i) = \log_2 m ,$$

Если для кодирования используется числовой код по основанию *k*(объем алфавита элементов кодовых символов равен *k*), то энтропия элементов кодовых символов (H_1), при условии, что элементы символов кода появляются равновероятно и взаимнонезависимо, определится из соотношения:

$H_1 = \log_2 k$.

Тогда длина эффективного равномерного кода, т.е. число элементов в кодовом символе ($l_{эфф.}$), может быть найдена из соотношения:

$$l_{эфф} = \frac{H_x}{H_1} = \frac{\log_2 m}{\log_2 k} = \frac{\log_2 k^n}{\log_2 k} \cong n,$$

где $m = k^n$.

Эффективное кодирование неравновероятных символов сообщений

В случае, если символы кодируемого сообщения неравновероятны, в общем виде правило получения оптимального эффективного кода неизвестно. Однако из общих соображений можно представить принципы его построения.

Очевидно, что эффективное кодирование будет оптимальным, если неравномерное распределение вероятностей появления символов алфавита источника сообщений с помощью определенным образом выбранного кода переводят в равновероятное распределение вероятностей появления независимых элементов кодовых символов. В этом случае среднее количество информации, приходящееся на один элемент символа кода, будет максимальным. Для определения вида кода, удовлетворяющего этому требованию, можно рассмотреть «функцию стоимости» (цены передачи символов сообщения) в виде:

$$Q = \sum_{i=1}^{m} P(x_i) \cdot W_i,$$

где $P(x_i)$ — вероятность появленияi-го символа алфавита исходного кодируемого сообщения;

m— объем алфавита;

W_i— стоимость передачи i-го символа алфавита, которая пропорциональна длине кодового слова.

Эффективный код должен минимизировать функцию Q. Если передача всех элементов символов кода имеет одинаковую стоимость, то стоимость кодового символа будет пропорциональна длине соответствующего кодового символа. Следовательно, в общем случае (при неравновероятных символах исходного сообщения) код должен быть неравномерным, поэтому построение эффективного кода должно основываться на следующих принципах:

1. Длина кодового символа (n_i)должна быть обратно пропорциональна вероятности появления соответствующего символа исходного кодируемого сообщения (x_i);
2. Начало более длинного кодового символа не должно совпадать с началом более короткого (для возможности разделения кодовых символов без применения разделительных знаков);
3. В длинной последовательности элементы символов кода должны быть

независимы и равновероятны.

Теоретическое обоснование возможности эффективного кодирования передаваемых по каналу сообщений обеспечивает теорема, доказанная К. Шенноном и которая носит название первая теорема Шеннона или основная теорема Шеннона о кодировании для каналов без помех. Эта теорема гласит, что если источник сообщений имеет энтропию H [бит/символ], а канал обладает пропускной способностьюC[бит/сек] (пропускная способность характеризует максимально возможное значение скорости передачи информации), то всегда можно найти способ кодирования, который обеспечит передачу символов сообщения по каналу со средней скоростью

$V_{ср} = \left(\frac{C}{H} - \varepsilon \right)$ [символ/сек],

где ε — сколь угодно малая величина.

Обратное утверждение говорит, что передача символов сообщения по каналу со средней скоростью $V_{ср.} > H$ невозможна и, следовательно,

$V_{\max} = \frac{C}{H}$ [символ/сек].

Эта теорема часто приводится в иной формулировке: сообщения источника сообщений с энтропией H всегда можно закодировать последовательностями элементов кодовых символов с объемом алфавита kтак, что среднее число элементов символов кода на один символ кодируемого сообщения ($l_{\text{ср.}}$) будет сколь угодно близко к величине $H/ \log_2 k$, но не менее ее.

Не вдаваясь в доказательство этой теоремы, отметим, что эта теорема показывает возможность наилучшего эффективного кодирования, при котором обеспечивается равновероятное и независимое поступление элементов символов кода, а следовательно, и максимальное количество переносимой каждым из них информации, равное $\log_2 k$(бит/элемент) . К сожалению, теорема не указывает конкретного способа эффективного кодирования, она лишь говорит о том, что при выборе каждого элемента кодового символа необходимо чтобы он нес максимальное количество информации, а, следовательно, все элементы символов кода должны появляться с равными вероятностями и взаимно независимо.

Исходя из изложенных принципов, разработаны ряд алгоритмов эффективного кодирования как для взаимно независимых символов сообщения, так и для взаимозависимых. Суть их состоит в том, что они сокращают среднюю длину кодовых символов путем присвоения кодовых символов минимальной длины символам исходного сообщения, которые встречаются наиболее часто.

Алгоритм Шеннона-Фано.

Суть этого алгоритма, при использовании двоичного кода (объем алфавита элементов символов кода равен 2), заключается в следующем.

Все символы алфавита источника сообщений ранжируют, т. е. располагают в порядке убывания вероятностей их появления. Затем символы алфавита делятся на две группы приблизительно равной суммарной вероятности их появления. Все символы первой группы получают «0» в качестве первого элемента кодового символа, а все символы второй группы — «1». Далее группы делятся на подгруппы, по тому же правилу примерно равных суммарных вероятностей, и в каждой подгруппе присваивается вторая позиция кодовых символов. Процесс повторяется до закодирования всех символов алфавита кодируемого источника сообщений. В кодовый символ, соответствующий последней группе, добавляется в качестве последнего элемента «0» для того, чтобы начальный элемент символов кода не совпадал с конечным, что позволяет исключить разделительные элементы между символами кода.

Таблица 1 иллюстрирует процесс построения кода Шеннона–Фано на примере источника сообщений, алфавит которого состоит из восьми символов.

Таблица 1

Номер символа. (i)	Символы алфавита. (m_i)	Вероятности (P_i).	Номера Разбиений.	Кодовые Символы.
1	m_1	1/2	I	0
2	m_2	1/4	II	10
3	m_3	1/8	III	110
4	m_4	1/16	IV	1110
5	m_5	1/32		11110
6	m_6	1/64	V	111110
7	m_7	1/128		1111110
8	m_8	1/256	VI	11111110

На рис. 1 представлен граф кодирования (кодовое дерево), который показывает, как «расщепляется» ранжированная последовательность символов кодируемого источника сообщений на группы и отдельные

символы и какие кодовые символы присваиваются группам и отдельным символам алфавита источника сообщений на каждом шаге разбиения.

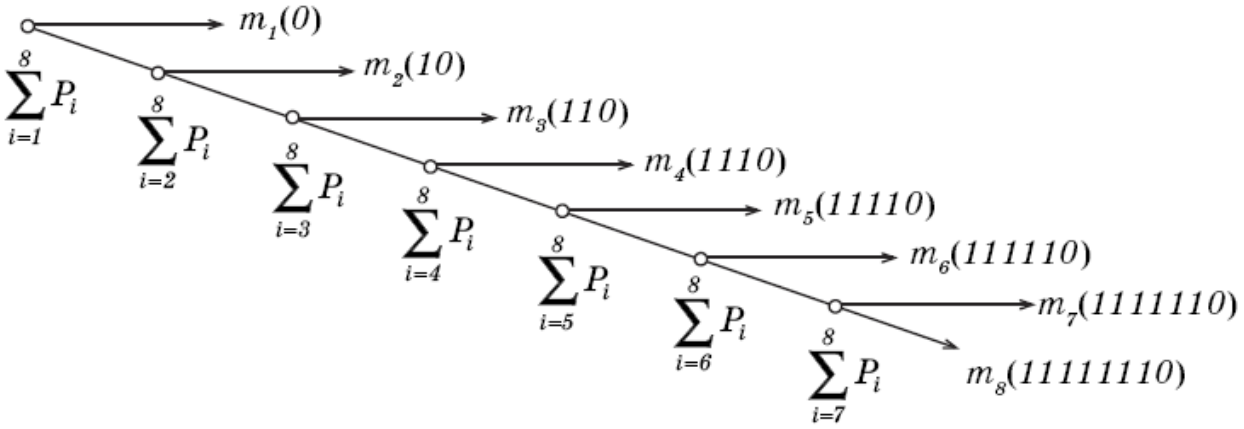


Рис 1. Граф кодирования по алгоритму Шеннона–Фано.

Алгоритм Шеннона–Фано применим и при иных числовых основаниях кода (k> 2). В этом случае алгоритм получения кода аналогичен рассмотренному примеру, только алфавит кодируемого источника сообщений разбивается на kгрупп и подгрупп примерно одинаковой суммарной вероятности.

Представляет интерес сравнение эффективного кодирования равномерным кодом и неравномерным кодом по алгоритму Шеннона–Фано.

В качестве примера рассмотрим предложенный выше (Табл. 1) источник сообщений с объемом алфавита равным 8 и соответствующими вероятностями появления отдельных символов (P_i). Для кодирования используем двоичный код (k= 2).

Энтропия рассмотренного источника сообщений (H_u) определяется по формуле Шеннона:

$$H_u = -\sum_{i=1}^{8} P_i \cdot \log_2 P_i = 1{,}96 \text{ (бит/символ).}$$

Максимально же возможное значение энтропии источника сообщений ($H_{u.\max}$), при условии равновероятного и взаимно независимого появления символов, находится по формуле Хартли:

$$H_{u.\max} = -\log_2 \frac{1}{m} = \log_2 8 = 3.$$

Следовательно, избыточность рассматриваемого источника сообщений (R_u) может быть найдена из соотношения:

$$R_u = 1 - \frac{H_u}{H_{u\max}} = 0.35.$$

Используя формулу для эффективного равномерного кода (2.4) при $k = 2$, получим значность равномерного двоичного кода (n_p):

$$n_p = \log_k m = \log_2 8 = 3,$$

и избыточность равномерного кода $(R_{рк})$:

$$R_{рк} = 1 - \frac{H_u}{n_p \cdot \log_2 k} \cong 0{,}35.$$

Энтропия элементов символов равномерного кода $(H_{1.P})$, т.е. количество информации, приходящееся на один элемент символа кода, будет равна:

$$H_{1.P} = \frac{H_u}{n_P} \cong 0{,}65 \quad \text{(бит / элемент символа кода).}$$

При использовании эффективного кодирования по алгоритму Шеннона-Фано соответствующие информационные параметры кода будут следующие.

Средняя длина неравномерного кода (n_H) определяется выражением:

$$n_H = \sum_{i=1}^{m} n_i \cdot p_i = 2,$$

где n_i— значность i - го кодового символа, соответствующего символу алфавита m_i.

Избыточность неравномерного кода $(R_{НК})$ определим из соотношения:

$$R_{НК} = 1 - \frac{H_u}{n_H \cdot \log_2 K} \cong 0{,}02$$

Энтропия элементов символов эффективного неравномерного кода (H_{1H}) может быть легко найдена:

$$H_{1H} = \frac{H_u}{n_H} = 0.98. \quad \text{(бит/элемент символа кода).}$$

При использовании эффективного кодирования по алгоритму Шеннона-Фано, энтропия элементов символов такого неравномерного кода на 50% выше чем энтропия элементов символов в случае использования равномерного кода. Если предположить, что скорость передачи по каналу элементов символов кода (W) одинакова для равномерного и неравномерного кода, то скорость передачи информации (V), определяемая выражением

$$V = H \cdot W,$$

где H — энтропия элементов символа кода,

также будет на 50% выше при использовании эффективного кодирования по алгоритму Шеннона–Фано по сравнению с равномерным кодированием.

Алгоритм Шеннона–Фано часто применяют и для блочного кодирования. При этом также существенно повышается эффективность кодирования. Для иллюстрации такого кодирования рассмотрим процедуру эффективного кодирования двоичным числовым кодом сообщений,

генерируемых источником сообщений с объемом алфавита равным 2 (m= 2), т.е. с алфавитом, состоящим только из двух символов m_1 и m_2с вероятностями появления $P(m_1)$ = 0,9 и $P(m_2)$ = 0,1 и, следовательно, с энтропией H = 0,47.

При посимвольном кодировании по алгоритму Шеннона–Фано эффект отсутствует, так как на каждый символ сообщения будет приходится один символ кода, состоящий из одного элемента.

Произведем теперь кодирование по алгоритму Шеннона–Фано блоков, состоящих из комбинаций двух символов источника сообщений, считая символы взаимнонезависимыми. Результат приведен в таблице 2.

Таблица 2.3.

Блоки	Вероятности	Номера разбиений	Кодовые комбинации
m_1m_1	0,81	I	1
m_1m_2	0,09	II	01
m_2m_1	0,09	III	001
m_2m_2	0,01		0001

Среднее число элементов символов кода на один символ исходного сообщения, равно 0,645, что значительно ниже, чем при посимвольном кодировании.

Кодирование блоков, соответствующих комбинациям из трех символов источника сообщений, дает еще больший эффект. Результат приведен в таблице 3.

Таблица 3.

Блоки.	Вероятности.	Номера разбиений.	Кодовые комбинации.

$m_1 m_1 m_1$	0,729	I	1
$m_2 m_1 m_1$	0,081	III	011
$m_1 m_2 m_1$	0,081	II	010
$m_1 m_1 m_2$	0,081	IV	001
$m_2 m_2 m_1$	0,009	VI	00011
$m_2 m_1 m_2$	0,009	V	00010
$m_1 m_2 m_2$	0,009	VII	00001
$m_2 m_2 m_2$	0,001		00000

В этом случае среднее число элементов символов кода на один символ исходного источника сообщений равно 0,53.

Теоретический минимум H = 0,47 может быть достигнут при кодировании блоков неограниченной длины.

Алгоритм Шеннона-Фано не всегда приводит к однозначному построению кода, так как при разбиении на подгруппы можно сделать большей по суммарной вероятности как верхнюю, так и нижнюю подгруппу. Этого недостатка лишен алгоритм Хаффмена, который гарантирует однозначное построение эффективного кода.

Алгоритм Хаффмена.

Суть этого алгоритма, при использовании двоичного кода, состоит в следующем. Все символы алфавита источника сообщений ранжируют, т.е. выписывают в столбец в порядке убывания вероятностей их появления. Два последних символа объединяют в один вспомогательный символ, которому приписывают суммарную вероятность.

Вероятности символов, не участвовавших в объединении, и вероятность вспомогательного символа вновь ранжируют, т.е. располагают в порядке убывания вероятностей в дополнительном столбце и два последних символа объединяются. Процесс продолжается до тех пор, пока не получим единственный вспомогательный символ с вероятностью равной единице. Пример кодирования по алгоритму Хаффмена приведен в таблице 4.

Таблица 4

Символы	Вероятности	Вспомогательные столбцы						
		1	2	3	4	5	6	7
m_1	0,22	0,22	0,22	0,26	0,32	0,42	0,52	1
m_2	0,20	0,20	0,20	0,22	0,26	0,32	0,42	
m_3	0,16	0,16	0,16	0,20	0,22	0,26		
m_4	0,16	0,16	0,16	0,16	0,20			
m_5	0,10	0,10	0,16	0,16				
m_6	0,10	0,10	0,10					
m_7	0,04	0,06						
m_8	0,02							

На рис. 2 показан граф кодирования (кодовое дерево), который иллюстрирует ранжирование символов на группы и отдельные символы, причем из точки, соответствующей вероятности 1, направляем две ветви: одной из них (с большей вероятностью) присваиваем символ 1, а второй – символ 0.

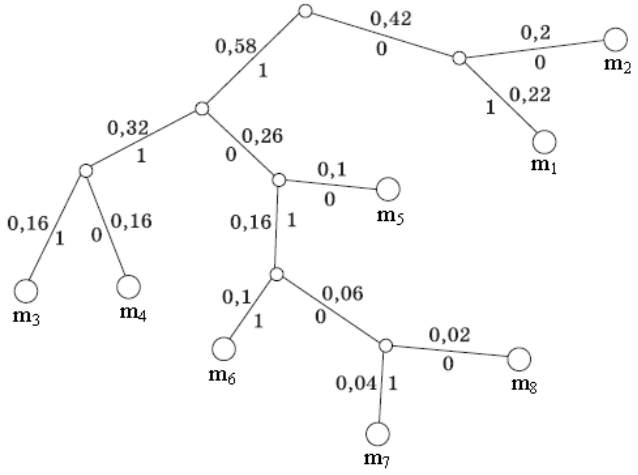


Рис 2. Граф кодирования по алгоритму Хаффмена.

Такое последовательное ветвление продолжим до тех пор, пока не дойдем до вероятности каждого символа. Двигаясь по кодовому дереву сверху вниз, можно записать для каждого символа источника сообщений соответствующую ему комбинацию (кодовый символ).

Различные символы, генерируемые источником сообщения, и соответствующие им кодовые символы представлены в таблице 5.

Таблица 5.

m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
01	00	111	110	100	1011	10101	10100

Этот алгоритм можно использовать и при ином числовом основании кода, а также использовать блоки, как это рассмотрено в алгоритме Шеннона-Фано.

Эффективность рассмотренных алгоритмов достигается благодаря присвоению более коротких кодовых комбинаций (кодовых символов) символам источника сообщений, вероятность которых более высока, и более длинных кодовых комбинаций - символам источника сообщений с малой вероятностью. Это ведет к различиям в длине кодовых символов и, как следствие, к трудностям при их декодировании. Для разделения отдельных кодовых символов можно использовать специальный разделительный элемент, но при этом существенно снижается эффективность кода, т.к. средняя длина кодового символа фактически увеличивается на один элемент символа кода.

Целесообразнее обеспечить декодирование без введения дополнительных элементов символов. Этого можно добиться, если в эффективном коде ни одна кодовая комбинация не будет совпадать с началом более длинной кодовой комбинации. Коды, удовлетворяющие этому условию, называют префиксными кодами (префиксом или началом называют первый элемент в кодовом символе, а последний элемент – окончанием или постфиксом).

Легко заметить, что коды, построенные по алгоритмам Шеннона–Фано или Хаффмена, являются префиксными.

Алгоритмы эффективного кодирования неравновероятных взаимозависимых символов сообщений

Устранение взаимной зависимости символов источника сообщения может быть осуществлено путем укрупнения алфавита исходного источника сообщения. Для этого подлежащие кодированию сообщения последовательно разбиваются на двух-, трех- или n-знаковые сочетания (блоки), вероятности которых известны, а затем эти сочетания кодируются в соответствии с алгоритмами Шеннона-Фано или Хаффмена.

Недостаток этого алгоритма состоит в том, что при его использовании не учитываются связи между символами, входящими в состав соседних сочетаний (блоков).

Этот недостаток может быть устранен кодированием по методу диаграмм, триграмм или в общем случае k-грамм. k-граммой называют последовательность из kсмежных символов сообщения. При $k=2$ сочетание смежных знаков называют диаграммой, при $k=3$ — триграммой и т.д.

В процессе кодирования по методу *k*-грамм производят непрерывное последовательное перемещение *k*-граммы по сообщению с шагом равным одному символу. Этот процесс (получение 3-х *k*-грамм) иллюстрируется рис.3, где x_i - символы сообщения.

Рис 3. Процесс непрерывного последовательного перемещения *k*-граммы по сообщению.

Если вероятности появления различных *k*-грамм известны, то их эффективное кодирование, в частности, может быть выполнено по алгоритмам Шеннона-Фано или Хаффмена.Конкретное значение *k*выбирается исходя из степени взаимозависимости между символами сообщения и сложности технической реализации кодирующих и декодирующих устройств.

Недостатки алгоритмов эффективного кодирования.

Основным недостатком этих алгоритмов является специфическое влияние помех на достоверность декодирования, которое проявляется в том, что одиночная ошибка в кодовой комбинации может перевести ее в другую кодовую комбинацию, не равную ей по длительности. Это может привести к неверному декодированию ряда последующих комбинаций, что называют треком ошибки, хотя существуют методы, позволяющие свести трек ошибки к минимуму.

Существенным недостатком является также сложность технической реализации систем эффективного кодирования, которые должны включать в себя буферные устройства и устройства накопления. Использование этих устройств вызвано тем, что длина кодовых комбинаций различна, а каналы связи эффективно работают только в том случае, если символы поступают на них с постоянной скоростью. Кроме этого, при кодировании блоками необходимо накапливать символы, прежде чем присвоить их совокупности какую-либо кодовую комбинацию.

Контрольные вопросы:

1. Кодирование – это…
2. Код (кодовыйалфавит) — это…
3. В каких каналах используется эффективное кодирование?
4. Из какого соотношения может быть найдена длина эффективного равномерного кода?
5. На каких принципах должно основываться построение эффективного кода?
6. В чем суть алгоритма Шеннона-Фено?

Лекция №5. Сжатие данных. Методы сжатия с потерями и без потерь

Сжатие данных (англ. *datacompression*) — алгоритмическое преобразование данных, производимое с целью уменьшения занимаемого ими объёма. Применяется для более рационального использования устройств хранения и передачи данных. Синонимы — *упаковка данных, компрессия, сжимающее кодирование, кодирование источника*. Обратная процедура называется восстановлением данных (распаковкой, декомпрессией).

Сжатие основано на устранении избыточности, содержащейся в исходных данных. Простейшим примером избыточности является повторение в тексте фрагментов (например, слов естественного или машинного языка). Подобная избыточность обычно устраняется заменой повторяющейся последовательности ссылкой на уже закодированный фрагмент с указанием его длины. Другой вид избыточности связан с тем, что некоторые значения в сжимаемых данных встречаются чаще других. Сокращение объёма данных достигается за счёт замены часто встречающихся данных короткими кодовыми словами, а редких — длинными (энтропийное кодирование). Сжатие данных, не обладающих свойством избыточности (например, случайный сигнал или белый шум, зашифрованные сообщения), принципиально невозможно без потерь

Принципы сжатия данных

В основе любого способа сжатия лежит модель источника данных, или, точнее, модель избыточности. Иными словами, для сжатия данных используются некоторые априорные сведения о том, какого рода данные сжимаются. Не обладая такими сведениями об источнике, невозможно сделать никаких предположений о преобразовании, которое позволило бы

уменьшить объём сообщения. Модель избыточности может быть статической, неизменной для всего сжимаемого сообщения, либо строиться или параметризоваться на этапе сжатия (и восстановления).

Методы, позволяющие на основе входных данных изменять модель избыточности информации, называются адаптивными. Неадаптивными являются обычно узкоспециализированные алгоритмы, применяемые для работы с данными, обладающими хорошо определёнными и неизменными характеристиками. Подавляющая часть достаточно универсальных алгоритмов являются в той или иной мере адаптивными.

Все методы сжатия данных делятся на два основных класса:

Сжатие без потерь

Сжатие данных без потерь (англ. *lossless data compression*) — метод сжатия данных (видео, аудио, графики, документов, представленных в цифровом виде), при использовании которого закодированные данные однозначно могут быть восстановлены с точностью до бита. При этом оригинальные данные полностью восстанавливаются из сжатого состояния. Этот тип сжатия принципиально отличается от сжатия данных с потерями. Для каждого из типов цифровой информации, как правило, существуют свои оптимальные алгоритмы сжатия без потерь.

Сжатие данных без потерь используется во многих приложениях. Например, оно используется во всех файловых архиваторах. Оно также используется как компонент в сжатии с потерями.

Сжатие без потерь используется, когда важна идентичность сжатых данных оригиналу. Обычный пример — исполняемые файлы и исходный код. Некоторые графические файловые форматы, такие как PNG, используют только сжатие без потерь; тогда как другие (TIFF, MNG) или GIF могут использовать сжатие как с потерями, так и без.

В общих чертах смысл сжатия без потерь таков. В исходных данных находят какую-либо закономерность и с учётом этой закономерности генерируют вторую последовательность, которая полностью описывает исходную. Например, для кодирования двоичных последовательностей, в которых много нулей и мало единиц, мы можем использовать такую замену:

$00 \rightarrow 0$

$01 \rightarrow 10$

$10 \rightarrow 110$

$11 \rightarrow 111$

В таком случае шестнадцать битов
00 01 00 00 11 10 00 00
будут преобразованы в тринадцать битов
0 10 0 0 111 110 0 0
Такая подстановка является префиксным кодом, то есть обладает такой особенностью: если мы запишем сжатую строку без пробелов, мы всё равно сможем расставить в ней пробелы — а значит, восстановить исходную последовательность. Наиболее известным префиксным кодом является код Хаффмана.
Большинство алгоритмов сжатия без потерь работают в две стадии: на первой генерируется *статистическая модель* для входящих данных, вторая отображает входящие данные в битовом представлении, используя модель для получения «вероятностных» (то есть часто встречаемых) данных, которые используются чаще, чем «невероятностные».
Статистические модели алгоритмов для текста (или текстовых бинарных данных, таких как исполняемые файлы) включают:

- Преобразование Барроуза — Уилера (блочно-сортирующая пре-обработка, которая делает сжатие более эффективным)
- LZ77 и LZ78 (используется DEFLATE)
- LZW
- Алгоритмы кодирования через генерирование битовых последовательностей:
- Алгоритм Хаффмана (также используется DEFLATE)
- Арифметическоекодирование

Многоцелевые
Кодирование длин серий — простая схема, дающая хорошее сжатие данных, которые содержат много повторяющихся значений

- LZW — используется в gif и во многих других.
- Deflate — используется в gzip, усовершенствованной версии zip и как часть процесса сжатия PNG.
- LZMA — используется в 7-zip.
- Сжатиеаудио
- Apple Lossless — ALAC (Apple Lossless Audio Codec)
- Audio Lossless Coding — такжеизвестенкак MPEG-4 ALS
- DirectStreamTransfer — DST
- DolbyTrueHD
- DTS-HD MasterAudio
- FreeLosslessAudioCodec — FLAC
- MeridianLosslessPacking — MLP

- Monkey'sAudio — Monkey’sAudio APE
- OptimFROG
- RealPlayer — RealAudioLossless
- Shorten — SHN
- TAK — (T)om’sverlustfreier (A)udio (K)ompressor (нем.)
- TTA — TrueAudioLossless
- WavPack — WavPacklossless
- WMA Lossless — WindowsMediaLossless

Сжатие графики

- ABO — AdaptiveBinaryOptimization
- BTPC
- CALIC
- CREW
- CTW
- DPCM
- GIF — (без потерь только для изображений содержащих не более 256 цветов)
- JBIG2 — (с потерями или без Ч/Б изображений)
- Lossless JPEG — (расширение стандарта сжатия JPEG, обеспечивающее сжатие без потерь)
- JPEG-LS — (стандарт сжатия без потерь/почти без потерь)
- JPEG 2000 — (в режиме сжатия без потерь)
- LOCO-I
- MRP
- PGF — Progressive Graphics File (сжатиес/безпотерь)
- PNG — PortableNetworkGraphics
- PWC
- TIFF — (исключая режимы сжатия с потерями[1])
- TMW
- Truevision TGA
- HDPhoto — (включая метод сжатия без потерь)

Сжатие видео

- Animationcodec
- CamStudioVideoCodec
- CorePNG
- FFV1
- Huffyuv
- Lagarith

- LCL
- MSU LosslessVideoCodec
- QbitLosslessCodec
- SheerVideo
- TSCC — TechSmithScreenCaptureCodec
- WMC — WaveletMediaCodec
- Motion JPEG 2000

Сжатие текстов

PPM — архиватор HA (автор HarryHirvola), использующий алгоритм PPM, известен высокой степенью сжатия на текстовых файлах; по этому параметру он превосходил первые версии появившегося несколько лет спустя RAR. Поэтому популярные в конце 90-х годов компакт-диски наподобие «Библиотека в кармане» использовали именно HA.

Примеры алгоритмов

- Семействоалгоритмов Лемпеля-Зива
- RLE (Run-lengthencoding — Кодирование длин серий)
- Примеры форматов и их реализаций[править | править вики-текст]
- универсальные — Zip, 7-Zip, RAR, GZip, PAQ и др.
- звук — FLAC (Free Lossless Audio Codec), Monkey's Audio (APE), TTA (True Audio), TTE, LA (LosslessAudio), RealAudio Lossless, WavPack идр.
- изображения — BMP, PNG
- видео — Huffyuv.

Сжатие с потерями

Сжатие данных с потерями — метод сжатия (компрессии) данных, при использовании которого распакованные данные отличаются отисходных, но степень отличия не является существенной с точки зрения их дальнейшего использования. Этот тип компрессии часто применяется для сжатия аудио- и видеоданных, статических изображений, вИнтернете, особенно в потоковой передаче данных, и цифровой телефонии. Альтернативой является сжатие без потерь.

Существуют две основных схемы сжатия с потерями:

В трансформирующих кодеках фреймы изображений или звука обычно трансформируются в новое базисное пространство и производится квантование. Трансформация может осуществляться либо для всего фрейма целиком (как, например, в схемах на основе wavelet-

преобразования), либо поблочно (характерный пример — JPEG). Результат затем сжимается энтропийными методами.
В предсказывающих кодеках предыдущие и/или последующие отсчеты данных используются для того, чтобы предсказать текущий отсчет изображения или звука. Ошибка между предсказанными данными и реальными вместе с добавочной информацией, необходимой для производства предсказания, затем квантуется и кодируется.
В некоторых системах эти две техники комбинируются путём использования трансформирующих кодеков для сжатия ошибочных сигналов, сгенерированных на стадии предсказания.

Сжатие с потерями против сжатия без потерь

Преимущество методов сжатия с потерями над методами сжатия без потерь состоит в том, что первые существенно превосходят по степени сжатия, продолжая удовлетворять поставленным требованиям, а именно — искажения д.б. в допустимых пределах чувствительности человеческих органов.
Методы сжатия с потерями часто используются для сжатия аналоговых данных — чаще всего звука или изображений.
В таких случаях распакованный файл может очень сильно отличаться от оригинала на уровне сравнения «бит в бит», но практически неотличим для человеческого уха или глаза в большинстве практических применений.
Много методов фокусируются на особенностях строения органов чувств человека. Психоакустическая модель определяет то, как сильно звук может быть сжат без ухудшения воспринимаемого качества звука. Недостатки, причинённые сжатием с потерями, которые заметны для человеческого уха или глаза, известны как артефакты сжатия.
Фотографии, записанные в формате JPEG, могут быть приняты судом (несмотря на то, что данные прошли сжатие с потерями).
Недостатки
При использовании сжатия с потерями необходимо учитывать, что повторное сжатие с потерями снижает качество, а декодирование увеличивает размер, не возвращая или не повышая качество. Поэтому для данных, которые когда-либо могут подвергнуться редактированию либо преобразованию в другие форматы (для совместимости или из-за невозможности платить патентные отчисления за декодирование или распространение сжатых данных), следует сохранять оригинал.

Методы сжатия данных с потерями (примеры)

- Компрессияизображений
- Снижение глубиныцвета
- Методглавныхкомпонент
- Фрактальноесжатие
- Сжатиенаосновепредсказателей
- JPEG-LS
- ДИКМ
- Иерархическаясеточнаяинтерполяция
- CALIC
- JPEG
- Вэйвлетнаякомпрессия
- JPEG 2000
- DjVu
- Дифференциальноесжатие
- Сжатие изображений на базе дифференциального анализа

Компрессия видео

- Motion JPEG
- Flash (поддерживает Motion JPEG)
- H.261
- H.263
- H.264
- H.265
- MNG (поддерживает Motion JPEG)
- MPEG-1 Part 2
- MPEG-2 Part 2
- MPEG-4 Part 2
- Ogg Theora (отличается отсутствием патентных ограничений)
- Sorensonvideocodec (*англ.*)
- VC-1 — открытая спецификация для формата WMV (Microsoft)

Компрессия звука

- MP3 — Определёнспецификацией MPEG-1
- OggVorbis (отличается отсутствием патентных ограничений и более высоким качеством)
- AAC, AAC+ — существует в нескольких вариантах, определённых спецификациями MPEG-2 и MPEG-4, используется, например, в Apple
- eAAC+ — формат, предлагаемый Sony, как альтернатива AAC и AAC+

- Musepack
- WMA — собственность Microsoft
- ADPCM
- ATRAC
- Dolby AC-3
- DTS
- MPEG-1 AudioLayer II
- VQF

При использовании сжатия без потерь возможно полное восстановление исходных данных, сжатие с потерями позволяет восстановить данные с искажениями, обычно несущественными с точки зрения дальнейшего использования восстановленных данных. Сжатие без потерь обычно используется для передачи и хранения текстовых данных, компьютерных программ, реже — для сокращения объёма аудио- и видеоданных, цифровых фотографий и т. п., в случаях, когда искажения недопустимы или нежелательны. Сжатие с потерями, обладающее значительно большей, чем сжатие без потерь, эффективностью, обычно применяется для сокращения объёма аудио- и видеоданных и цифровых фотографий в тех случаях, когда такое сокращение является приоритетным, а полное соответствие исходных и восстановленных данных не требуется.

Характеристики алгоритмов сжатия и их применимость

Коэффициент сжатия — основная характеристика алгоритма сжатия. Она определяется как отношение объёма исходных несжатых данных к объёму сжатых, то есть: $k = \frac{S_o}{S_c}$, где k — коэффициент сжатия, S_o — объём исходных данных, а S_c — объём сжатых. Таким образом, чем выше коэффициент сжатия, тем алгоритм эффективнее. Следует отметить:
если $k = 1$, то алгоритм не производит сжатия, то есть выходное сообщение оказывается по объёму равным входному;
если $k < 1$, то алгоритм порождает сообщение большего размера, нежели несжатое, то есть, совершает «вредную» работу.
Ситуация с $k < 1$ вполне возможна при сжатии. Принципиально невозможно получить алгоритм сжатия без потерь, который при любых данных образовывал бы на выходе данные меньшей или равной длины. Обоснование этого факта заключается в том, что поскольку число

различных сообщений длиной n бит составляет ровно 2^n, число различных сообщений с длиной меньшей или равной n (при наличии хотя бы одного сообщения меньшей длины) будет не больше 2^n. Это значит, что невозможно однозначно сопоставить все исходные сообщения сжатым: либо некоторые исходные сообщения не будут иметь сжатого представления, либо нескольким исходным сообщениям будет соответствовать одно и то же сжатое, а значит их нельзя отличить. Однако даже когда алгоритм сжатия увеличивает размер исходных данных, легко добиться того, чтобы их объём гарантировано не мог увеличиться более, чем на 1 бит. Тогда даже в самом худшем случае будет иметь место неравенство: $k \geqslant \frac{S_o}{S_o + 1}$

Делается это следующим образом: если объём сжатых данных меньше объёма исходных, возвращаем сжатые данные, добавив к ним «1», иначе возвращаем исходные данные, добавив к ним «0»).

Коэффициент сжатия может быть как постоянным (некоторые алгоритмы сжатия звука, изображения и т. п., например A-закон, μ-закон, ADPCM, усечённое блочное кодирование), так и переменным. Во втором случае он может быть определён либо для каждого конкретного сообщения, либо оценён по некоторым критериям:

- средний (обычно по некоторому тестовому набору данных);
- максимальный (случайнаилучшегосжатия);
- минимальный (случайнаихудшегосжатия);

или каким-либо другим. Коэффициент сжатия с потерями при этом сильно зависит от допустимой погрешности сжатия или *качества*, которое обычно выступает как параметр алгоритма. В общем случае постоянный коэффициент сжатия способны обеспечить только методы сжатия данных с потерями.

Допустимость потерь

Основным критерием различия между алгоритмами сжатия является описанное выше наличие или отсутствие потерь. В общем случае алгоритмы сжатия без потерь универсальны в том смысле, что их применение безусловно возможно для данных любого типа, в то время как возможность применения сжатия с потерями должна быть обоснована. Для некоторых типов данных искажения не допустимы в принципе. В их числе символические данные, изменение которых неминуемо приводит к изменению их семантики: программы и их исходные тексты, двоичные массивы и т. п.;

жизненно важные данные, изменения в которых могут привести к критическим ошибкам: например, получаемые с медицинской измерительной аппаратуры или контрольных приборов летательных, космических аппаратов и т. п.;
многократно подвергаемые сжатию и восстановлению промежуточные данные при многоэтапной обработке графических, звуковых и видеоданных.

Системные требования алгоритмов

Различные алгоритмы могут требовать различного количества ресурсов вычислительной системы, на которых они реализованы:

- оперативной памяти (под промежуточные данные);
- постоянной памяти (под код программы и константы);
- процессорноговремени.

В целом, эти требования зависят от сложности и «интеллектуальности» алгоритма. Общая тенденция такова: чем эффективнее и универсальнее алгоритм, тем большие требования к вычислительным ресурсам он предъявляет. Тем не менее, в специфических случаях простые и компактные алгоритмы могут работать не хуже сложных и универсальных. Системные требования определяют их потребительские качества: чем менее требователен алгоритм, тем на более простой, а следовательно, компактной, надёжной и дешёвой системе он может быть реализован.
Так как алгоритмы сжатия и восстановления работают в паре, имеет значение соотношение системных требований к ним. Нередко можно усложнив один алгоритм значительно упростить другой. Таким образом, возможны три варианта:
Алгоритм сжатия требует больших вычислительных ресурсов, нежели алгоритм восстановления.
Это наиболее распространённое соотношение, характерное для случаев, когда однократно сжатые данные будут использоваться многократно. В качестве примера можно привести цифровые аудио- и видеопроигрыватели.
Алгоритмы сжатия и восстановления требуют приблизительно равных вычислительных ресурсов.
Наиболее приемлемый вариант для линий связи, когда сжатие и восстановление происходит однократно на двух её концах (например, в цифровой телефонии).
Алгоритм сжатия существенно менее требователен, чем алгоритм восстановления.

Такая ситуация характерна для случаев, когда процедура сжатия реализуется простым, часто портативным устройством, для которого объём доступных ресурсов весьма критичен, например, космический аппарат или большая распределённая сеть датчиков. Это могут быть также данные, распаковка которых требуется в очень малом проценте случаев, например запись камер видеонаблюдения.

Алгоритмы сжатия данных неизвестного формата

Имеется два основных подхода к сжатию данных неизвестного формата:

- На каждом шаге алгоритма сжатия очередной сжимаемый символ либо помещается в выходной буфер сжимающего кодера как есть (со специальным флагом, помечающим, что он не был сжат), либо группа из нескольких сжимаемых символов заменяется ссылкой на совпадающую с ней группу из уже закодированных символов. Поскольку восстановление сжатых таким образом данных выполняется очень быстро, такой подход часто используется для создания самораспаковывающихся программ.
- Для каждой сжимаемой последовательности символов однократно либо в каждый момент времени собирается статистика её встречаемости в кодируемых данных. На основе этой статистики вычисляется вероятность значения очередного кодируемого символа (либо последовательности символов). После этого применяется та или иная разновидность энтропийного кодирования, например, арифметическое кодирование или кодирование Хаффмана, для представления часто встречающихся последовательностей короткими кодовыми словами, а редко встречающихся — более длинными.

Контрольные вопросы:

1. Сжатиеданных – это…
2. На чем основывается сжатие данных?
3. Сжатие данных без потерь – это…
4. Сжатие данных с потерями – это…
5. Что лежит в основе алгоритма Хаффмана?

Лекция №6. Кодирование в дискретных каналах с шумами. Теорема Шеннона.

В дискретном канале с шумами принятый символ y' не определяется однозначно переданным символом y. Существуют определенные вероятности переходов $p(y'_j|y_i)$ вообще говоря зависящие от ранее переданных и принятых символов.

Рассмотрим различные последовательности Y^n из символов, поступающих на вход канала. Каждая из таких последовательностей Y^n_i может перейти в различные последовательности Y'^n_j на выходе канала. Количество информации, содержащееся в такой принятой последовательности относительно переданной, в соответствии с (1) равно

$$I\left(Y_i^n, Y_j'^n\right) = \log \frac{p\left(Y_i^n \middle| Y_j'^n\right)}{p\left(Y_j^n\right)}, \quad (1)$$

а среднее количество информации $I\left(Y^n, Y'^n\right)$, приходящееся на последовательность из n символов, переданное по каналу с шумами, определяется как математическое ожидание по всем возможным передаваемым и принимаемым последовательностям и по всем состояниям канала, если оно существует. Это количество информации зависит как от свойств канала, так и от распределения вероятностей символов на входе канала.

Пусть на вход дискретного канала поступает v символов в единицу времени. Если при некотором распределении вероятностей символов на входе существует предел

$$I'(y,y') = \lim_{n\to\infty} \frac{v}{n} I\left(Y^n, Y'^n\right), \quad (2)$$

то он представляет скорость передачи информации по каналу. Пропускной способностью канала является верхняя грань $I'(y,y')$ по всем возможным распределениям вероятностей символов на входе.

В частности, для постоянного канала

$$I'(y,y') = v\sum_{i=1}^{m}\sum_{j=1}^{m'} p\left(y_i, y'_j\right)\log\frac{p\left(y_i, y'_j\right)}{p(y_i)P\left(y'_j\right)} =$$
$$= H'(y) - H'\left(y|y'\right) = H'(y') - H'\left(y'|y\right). \quad (3)$$

Вычисление пропускной способности даже для постоянных каналов в общем случае является довольно трудной задачей. Что касается каналов с памятью, то далеко не всегда их пропускная способность вообще может быть определена. Тем не менее, для дискретных отображений реальных каналов связи обычно удается построить математические модели в виде информационно устойчивых, т. е. имеющих определенную пропускную способность дискретных каналов. В частности, такими моделями служат каналы с ограниченной памятью, в которых переходные вероятности зависят только от конечного отрезка предыдущей последовательности символов , или каналы, описываемые конечным числом состояний, если текущее состояние можно определить по предыдущему состоянию и последнему переданному символу .

Очевидно, что многократное повторение сообщений ведет к росту помехозащищенности, но в тоже время значительно возрастает избыточность. Для полной достоверности передачи сообщений необходимо бесконечно увеличивать число повторных передач, при этом бесконечно возрастает избыточность и, следовательно, пропускная способность канала будет стремиться к нулю, что неприемлемо с практической точки зрения.

Другим путем обеспечения достоверной передачи сообщений по каналу с шумами является рациональное помехоустойчивое кодирование, теоретическим обоснованием которого служит теорема Шеннона. Эта теорема, применительно к каналам передачи информации, может быть сформулирована следующим образом.

Пусть дискретный канал обладает пропускной способностьюC, а дискретный источник сообщения имеет производительность H (бит/сек), т. е. источник сообщения создает H единиц информации в секунду. Если $H \leq C$, то существует такая система кодирования, что сообщения могут быть переданы по каналу с произвольно малой вероятностью ошибок (со сколь угодно малой ненадежностью). Однако, если $H > C$, то наименьшая ненадежность, которая может быть достигнута применением того или иного способа кодирования ограничена пределом разности ($H - C$).

Из теоремы следует, что для того, чтобы осуществить передачу с наперед заданной малой вероятностью ошибок, нужно уменьшить скорость передачи сообщений так, чтобы она стала равной или меньшей пропускной способности канала.

Передача дискретных сообщений по каналам связи.

Канал связи представляет собой совокупность технических средств и физических сред, предназначенную для передачи сообщений из одной

точки пространства в другую. Эта передача чаще всего осуществляется в условиях неизбежных помех. В результате воздействия помех каждый отправленный символ x_iможет быть опознан получателем как символ y_k, причем $y_k \neq x_i$. Такое событие называют ошибкой.

Передачу символов сообщения можно рассматривать как составной эксперимент, состоящий в отправлении символов сообщения x_i и получения символов y_k. С точки зрения теории информации физическое устройство канала несущественно, а свойства канала при этом полностью описываются матрицей переходных вероятностей $p\left(x_i / y_k\right)$ или $p\left(y_k / x_i\right)$,

где $p\left(x_i / y_k\right)$ есть вероятность передачи символа x_i, если зафиксирован полученный символ y_k,

$p\left(y_k / x_i\right)$ – вероятность получения символа y_k, если зафиксирован (передается) символ x_i.

При этом предполагается, что новые символы (сверх заданного объема алфавита m) не могут быть созданы под влиянием помех.

Следовательно,

$$\sum_{k=1}^{m} p\left(y_k / x_i\right) = 1;$$

$$\sum_{i=1}^{m} p\left(x_i / y_k\right) = 1.$$

Если помехи отсутствуют, то все диагональные элементы матрицы $p\left(y_k / x_i\right)$ или матрицы $p\left(x_i / y_k\right)$ равны единице, а остальные – нулю. При очень больших помехах все элементы матриц могут быть приблизительно одинаковыми.

При наличии помех, передача символа x_i не снимает полностью неопределенность относительно полученного символа y_k. Таким образом, передача символов по каналу описывается ниже перечисленными мерами неопределенности (энтропиями).

Неопределенность передаваемых символов при условии их независимости $(H(x))$:

$$H(x) = -\sum_{i=1}^{m} P(x_i) \cdot \log_2 P(x_i).$$

Неопределенность полученных символов $(H(y))$:

$$H(y) = -\sum_{k=1}^{m} P(y_k) \cdot \log_2 P(y_k).$$

Неопределенность получения символов при зафиксированном символе $x_i \left(H\left(y/x_i \right) \right)$:

$$H\left(y/x_i \right) = -\sum_{k=1}^{m} P\left(y_k/x_i \right) \cdot \log_2 P\left(y_k/x_i \right).$$

Эта величина называется частной энтропией принятых символов.

Полная энтропия принятых символов вычисляется усреднением $H\left(y/x_i \right)$ по вероятностям передаваемых символов x_i:

$$H\left(y/x \right) = \sum_{i=1}^{m} H\left(y/x_i \right) \cdot P\left(x_i \right)$$

Величину $H\left(y/x \right)$ называют средней условной энтропией принимаемых символов.

Неопределенность передаваемых символов при зафиксированном принятом символе $y_k \left(H\left(x/y_k \right) \right)$:

$$H\left(x/y_k \right) = -\sum_{i=1}^{M} P\left(x_i/y_k \right) \cdot \log_2 P\left(x_i/y_k \right)$$

Эта величина является частной энтропией передаваемых символов.

Полную энтропию передаваемых символов находят усреднением энтропии $H\left(x/y_k \right)$ по вероятностям принимаемых символов y_k:

$$H\left(x/y \right) = \sum_{k=1}^{m} H\left(x/y_k \right) \cdot P\left(y_k \right).$$

В соответствии с основным соотношением теории информации (4), прирост количества информации (I), связанный с приемом одного символа сообщения, определяется выражением:

$$I = \log_2 \frac{P_2}{P_1} = \log_2 P_2 - \log_2 P_1 = H(x) - H\left(x/y \right) \qquad (4)$$

где P_1 – априорная вероятность появления символа;

P_2 – апостериорная вероятность появления этого же символа. Справедливотакже соотношение:

$$I = H(y) - H\left(y/x \right). \qquad (5)$$

Из выражений (4) и (5) видно, что, по мере уменьшения помех, величина I будет стремиться к $H(x)$, а при увеличении помех будет стремиться к нулю.

Контрольные вопросы:

1. Что нужно сделать для осуществления передачи с наперед заданной малой вероятностью ошибок?
2. Какоесобытиеназываютошибкой?
3. Каналсвязипредставляетсобой…
4. Какая величина называется частной энтропией принятых символов?
5. Какая величина является частной энтропией передаваемых символов?

Лекция №7. Помехоустойчивое кодирование. Линейные блоковые коды.

Теория помехоустойчивого кодирования базируется на результатах исследований, проведенных Клодом Шенноном. Он сформулировал теорему для дискретного канала с шумом: при любой скорости передачи двоичных символов, меньшей, чем пропускная способность канала, существует такой код, при котором вероятность ошибочного декодирования будет сколь угодно мала.

Построение такого кода достигается ценой введения избыточности. То есть, применяя для передачи информации код, у которого используются не все возможные комбинации, а только некоторые из них, можно повысить помехоустойчивость приема. Такие кодыназывают ***избыточными*** или ***корректирующими***. Корректирующие свойства избыточных кодов зависят от правил построения этих кодов и параметров кода (длительности символов, числа разрядов, избыточности и др.).

В настоящее время наибольшее внимание уделяется двоичным равномерным корректирующим кодам. Они обладают хорошими корректирующими свойствами и их реализация сравнительно проста.

Наиболее часто применяются блоковые коды. При использовании блоковых кодов цифровая информация передается в виде отдельных кодовых комбинаций (блоков) равной длины. Кодирование и декодирование каждого блока осуществляется независимо друг от друга, то есть каждой букве сообщения соответствует блок из *n* символов.

Блоковый код называется ***равномерным***, если *n* (значность) остается одинаковой для всех букв сообщения.

Различают *разделимые* и *неразделимые* блоковые коды.

При кодировании *разделимыми* кодами кодовые операции состоят из двух разделяющихся частей: информационной и проверочной. Информационные и проверочные разряды во всех кодовых комбинациях разделимого кода занимают одни и те же позиции.

При кодировании *неразделимыми* кодами разделить символы выходной последовательности на информационные и проверочные невозможно.

Непрерывными называются такие коды, в которых введение избыточных символов в кодируемую последовательность информационных символов осуществляется непрерывно, без разделения ее на независимые блоки. Непрерывные коды также могут быть разделимыми и неразделимыми.

Общие принципы использования избыточности

Способность кода обнаруживать и исправлять ошибки обусловлена наличием избыточных символов. На ввод кодирующего устройства поступает последовательность из k информационных двоичных символов. На выходе ей соответствует последовательность из n двоичных символов, причем $n>k$. Всего может быть 2^k различных входных последовательностей и 2^n различных выходных последовательностей. Из общего числа 2^n выходных последовательностей только 2^k последовательностей соответствуют входным. Будем называть их *разрешенными* кодовыми комбинациями. Остальные $(2^n - 2^k)$ возможных выходных последовательностей для передачи не используются. Их будем называть *запрещенными* кодовыми комбинациями.

Искажение информации в процессе передачи сводится к тому, что некоторые из передаточных символов заменяются другими - неверными. Каждая из 2^k разрешенных комбинаций в результате действия помех может трансформироваться в любую другую. Всего может быть $2^k \cdot 2^n$ возможных случаев. В это число входит:

- 2^k случаев безошибочной передачи;
- $2^k \cdot (2^k - 1)$ случаев перевода в другие разрешенные комбинации, что соответствует необнаруживаемым ошибкам;
- $2^k \cdot (2^n - 2^k)$ случаев перехода в неразрешенные комбинации, которые могут быть обнаружены.

Часть обнаруживаемых ошибочных кодовых комбинаций от общего числа возможных случаев передачи соответствует:

$$K_{\text{обн}} = \frac{2^k \cdot (2^n - 2^k)}{2^k 2^n} = 1 - \frac{2^k}{2^n}.$$

Рассмотрим, например, обнаруживающую способность кода, каждая комбинация которого содержит всего один избыточный символ ($n=k+1$). Общее число выходных последовательностей составит 2^{k+1}, то есть вдвое больше общего числа кодируемых входных последовательностей. За подмножество разрешенных кодовых комбинаций можно принять, например, подмножество 2^k комбинаций, содержащих четное число единиц (или нулей). При кодировании к каждой последовательности из k информационных символов добавляется один символ (0 или 1), такой, чтобы число единиц в кодовой комбинации было четным. Искажение любого четного числа символов переводит разрешенную кодовую комбинацию в подмножество запрещенных комбинаций, что обнаруживается на приемной стороне по нечетности числа единиц. Часть обнаруженных ошибок составляет:

$$K_{\text{обн}} = 1 - \frac{2^k}{2^n} = 1 - \frac{2^k}{2^{k+1}} = \frac{1}{2}.$$

Пример кодирующего устройства с проверкой на четность показан на рис.

Основные параметры корректирующих кодов

Основными параметрами, характеризующими корректирующие свойства кодов являются *избыточность кода, кодовое расстояние, число обнаруживаемых* или *исправленных ошибок.*

Рассмотрим суть этих параметров.

Избыточность корректирующего кода может быть абсолютной и относительной. Под абсолютной избыточностью понимают число вводимых дополнительных разрядов

$r = n - k.$

Относительной избыточностью корректирующего кода называют величину

$$_{\text{отн}} = \frac{r}{n} = \frac{(n-k)}{n} = 1 - \frac{k}{n}$$

или

$$\frac{k}{n} = 1 - _{\text{отн.}}$$

Эта величина показывает, какую часть общего числа символов кодовой комбинации составляют информационные символы. Ее еще называют относительной скоростью передачи информации.

Если производительность источника равна H символов в секунду, то скорость передачи после кодирования этой информации будет равна

$$R = \frac{Hk}{n}$$

поскольку в последовательности из n символов только k информационных.

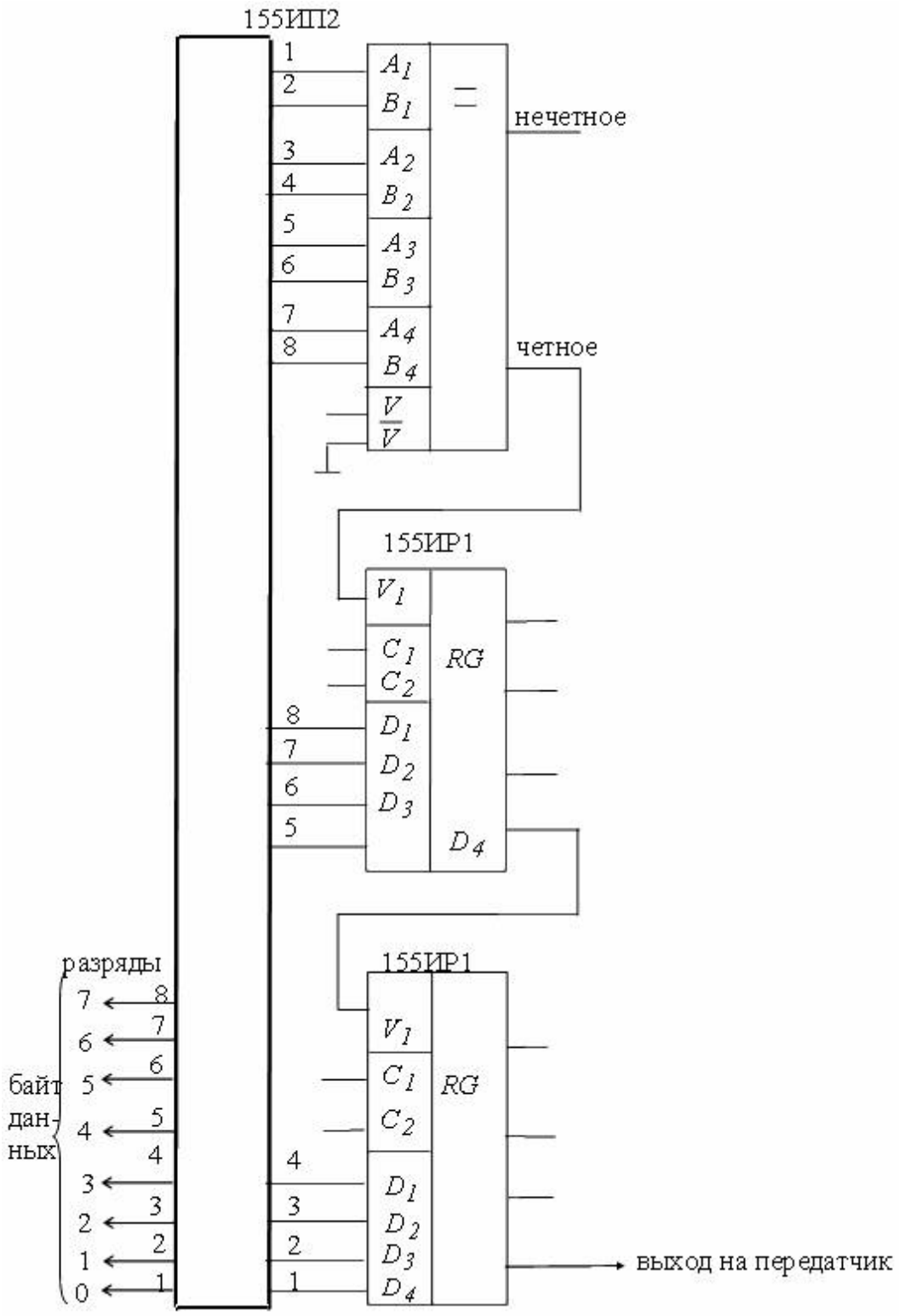


Рис. 1. Кодер с контролем на четность

Если число ошибок, которое нужно обнаружить или исправить, значительно, необходимо иметь код с большим числом проверочных символов. Скорость передачи информации при этом будет уменьшена, так как появляется временная задержка информации. Она тем больше, чем сложнее кодирование.

Кодовое расстояние характеризует степень различия любых двух кодовых комбинаций. Оно выражается числом символов, которыми комбинации отличаются одна от другой.

Чтобы получить кодовое расстояние между двумя комбинациями двоичного кода, достаточно подсчитать число единиц в сумме этих комбинаций по модулю 2.

Например:
$$\begin{array}{r} 101101 \\ \oplus\ 100100 \\ \hline 001001 \end{array} \qquad d=2.$$

Кодовое расстояние может быть различным. Так, в первичном натуральном безызбыточном коде это расстояние для различных комбинаций может различаться от единицы до *n*, равной значности кода.

Число обнаруживаемых ошибок определяется минимальным расстоянием $d_{\min}$ между кодовыми комбинациями. *Это расстояние называется хэмминговым*.

В без избыточном коде все комбинации являются разрешенными, $d_{\min}=1$. Достаточно только исказиться одному символу, и будет ошибка в сообщении.

Теорема. Чтобы код обладал свойствами обнаруживать одиночные ошибки, необходимо ввести избыточность, которая обеспечивала бы минимальное расстояние между любыми двумя разрешенными комбинациями не менее двух.

Доказательство. Возьмем значность кода *n*=3. Возможные комбинации натурального кода образуют следующее множество: 000, 001, 010, 011, 100, 101, 110, 111. Любая одиночная ошибка трансформирует данную комбинацию в другую разрешенную комбинацию. Ошибки здесь не обнаруживаются и не исправляются, так как $d_{\min}=1$. Если $d_{\min}=2$, то ни одна из разрешенных кодовых комбинаций при одиночной ошибке не переходит в другую разрешенную комбинацию.

Пусть подмножество разрешенных комбинаций образовано по принципу четности числа единиц. Тогда подмножества разрешенных и запрещенных комбинаций будут такие:

000, 011, 101, 110 - разрешенные комбинации;

001, 010, 100, 111 - запрещенные комбинации.

Очевидно, что искажение помехой одного разряда (одиночная ошибка) приводит к переходу комбинации в подмножество запрещенных комбинаций. То есть этот код обнаруживает все одиночные ошибки.

В общем случае при необходимости обнаруживать ошибки кратности t_0 - минимальное хэммингово расстояние должно быть, по крайней мере, на единицу больше t_0, то есть

$d_{\min} \geq t_0 + 1.$

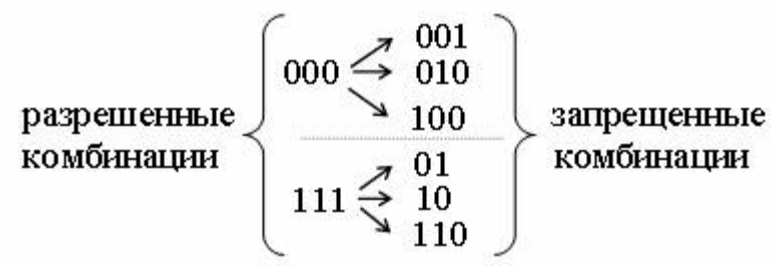
В этом случае никакая ошибка кратности t_0 не в состоянии перевести одну разрешенную комбинацию в другую.

Ошибки можно не только обнаруживать, но и исправлять.

Теорема. Для исправления одиночной ошибки каждой разрешенной кодовой комбинации необходимо сопоставить подмножество запрещенных кодовых комбинаций. Чтобы эти подмножества не пересекались, хэммингово расстояние должно быть не менее трех.

Доказательство. Пусть, как и в предыдущем примере, n=3. Примем разрешенные комбинации 000 и 111 (кодовое расстояние между ними равно 3). Разрешенной комбинации 000 поставим в соответствие подмножество запрещенных комбинаций 001, 010, 100. Эти запрещенные комбинации образуются в результате возникновения единичной ошибки в комбинации 000.

Аналогично разрешенной комбинации 111 необходимо поставить в соответствие подмножество запрещенных комбинаций 110, 011, 101. Если сопоставить эти подмножества запрещенных комбинаций, то очевидно, что они не пересекаются:



В общем случае исправляемые ошибки кратности $t_и$ связаны с кодовым расстоянием соотношением

$d_{\min} = 2t_и + 1.$ (2.1)

Для ориентировочного определения необходимой избыточности кода при заданном кодовом расстоянии d можно воспользоваться верхней граничной оценкой для $r = n - k$, называемой оценкой Хэмминга:

$$r = n - k \geq \log_2\left(1 + \sum_{t=1}^{t_и} C_n^t\right),$$

где C_n^t - сочетание из n элементов по t (число возможных ошибок кратности t на длине n-разрядной комбинации).

Если, например, n=7, $t_и$ =1, то из (2.1)

$d_{\min}$=3, n - k $\log_2(1+7)$=3 .

Нужно отметить, что каждый конкретный корректирующий код не гарантирует исправления любой комбинации ошибок. Коды предназначены для исправления комбинаций ошибок, наиболее вероятных для заданного канала связи.

Групповой код с проверкой на четность

Недостатком кода с четным числом единиц является необнаружение четных групповых ошибок. Этого недостатка лишены коды с проверкой на четность, где комбинации разбиваются на части, из них формируется матрица, состоящая из некоторого числа строк и столбцов:

$$\begin{array}{l} x_{11}\ x_{12}\ x_{13}\ \dots x_{1n} \\ x_{21}\ x_{22}\ x_{23}\ \dots x_{2n} \\ \dots\dots\dots\dots\dots\dots \\ x_{m1}\ x_{m2}\ x_{m3}\ \dots x_{mn} \\ \hline x_{k1}\ x_{k2}\ x_{k3}\ \dots x_{kn} \end{array}$$

Строки образуются последовательно по мере поступления символов исходного кода. Затем после формирования m строк матрицы производится проверка на четность ее столбцов и образуются контрольные символы x_{ki} . Контрольные символы образуются путем суммирования по модулю 2 информационных символов, расположенных в столбце:

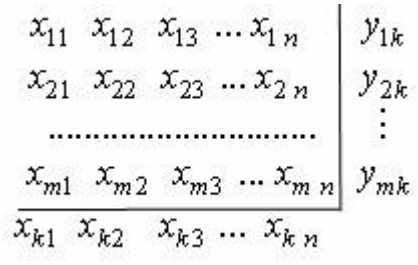
$$x_{ki} = \sum_{j=1}^{m} x_{ji} \bmod 2 \quad .$$

При таком кодировании четные групповые ошибки обнаруживаются. Не обнаруживаются лишь такие ошибки, при которых искажено четное число символов в столбце.

Можно повысить обнаруживающую способность кода путем одновременной проверки на четность по столбцам и строкам или столбцам и диагоналям (поперечная и диагональная проверка).

Если проверка проводится по строкам и столбцам, то код называется ***матричным.***

Проверочные символы располагаются следующим образом:



$$x_{ki} = \sum_{j=1}^{m} x_{ji} \bmod 2 \quad ;$$

$$y_{jk} = \sum_{j=1}^{n} x_{ij} \bmod 2 \quad .$$

В этом случае не обнаруживаются только ошибки четной кратности с кратностью 4, 8, 16 и т.д., при которых происходит искажение символов с попарно одинаковыми индексами строк столбцов. Наименьшая избыточность кода получается в том случае, когда образуемая матрица является квадратной.

Недостатком такого кода является необходимость внесения задержки в передачу информации на время, необходимое для формирования матрицы.

Матричный код позволяет исправлять одиночные ошибки. Ошибочный элемент находится на пересечении строки и столбца, в которых имеется нарушение четности.

Коды с постоянным весом

Весом называется число единиц, содержащихся в кодовых комбинациях.

Если число единиц во всех комбинациях кода будет постоянным, то такой код будет кодом с постоянным весом. Коды с постоянным весом относятся к классу *блочных неразделимых кодов*, поскольку здесь невозможно выделить информационные и проверочные символы. Наибольшее применение получили коды «3 из 7», «3 из 8», хотя возможны другие варианты. Первая цифра указывает на вес кода, вторая - на общее число символов в комбинации.

Разрешенными комбинациями кода «3 из 7» являются такие, которые содержат три единицы независимо от их места в комбинации, например 1110000 или 1010100 и т.д. Обнаружение ошибок сводится к определению их веса. Если вес отличается от заданного, то считается, что произошла ошибка. Код обнаруживает веса ошибок нечетной кратности и части ошибок четной кратности. Не обнаруживаются ошибки, при которых несколько единиц превращается в нули и столько же нулей - в единицы (ошибки смещения), так как при этом вес кода не изменяется.

В коде «3 из 7» возможных комбинаций сто двадцать восемь (2^7=128), а разрешенных кода только тридцать пять. Относительная избыточность $_{отн}$ = 0,28.

Схема устройства определения веса комбинаций кода «3 из 7» приведена на рис. 2.6.

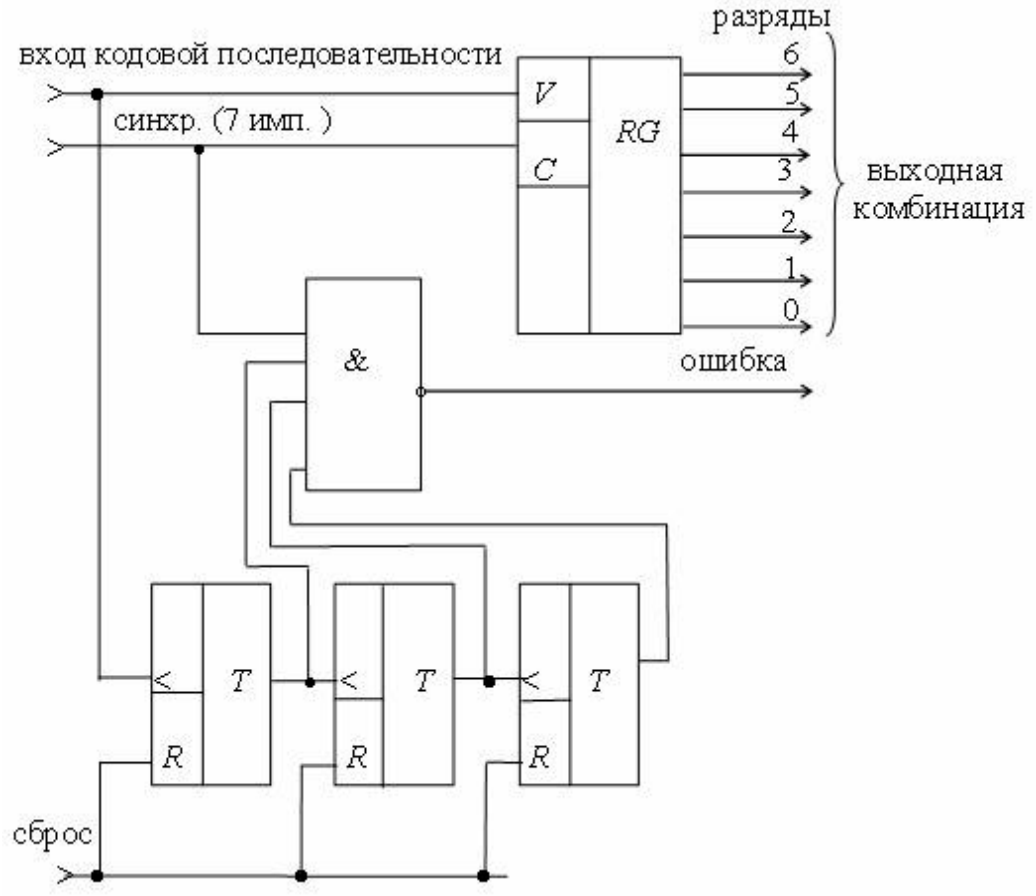


Рис. 2. Схема определения веса комбинаций кода «3 из 7»

Контрольные вопросы:

1. На чем базируется теория помехоустойчивого кодирования?
2. Какие коды называют избыточными или корректирующими?
3. Каковы основные параметры корректирующих кодов?
4. Чемхарактеризуетсякодовоерасстояние?
5. Что является недостатком кода с четным числом единиц?
6. К какому классу относятся коды с постоянным весом?

Лекция №8. Циклические коды.
Коды Хэмминга и Голея

Циклические коды характеризуются тем, что при циклической перестановке всех символов кодовой комбинации данного кода образуется другая кодовая комбинация этого же кода.

$x_n \ x_{n-1} \dots x_2 \ x_1$ - комбинация циклического кода;

$x_{n-1} \ x_{n-2} \dots x_2 \ x_1 \ x_n$ - также комбинация циклического кода.

При рассмотрении циклических кодов двоичные числа представляют в виде многочлена, степень которого (n - 1), n - длина кодовой комбинации.

Например, комбинация 1001111 (n=7) будет представлена многочленом

$$1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0 = x^6 + x^3 + x^2 + x + 1.$$

При таком представлении действия над кодовыми комбинациями сводятся к действиям над многочленами. Эти действия производятся в соответствии с обычной алгебры, за исключением того, что приведение подобных членов осуществляется по модулю 2.

Обнаружение ошибок при помощи циклического кода обеспечивается тем, что в качестве разрешенных комбинаций выбираются такие, которые делятся без остатка на некоторый заранее выбранный полином $G(x)$. Если принятая комбинация содержит искаженные символы, то деление на полином $G(x)$ осуществляется с остатком. При этом формируется сигнал, свидетельствующий об ошибке.*Полином*$G(x)$ *называется образующим*.

Построение комбинаций циклического кода возможно путем умножения исходной комбинации $A(x)$ на образующий полином $G(x)$с приведением подобных членов по модулю 2:

- если старшая степень произведения не превышает (n - 1), то полученный полином будет представлять кодовую комбинацию циклического кода;
- если старшая степень произведения больше или равна n, то полином произведения делится на заранее выбранный полином степени n и результатом умножения считается полученный остаток от деления.

Таким образом, все полиномы, отображающие комбинации циклического кода, будут иметь степень ниже n.

Часто в качестве полинома, на который осуществляется деление, берется полином $G(x)=x^n+1$. При таком формировании кодовых комбинаций позиции информационных и контрольных символов заранее определить нельзя.

Большим преимуществом циклических кодов является простота построения кодирующих и декодирующих устройств, которые по своей структуре представляют регистры сдвига с обратными связями.

Число разрядов регистра выбирается равным степени образующего полинома.

Обратная связь осуществляется с выхода регистра на некоторые разряды через сумматоры, число которых выбирается на единицу меньше количества ненулевых членов образующего полинома. Сумматоры устанавливаются на входах тех разрядов регистра, которым соответствуют ненулевые члены образующего полинома.

На рис. 1 приведена схема кодирующего регистра для преобразования четырехразрядной комбинации в семиразрядную.

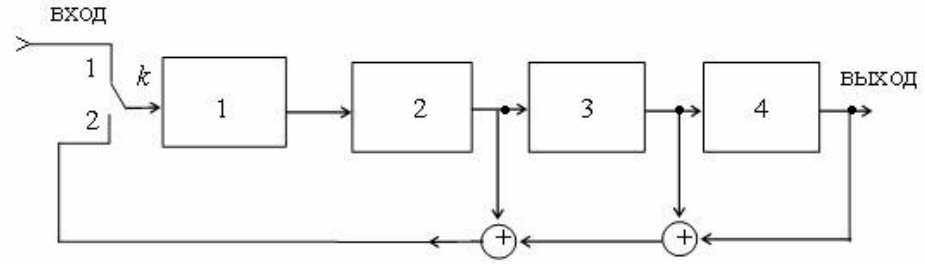


Рис. 1. Схема кодирующего регистра

В таблице показано, как путем сдвигов исходной комбинации 0101 получается комбинация циклического кода 1010011. n=7, k=4. Комбинация 0101, ключ в положении 1. В течение первых четырех тактов регистр будет заполнен, затем ключ переводится в положение 2. Обратная связь замыкается. Под действием семи сдвигающих тактов проходит формирование семиразрядного циклического кода.

Таблица1

Разряды	Выход
0101	–
0010	1
1001	0
1100	1
0110	0
0011	0
0001	1
0000	1

Свойства циклического кода:

1) циклический код обнаруживает все одиночные ошибки, если образующий полином содержит более одного члена. Если $G(x)=x+1$, то код обнаруживает одиночные ошибки и все нечетные;

2) циклический код с $G(x)=(x+1)G(x)$ обнаруживает все одиночные, двойные и тройные ошибки;

3) циклический код с образующим полиномом $G(x)$ степени $r = n - k$ обнаруживает все групповые ошибки длительностью в r символов.

Коды Хэмминга — вероятно, наиболее известный из первых самоконтролирующихся и самокорректирующихся кодов. Построены они применительно к двоичной системе счисления.

Построение кодов Хемминга основано на принципе проверки на четность числа единичных символов: к последовательности добавляется такой элемент, чтобы число единичных символов в получившейся последовательности было четным. $r_1 = i_1 \oplus i_2 \oplus ... \oplus i_k$. знак $\oplus$ здесь означает сложение по модулю 2

$S = i_1 \oplus i_2 \oplus ... \oplus i_n \oplus r_1$. $S = 0$ - ошибки нет, $s = 1$ однократная ошибка. Такой код называется $(k+1, k)$ или $(n, n-1)$. Первое число - количество элементов последовательности, второе - количество информационных символов. Для каждого числа проверочных символов $r = 3, 4, 5..$ существует классический код Хемминга с маркировкой $(n, k) = (2^r - 1, 2^r - 1 - r)$ т.е. - $(7,4), (15,11), (31,26)$. При иных значениях k получается так называемый усеченный код, например международный телеграфный код МТК-2, у которого $k = 5$. Для него необходим код Хемминга $(9,5)$, который является усеченным от классического $(15,11)$. Для Примера рассмотрим классический код Хемминга $(7,4)$. Сгруппируем проверочные символы следующим образом:

$r_1 = i_1 \oplus i_2 \oplus i_3$

$r_2 = i_2 \oplus i_3 \oplus i_4$

$r_3 = i_1 \oplus i_2 \oplus i_4$

знак $\oplus$ здесь означает сложение по модулю 2.

Получение кодового слова выглядит следующим образом:

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} i_1 & i_2 & i_3 & i_4 & r_1 & r_2 & r_3 \end{pmatrix}$$

На вход декодера поступает кодовое слово $V = (i'_1, i'_2, i'_3, i'_4, r'_1, r'_2, r'_3)$ где штрихом помечены символы, которые могут исказиться в результате помехи. В декодере в режиме исправления ошибок строится последовательность синдромов:

$S_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_3$

$S_2 = r_2 \oplus i_2 \oplus i_3 \oplus i_4$

$S_3 = r_3 \oplus i_1 \oplus i_2 \oplus i_4$

$S = (S_1, S_2, S_3)$ называется синдромом последовательности.

Получение синдрома выглядит следующим образом:

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & r_1 & r_2 & r_3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_1 & S_2 & S_3 \end{pmatrix}$$

Кодовыеслова $(7,4)$ кодаХемминга

i_1	i_2	i_3	i_4	r_1	r_2	r_3
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	1	0	0
1	1	1	1	1	1	1

Синдром $(0,0,0)$ указывает на то, что в последовательности нет искажений. Каждому ненулевому синдрому соответствует определенная конфигурация ошибок, которая исправляется на этапе декодирования. Для кода $(7,4)$ в таблице указаны ненулевые синдромы и соответствующие им конфигурации ошибок (для вида: $i_1\, i_2\, i_3\, i_4\, r_1\, r_2\, r_3$).

Синдром	001	010	011	100	101	110	111
Конфигурация ошибок	0000001	0000010	0001000	0000100	1000000	0010000	0100000
Ошибка в символе	r_3	r_2	i_4	r_1	i_1	i_3	i_2

Одним из наиболее практичных блочных кодов является двоичный *расширенный код Голея* , который образован путем прибавления битов четности к совершенному коду, известному как *код Голея* (Golaycode). Эти дополнительные биты повышают минимальное расстояние d_{min} с 7 до 8, что дает степень кодирования 1/2, реализовать которую проще (с точки зрения системного тактового генератора), чем степень кодирования кода Голея, равную 12/23. Расширенный код Голея значительно мощнее рассмотренного в предыдущем разделе кода Хэмминга. Цена, которую приходится платить за повышение эффективности, заключается в более сложном декодере и, соответственно, более широкой полосе пропускания.

Для расширенного кода Голея $d_{min} = 8$, можно сказать, что код гарантирует исправление всех трехбитовых ошибок. Кроме того, декодер можно сконструировать так, чтобы он исправлял *некоторые* комбинации с четырьмя ошибками. Поскольку исправить можно только 16,7% комбинаций с четырьмя ошибками, декодер, для упрощения, обычно реализуется для исправления только трехбитовых ошибочных комбинаций. Если предположить жесткое декодирование, то вероятность битовой ошибки для расширенного кода Голея можно представить как функцию вероятности *p* ошибки в канальном символе

$$P_B = \frac{1}{24}\sum_{j=4}^{24} jC_j^{24} p^j (1-p)^{24-j}$$

Контрольные вопросы:

1. Чемхарактеризуютсяциклическиекоды?
2. Какие свойства у циклического кода?
3. Какой код является одним из наиболее практичных блочных кодов?
4. Какойсигналназываетсяобразующим?
5. Какосуществляетсяобратнаясвязь?

Лекция №9. Сверточные коды

Сверточные коды относятся к непрерывным рекуррентным кодам. Кодовое слово является сверткой отклика линейной системы (кодера) на входную информационную последовательность. Поэтому сверточные коды являются

линейными, для которых сумма любых кодовых слов также является кодовой последовательностью.

Сверточные коды имеют большой научный и практический интерес для современных систем и сетей телекоммуникаций. Это определяется многими их достоинствами, а именно:

- высокой скоростью обработки информации (десятки и сотни Мбит/с),
- высокой корректирующей способностью как случайных, так пакетных ошибок,
- реализациейэффективныхкодеков,
- эффективным применением в каналах связи с фазовой неопределенностью и др.

В общем виде кодирование информации сверточными кодами может быть представлено следующим образом:

$$T^{(i)}(x)=\sum_{j=1}^{k_0} I^{(j)}(x)\cdot g^{(j)}(x),\quad j=1,2,\ldots k_0,\ i=j+1, \qquad (1)$$

где *I(x)* – последовательность передаваемых информационных символов; *x* – формальная переменная; *g(x)* – порождающий, или образующий, полином (многочлен); k_0 – блок информационных символов, одновременно поступающих на вход кодирующего устройства $(k_0 \geq 1)$.

Способ формирования кодовых символов, выполняемых согласно (1), соответствует форме записи свертки двух функций, что и послужило названию данных кодов. *Сверточный код* – это рекуррентный код (т.е. операции выполняются шаг за шагом) с периодической полубесконечной структурой символов кодовой последовательности. Обобщенная структурная схема кодера СК представлена на рис. 1.

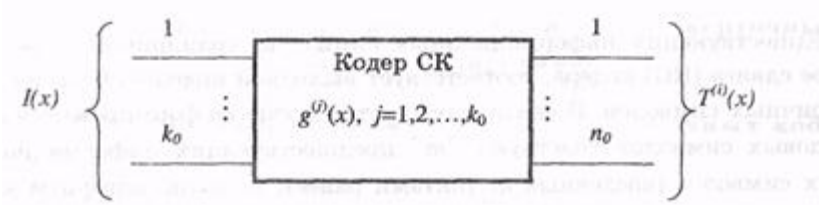


Рис. 1. Обобщенная структура кодера СК

Входные информационные символы *I(x)* делятся на k_0 символов, которые одновременно с каждым тактом поступают на входы кодера сверточного кода, в котором согласно (1) формируются кодовые символы n_0. Таким образом, кодовая последовательность $T^{(i)}(x)$ представляет собой полубесконечную последовательность блоков n_0.

Существенное отличие сверточных кодов от линейных блоковых кодов: для линейных блоковых кодов проверочные символы зависят от одного информационного блока, а для сверточных кодов проверочные символы зависят как от информационных символов на входе, так и от некоторого количества предшествующих.

Любому входному информационному блоку из k_0 информационных символов и m предшествующих символов, хранящихся в регистре сдвига кодера, соответствует выходной кодовый миниблок из n_0 двоичных символов. Т.к. в алгоритме кодирования участвуют предшествующие символы m, то такой алгоритм называется кодирование с памятью**.**

К основным параметрам сверточных кодов относятся:

1. $R = k_0/n_0 = 1/2; 2/3; 3/4$ и т.д. – скорость передачи кода, которая для сверточных кодов записывается в виде дроби;
2. $l=n_0-k_0$ – абсолютная избыточность;
3. $r=(n_0-k_0)/n_0\times100\%=(1-R)\times100\%$ – относительная избыточность;
4. $J\geq2$ – количество ортогональных проверочных уравнений, т.е. количество ненулевых членов в образующем полиноме;
5. $d_0=J+1$ – минимальное кодовое расстояние;
6. $t_{исп} \leq \frac{j}{2}$ – кратность или количество исправляемых ошибок;
7. $t_{обн} \leq d_0 - 1 = j$ – кратностьобнаруживаемыхошибок;
8. $n_A=(m+1)n_0$ – длина кодового ограничения, или длина кодовой последовательности, соответствующая кодированию информационных блоков из k_0 символов в течение $(m+1)$ такта; m – старшая степень ненулевого многочлена порождающего полинома;
9. $k_A=R\times n_A$ – количество информационных символов, приходящихся на n_A кодовых символов;
10. $n_E=J^2/2+J/2+1$ –эффективная длина кодового ограничения (количество двоичных символов, непосредственно участвующих в декодировании).

Классификация сверточных кодов.

- *По основанию кода*: двоичные и недвоичные;
- *В зависимости от используемого математического аппарата*: алгебраические и неалгебраические;
- *По алгоритму формирования проверочных символов*: линейные и нелинейные;
- *По способу передачи*: систематические и несистематические;
- *По структуре кодовой последовательности*: разделимые и неразделимые;

- *По алгоритму декодирования:* ортогональные и неортогональные;
- *По способу преобразования входных информационных символов* k_0 *в кодовые символы* СК являются непрерывными.

В зависимости от способа формирования проверочных уравнений СК бывают *ортогональными, самоортогональными и ортогонализируемыми.*

1. *Ортогональными* СК (ОСК) называют такие коды, в которых система из J $(J \geq 2)$ проверочных уравнений ортогональна относительно декодируемых k_0 информационных символов и неортогональна относительно информационных символов, входящих в данные проверочные уравнения.
2. *Самоортогональные* СК (ССК) – коды, в которых декодируемый информационный символ входит одновременно во все проверочные уравнения, а все остальные символы, участвующие в декодировании в данный момент времени, входят не более, чем в одно проверочное уравнение, т.е. СК формирует так называемую, систему раздельных проверок.
3. *Ортогонализируемыми* СК называются такие коды, у которых при декодировании информационного или k_0 символов требуется выполнить дополнительные линейные преобразования над проверочными символами для получения дополнительных, так называемых составных проверок.

Ограничимся ниже рассмотрением лишь наиболее характерных (базовых, или материнских) для мобильной связи сверточных кодов со скоростями вида Rk = 1/n0, где n0 - некоторое натуральное число. Последовательность символов такого сверточного кода состоит из элементарных блоков длиной n0, причем n0 символов текущего блока (занимающие реальное время, отвечающее одному информационному биту) являются линейной комбинацией текущего информационного бита и т предшествующих. Значение т определяет память кода, а параметр m + 1 называется длиной кодового ограничения. Если один (например, первый) из n0 символов текущего блока повторяет текущий информационный бит, код называется систематическим.

Способы задания сверточных кодов во многом совпадают с используемыми для линейных блоковых. Одним из основных является описание сверточного кода набором n0 порождающих многочленов. Каждый многочлен устанавливает закон формирования одного из n0 символов в группе и имеет степень, не превышающую т. Ненулевые коэффициенты порождающего полинома прямо указывают, какие из информационных символов (включая текущий и т предыдущих) входят в линейную

комбинацию, дающую данный символ кода (см. пример 1). Порождающие многочлены хороших сверточных кодов найдены перебором и табулированы.

Весьма важным с точки зрения понимания алгоритмов кодирования и декодирования инструментом описания сверточных кодов является кодовая решетка, смысл которой должен быть ясен из следующего примера.

Пример 1. Пусть несистематический сверточный код со скоростью Rk =1/2 и кодовым ограничением m + 1 = 3 задается порождающими многочленамиg1(x) = x2+x + 1 и g2(x) = x2+1.

Это означает, что первый из двух символов каждого двухсимвольного блока является линейной комбинацией (суммой по мо-дулю 2) текущего и двух предшествующих информационных би-тов, тогда как второй получается сложением по модулю 2 текущего информационного бита с тем, который поступил от источника двумя тактами раньше.

Кодовая решетка этого кода показана на рис. 1. При ее составлении учтено, что кодер содержит память в виде двухразрядного сдвигающего регистра. Каждому из четырех возможных состояний этого регистра отвечает один из четырех узлов решетки. Поэтому левый символ в обозначении узла равен последнему информационному биту, уже записанному в регистр. При записи в регистр очередного информационного символа регистр меняет состояние на одно из двух соседних. Этот переход обозначен ребрами решетки. Порядок узлов выбран таким, что при нулевом текущем информационном символе (a,=0) переход в следующее состояние соответствует верхнему ребру, а при ai = 1 - нижнему. Маркировка ребер воспроизводит n0-блок, посылаемый

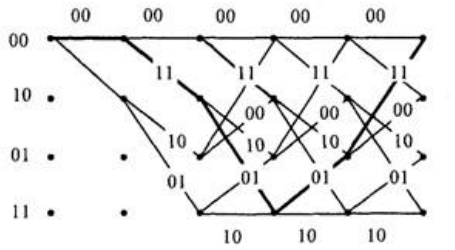


Рис.1. Кодовая решетка

в канал. Каждой информационной последовательности соответствует определенный путь на кодовой решетке и кодовая после-довательность, считываемая как метки, маркирующие последова-тельные ребра пути. К примеру, входным информационным би-там 01100 отвечает кодовое слово 00 11 01 01 11, которому соот-ветствует на рис. 3 путь, отмеченный жирной линией.

Известен ряд алгоритмов декодирования сверточных кодов. В практических системах и, в частности в мобильной связи, как правило, используется алгоритм Витерби, отличающийся простотой реализации при умеренных длинах кодового ограничения.

Контрольные вопросы:

1. К каким кодам относятся сверточные коды?
2. Способызаданиясверточныхкодов?
3. Какой алгоритм используется для декодирования сверточных кодов?
4. Составьте кодовую решетку для сверточного кода

Лекция №10. Двоичные коды БЧХ

Коды Боуза-Чоудхури-Хоквингема (БЧХ) составляют один из больших классов линейных кодов, исправляющих ошибки. Причем метод построения этих кодов задан явно.

Код БЧХ длины n, исправляющий $q_{\text{ис}}$-кратные ошибки, это циклический блочный код над полем $GF(p)$.

В соответствии с этим определением порождающий многочлен кода БЧХ может быть представлен наименьшим общим кратным

$$g(x) = НОК\left[M_v(x), M_{v+1}(x), \ldots, M_{v+2q_{\text{ис}}-1}(x)\right],$$

где $M_j(x)$ – минимальные многочлены элементов β^j.

Доказано, что наличие $2q_{\text{ис}}$ корней полинома $g(x)$, указанных в определении кода, гарантирует исправление всех ошибок кратности, меньшей или равной $q_{\text{ис}}$.

Основное внимание обратим на коды БЧХ, имеющие длину $n = p^m - 1$. Такие коды называются примитивными кодами БЧХ.

Часто выбирают $v = 1$ (случай кодов БЧХ в узком смысле), что, как правило, приводит к порождающему полиному наименьшей степени, а значит, и к наименьшему числу избыточных символов в кодовом слове. Кроме того, целесообразно выбрать $\beta = \alpha$ (α – примитивный элемент поля $GF(p^m)$, поскольку при этом получается наибольшая длина кодового слова. Список порождающих многочленов кодов БЧХ различных длин (вплоть до $n = 256$) имеется, например, в [26].

Построенные таким образом коды БЧХ, исправляющие как минимум $q_{ис}$-кратные ошибки, характеризуются конструктивным расстоянием кода $\delta = 2q_{ис}+1$. Истинное минимальное расстояние d_0 кода БЧХ может оказаться больше, чем δ. Это означает, что ряд кодов БЧХ может исправлять ошибки кратности большей, чем та, которую задают при построении этого кода.

Найдем проверочную матрицу двоичного циклического кода БЧХ, исправляющего $q_{ис}$-кратные ошибки. Учитывая свойство равенства минимальных многочленов с номерами j и $2j$, степень порождающего многочлена $g(x)$ может быть снижена. Действительно, если, например, $\nu = 1$, порождающий многочлен примет вид

$$g(x) = НОК\left[M_1(x), M_3(x), \ldots, M_{2q_{ис}-1}(x)\right]. \quad (1)$$

При этом проверочная матрица

$$H = \begin{vmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{(n-1)} \\ 1 & \alpha^3 & \alpha^6 & \ldots & \alpha^{(n-1)3} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & \alpha^{(2q_{ис}-1)} & \alpha^{2(2q_{ис}-1)} & \ldots & \alpha^{(n-1)(2q_{ис}-1)} \end{vmatrix}. \quad (2)$$

Сравнивая эту матрицу с матрицей кода Хэмминга, видим, что код Хэмминга представляет собой частный случай примитивного кода БЧХ, исправляющего однократные ошибки $(q_{ис}+1)$.

Представляет интерес воспользоваться возможностью описания кодов БЧХ в спектральной области (см. 5.4.5). Как следует из свойств дискретного преобразования Фурье, спектры слов циклического кода БЧХ должны содержать нулевые компоненты с номерами $j = \nu, \nu+1, \ldots, \nu+2q_{ис}-1$.

Таким образом, циклический код БЧХ, исправляющий $q_{ис}$-кратные ошибки, можно определить как множество всех слов над полем $GF(p)$, для которых $2q_{ис}$ последовательных компонентов спектра равна нулю. Указанное свойствокодов БЧХ используется при их декодировании.

К особенностям кодов БЧХ можно отнести тот факт, что с ростом длины n кода при фиксированном значении скорости кода k/n отношение δ/n стремится к нулю. В результате, несмотря на наличие у кодов БЧХ отмеченных положительных свойств, при больших длинах $(n > 1000)$ приходится отдавать предпочтение другим кодам [3, 30].

Контрольные вопросы:

1. К каком классу относятся коды Боуза-Чоудхури-Хоквингема (БЧХ)
2. Какие коды называются примитивными кодами БЧХ?
3. Найдите проверочную матрицу двоичного циклического кода БЧХ?
4. К особенностям кодов БЧХ относят…

Лекция №11. Недвоичные БЧХ коды – коды Рида-Соломона

Коды Рида — Соломона (англ. *Reed–Solomoncodes*)- недвоичные циклические коды, позволяющие исправлять ошибки в блоках данных. Элементами кодового вектора являются не биты, а группы битов (блоки). Очень распространены коды Рида — Соломона, работающие с байтами (октетами).

Код Рида — Соломона является частным случаем БЧХ-кода.

В настоящее время широко используется в системах восстановления данных с компакт-дисков, при создании архивов с информацией для восстановления в случае повреждений, в помехоустойчивом кодировании.

Коды Рида — Соломона являются важным частным случаем БЧХ-кода, корни порождающего полинома которого лежат в том же поле, над которым строится код ($m = 1$). Пусть α — элемент поля $GF(q)$ порядка n. Если α — *примитивный* элемент, то его порядок равен $q - 1$, то есть $\alpha^{q-1} = 1, \quad \alpha^i \neq 1, 0 < i < q - 1$. Тогда нормированный полином $g(x)$ минимальной степени над полем $GF(q)$, корнямикоторого являются $d - 1$ подряд идущих степеней $\alpha^{l_0}, \alpha^{l_0+1}, ..., \alpha^{l_0+d-2}$ элемента α, является порождающим полиномом кода Рида — Соломона над полем $GF(q)$:

$g(x) = (x - \alpha^{l_0})(x - \alpha^{l_0+1}) \dots (x - \alpha^{l_0+d-2})$

где l_0 — некоторое целое число (в том числе 0 и 1), с помощью которого иногда удается упростить кодер. Обычно полагается $l_0 = 1$. Степень многочлена $g(x)$ равна $d - 1$.

Длина полученного кода n, минимальное расстояние d (минимальное расстояние d линейного кода является минимальным из всех расстояний Хемминга всех пар кодовых слов, см. Линейный код). Код содержит $r = d - 1 = \deg(g(x))$ проверочных символов, где $\deg()$ обозначает степень полинома; число информационных символов $k = n - r = n - d + 1$. Таким образом $d = n - k + 1$ и код Рида — Соломона является *разделимым кодом с максимальным расстоянием* (является оптимальным в смысле границы Синглтона).

Кодовый полином $c(x)$ может быть получен из информационного полинома $m(x)$, $\deg m(x) \leqslant k - 1$, путем перемножения $m(x)$ и $g(x)$:

$c(x) = m(x)g(x)$

Свойства

Код Рида — Соломона над $GF(q^m)$, исправляющий t ошибок, требует $2t$ проверочных символов и с его помощью исправляются произвольные пакеты ошибок длиной t и меньше. Согласно теореме о границе Рейгера, коды Рида — Соломона являются оптимальными с точки зрения соотношения длины пакета и возможности исправления ошибок — используя $2t$ дополнительных проверочных символов исправляется tошибок (и менее).

Теорема (граница Рейгера). Каждый линейный блоковый код, исправляющий все пакеты длиной t и менее, должен содержать, по меньшей мере, $2t$ проверочных символов.

Код, двойственный коду Рида — Соломона, есть также код Рида-Соломона. Двойственным кодом для циклического кода называется код, порожденный его проверочным многочленом.

Матрица $G = [\ I_{k*k} \quad P_{k*(n-k)}\ ]$ порождает код Рида — Соломона тогда и только тогда когда любой минор матрицы $P_{k*(n-k)}$ отличен от нуля.

При выкалывании или укорочении кода Рида-Соломона снова получается код Рида — Соломона. Выкалывание — операция, состоящая в удалении одного проверочного символа. Длина n кода уменьшается на единицу, размерность k сохраняется. Расстояние кода d должно уменьшиться на единицу, ибо в противном случае удаленный символ был бы бесполезен.Укорочение - фиксируем произвольный столбец (n, k, d)кода и выбираем только те векторы, которые в данном столбце содержат 0. Это множество векторов образует подпространство.

Исправление многократных ошибок

Код Рида — Соломона является одним из наиболее мощных кодов, исправляющих многократные пакеты ошибок. Применяется в каналах, где пакеты ошибок могут образовываться столь часто, что их уже нельзя исправлять с помощью кодов, исправляющих одиночные ошибки.

$(q^m - 1, q^m - 2 - 2t)$-код Рида — Соломона над полем $GF(q^m)$ с кодовым расстоянием $d = 2t + 1$ можно рассматривать как $((q^m - 1)m, (q^m - 1 - 2t)m)$-код над полем $GF(q)$, который может исправлять любую комбинацию ошибок, сосредоточенную в t или меньшем числе блоков из m символов. Наибольшее число блоков длины m, которые может затронуть пакет длины l_i, где $l_i \leqslant mt_i - (m-1)$, не превосходит t_i, поэтому код, который может исправить t блоков ошибок, всегда может исправить и любую комбинацию из p пакетов общей длины l, если $l + (m-1) \leqslant mt$.

Практическая реализация

Кодирование с помощью кода Рида — Соломона может быть реализовано двумя способами: систематическим и несистематическим.
При несистематическом кодировании информационное слово умножается на некий неприводимый полином в поле Галуа. Полученное закодированное слово полностью отличается от исходного и для извлечения информационного слова нужно выполнить операцию декодирования и уже потом можно проверить данные на содержание ошибок. Такое кодирование требует большие затраты ресурсов только на извлечение информационных данных, при этом они могут быть без ошибок.
Структура систематического кодового слова Рида — Соломона
При систематическом кодировании к информационному блоку из k символов приписываются $2t$ проверочных символов, при вычислении каждого проверочного символа используются все k символов исходного блока. В этом случае нет затрат ресурсов при извлечении исходного блока, если информационное слово не содержит ошибок, но кодировщик/декодировщик должен выполнить $k(n-k)$ операций сложения и умножения для генерации проверочных символов. Кроме того, так как все операции проводятся в поле Галуа, то сами операции кодирования/декодирования требуют много ресурсов и времени. Быстрый алгоритм декодирования, основанный на быстром преобразовании Фурье, выполняется за время порядка $O(ln(n)^2)$.

Кодирование

При операции кодирования информационный полином умножается на порождающий многочлен. Умножение исходного слова S длины k на неприводимый полином при систематическом кодировании можно выполнить следующим образом:
К исходному слову приписываются $2t$ нулей, получается полином $T = Sx^{2t}$.

Этот полином делится на порождающий полином G, находится остаток R, $Sx^{2t} = QG + R$, где Q — частное.
Этот остаток и будет корректирующим кодом Рида — Соломона, он приписывается к исходному блоку символов. Полученное кодовое слово $C = Sx^{2t} + R$.
Кодировщик строится из сдвиговых регистров, сумматоров и умножителей. Сдвиговый регистр состоит из ячеек памяти, в каждой из которых находится один элемент поля Галуа.
Существует и другая процедура кодирования (более практичная и простая). Положим $a_i \in GF(q), (i = 1, 2, \ldots, k-1), \alpha \in GF(q)$ - примитивный элемент поля $GF(q)$, и пусть $a = (a_0, a_1, \ldots, a_{k-1})$ - вектор информационных символов , а значит $a(x) = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1}$ - информационный многочлен. Тогда вектор $u = (a(1), a(\alpha), \ldots, a(\alpha^{q-2}))$ есть вектор кода Рида - Соломона , соответствующий информационному вектору a. Этот способ кодирования показывает , что для кода РС вообще не нужно знать порождающего многочлена и порождающей матрицы коды, достаточно знать разложение поля $GF(q)$ по примитивному элементу α и размерность кода k (длина кода в этом случае определяется как $n = q - 1$). Все дело в том , что за разностью $n - k$ полностью скрывается порождающий многочлен $g(x)$ и кодовое расстояние.

Декодирование

Декодировщик, работающий по авторегрессивному спектральному методу декодирования, последовательно выполняет следующие действия:

- Вычисляетсиндромошибки
- Строитполиномошибки
- Находиткорниданногополинома
- Определяетхарактерошибки
- Исправляетошибки
- Вычислениесиндромаошибки

Вычисление синдрома ошибки выполняется синдромным декодером, который делит кодовое слово на порождающий многочлен. Если при делении возникает остаток, то в слове есть ошибка. Остаток от деления является синдромом ошибки.
Построение полинома ошибки - Вычисленный синдром ошибки не указывает на положение ошибок. Степень полинома синдрома равна $2t$, что

много меньше степени кодового слова n. Для получения соответствия между ошибкой и ее положением в сообщении строится полином ошибок. Полином ошибок реализуется с помощью алгоритма Берлекэмпа — Месси, либо с помощью алгоритма Евклида. Алгоритм Евклида имеет простую реализацию, но требует больших затрат ресурсов. Поэтому чаще применяется более сложный, но менее затратоемкий алгоритм Берлекэмпа — Месси. Коэффициенты найденного полинома непосредственно соответствуют коэффициентам ошибочных символов в кодовом слове.

*Нахождение корней*На этом этапе ищутся корни полинома ошибки, определяющие положение искаженных символов в кодовом слове. Реализуется с помощью процедуры Ченя, равносильной полному перебору. В полином ошибок последовательно подставляются все возможные значения, когда полином обращается в ноль — корни найдены.

*Определение характера ошибки и ее исправление*По синдрому ошибки и найденным корням полинома с помощью алгоритма Форни определяется характер ошибки и строится маска искаженных символов. Однако для кодов РС существует более простой способ отыскания характера ошибок. Как показано в [2] для кодов РС с произвольным множеством $2t_d$ последовательных нулей $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta}, \delta = 2t_d - 1$

$$e_{j_l} = \frac{(\alpha^{j_l})^{2-b}\Lambda(\alpha^{-j_l})}{\sigma'(\alpha^{-j_l})} \qquad (*)$$

где $\sigma'(x)$ формальная производная по x многочлена локаторов ошибок $\sigma(x)$,а $\Lambda(x) = \sigma(x)S(x) \mod x^{2t_d+1}$

Далее после того как маска найдена , она накладывается на кодовое слово с помощью операции XOR и искаженные символы восстанавливаются. После этого отбрасываются проверочные символы и получается восстановленное информационное слово.

Контрольные вопросы:

1. КодыРида — Соломона – это…
2. Какими двумя способами может быть реализовано кодирование с помощью кода Рида — Соломона?
3. Опишите процесс кодирования кодами Рида – Соломона
4. Опишите процесс декодирования кодами Рида – Соломона
5. Что называют двойственным кодом для циклического кода?

Лекция №12. Коды Файра

Принцип построения кода Файра

Экспериментальные исследования каналов связи показали, что ошибки символов при передаче по каналу связи, как правило, группируются в пакеты различной длительности. Под пакетом ошибок длиной L понимают такой вид комбинации ошибок, в котором между соседними разрядами, пораженными ошибками содержится b < L - 2 разряда.

Пакеты ошибок возникают в результате воздействия на канал передачи помех импульсного характера, длительность которых больше длительности одного символа. При этих условиях ошибки независимы, они возникают пакетами, общая длительность которых соответствует длительности помех. Существуют специально сконструированные коды для обнаружения и исправления пакетов ошибок. Код Файра есть наиболее известный циклический код, исправляющий одиночные пакеты ошибок, причем для этого требуется небольшое число проверочных символов.

Образующий полином кода Файра определяется как:

$P(x)=g(x)(x^c+1)$,

где g(x) - неприводимый многочлен степени *t*, принадлежащий степени *m*, причем c не кратно *m*.

Многочлен g(x) называется неприводимым, если его нельзя разложить на множители. Однако, этот многочлен всегда можно разложить на множители, используя элементы из некоторого расширения.

Порядком *m* элемента β конечного поля называется наименьшее значение *m*, для которого $\beta^m = 1$, β является корнем многочлена $x^m - 1$. Если β является также корнем некоторого неприводимого многочлена y(x), то g(x) должен быть делителем $x^m - 1$, c - простое число, которое не делится на *m* без остатка. Наименьшее значение *m*, для которого произвольный многочлен g(x) без кратных корней делит $x^m - 1$совпадает с наименьшим общим кратным порядков корней g(x). Поэтому *m* является длиной самого короткого цикла, порожденного регистром с обратными связями, определяемыми многочленом g(x).

Для любого *t* существует, по крайней мере, один неприводимый многочлен g(x) степени *t*, принадлежащий показателю степени $m = 2^t - 1$. Например: если $g(x)=x^3+x^2+1$ (t=3) то $m= 2^t - 1 =7$ и число может принимать значения, которые делятся на 7, то есть 15, 16, 17, 18, 19, 20, 22 и т.д.

Длина кода Файра равна наименьшему общему кратному чисел *c* и *m*

n = НОК (c, m) т, е. такому, что g(x) делит x^m-1 .
Число проверочных символов $r = c + t$.
Число информационных символов $k = n - c - t$.
Длина "b" исправляемого пакета ошибок удовлетворяет неравенству:
c≥b + d-1, t≥b
Код Файра может быть использован в режиме одновременного исправления пакета ошибок длины ≤b, и обнаружения пакета ошибок длины, d≥b причем должны выполняться неравенства:
c≥b + d - 1, t≥b
Наличие сомножителя x^c +1 в многочлене P(x) достаточно для обнаружения одиночной пачки ошибок длины "c" или меньше и для полного определения значений ошибок в пачках длины не превосходящей "b".
Дополнительная информация, требуемая для определения положения пачки ошибок, обеспечивается сомножителем g (x).
Пример: Пусть К=63, b=3, d=9, т.е. требуется построить код, исправляющий пакет ошибок длиной в 3 и меньше разрядов и одновременно обнаруживающий пакет ошибок длиной 9 и меньше разрядов Тогда t≥b = 3, c≥b + d-1 =3 + 9-1 =11. Выбираем неприводимый многочлен третьей степени, $g(x)=x^3+x^2+1$. Порядок корней такого многочлена равен 7. Числа, соответствующие c и m, взаимно простые.
Следовательно: n = НОК (c, m) = НОК(11, 7) = 77. Образующий многочлен заданного кода имеет вид: $P(x)=(x^3+x^2 +1)* (x^{11} + 1)$ Таким образом, при заданных корректирующих возможностях (b=3; d=9) код Файра имеет n=77, k=63, $c + t$ = 14

Принцип кодирования и декодирования кода Файра

Принципы кодирования кода Файра

Так как коды Файра относятся к классу циклических кодов, то они обладают всеми свойствами последних. Построение кода Файра ведется по тем же правилам, что и построение любого циклического кода. Как известно, кодовая комбинация циклического кода может быть получена двумя способами:
1) Умножением k-элементной комбинации простого кода на образующий полином P(x).
2) Умножением кодовой комбинации простого кода на одночлен xr и добавлением к этому произведению остатка от деления произведения G(x) x^r на P(x).

Циклический код, как и всякий систематический код, однозначно определяется подобранными определенным образом исходными кодовыми комбинациями. Эти комбинации записываются в виде производящей матрицы из *k* строк и *n* столбцов. Для формирования строк производящей матрицы по второму способу образования циклического кода берут не произвольные комбинации неизбыточного кода G(x), а лишь те из них, которые содержат 1 в одном разряде. Именно эти комбинации умножаются на xrи находится остаток от деления $G(x)\, x^r / P(x)$, равный Ri(x)

Соответствующая строка матрицы записывается в виде $Gi(x)x^r+Ri(x)$. При этом вся матрица разбивается на две подматрицы $G_{n,k}=| E^t_k\ T, C_{r,k}|$, где E^t_k - единичная транспонированная матрица; $C_{r,k}$- подматрица с числом столбцов *r* и строк *k*, образованная остатками от деления Ri(x)

Производящая матрица дает возможность получить первые к комбинаций кода. Остальные 2^k- 1-комбинаций получаются суммированием по модулю 2 строк производящей матрицы во всех возможных сочетаниях. Последняя комбинация кода является нулевой. Рассмотрим построение кода Файра с параметрами *n*=9; *k*=4. Для построения данного кода выберем образующий полином $P(x) =(x^2 + x + 1)(x^3 + 1) = x^5 + x^4 + x^3 + x^2 + 1$ Для этого кода *t*=2; *m*= 2^t -1 = 22 -1 = 3, c=3; r = c + t = 3 + 2 = 5

По второму способу построения построим производящую матрицу

$$G_{9,4} = \left|E_k^t, G_{r,k}\right| = \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{vmatrix}$$

Все остальные комбинации получаются путем суммирования по модулю два - строк производящей матрицы во всех возможных сочетаниях.

Принципы декодирования кода Файра

Код Файра позволяет исправить любой одиночный пакет ошибок длиной b и менее и одновременно обнаружить любой пакет ошибок длиной t≥b, появившиеся в пределах *n* - элементной кодовой комбинации. Известны два метода исправления ошибок кодом Файра. Рассмотрим каждый из них. Пусть на передаваемый кодовый вектор {f (x)} воздействует пакет ошибок { $x^iB(x)$}, здесь {} обозначают смежный класс, а член в скобках представитель смежного класса.

B(x) представляет собой многочлен описывающий пакет ошибок, а множитель x^i показывает разряд с которого начинается пакет. Если

принимаемый вектор принадлежит коду, то при делении на образующий полином остаток будет равен 0. Если же остаток не равен 0, то вектор содержит информацию об ошибках:

$\{x^iB(x) = g(x)S(x) + R(x)\}$, где R(x)- многочлен остатка степени меньшей *n-k*. Задача исправления ошибок состоит в том, чтобы по виду остатка R(x) найти

$x^iB(x)$. Тогда исправление сведется к суммированию по модулю 2 принятого вектора с $x^iB(x)$. Можно показать, что если $x^iB(x)$ принадлежит *k*-классу исправляемых пакетов ошибок, то $\{x^iB(x)\}=\{x^{n-i} R_1(x)\}$, где$\{R_1(x)\} =\{x^j R(x)\}$ и i = n-j

Следовательно, в качестве вектора ошибок берется остаток $R_1(x)$ и считается, что эта комбинация расположена в принимаемом кодовом слове, начиная с (*n*- i) - го разряда.

Сформируем алгоритм декодирования. Он состоит из следующих этапов:

1) Принятая комбинация делится на P(x); Если деление будет произведено без остатка, то ошибки в принимаемой комбинации отсутствуют или не обнаруживаются. Если же имеется остаток R(x), то переходят ко второму этапу:

2) Остаток от деления R(x) умножается на x^i(i= 1,2,3,..., *n*) и делится на P(x).

3) Проверяется будет ли остаток $R_1(x)$ исправимой комбинацией. Если степень искажений принятой *n* - разрядной кодовой комбинации такова, что она может быть исправлена, то полученный в результате остаток R(x) степени, меньшей или равной *r*, будет расположен в конце регистра-делителя, а в остальных "с" разрядах регистра делителя будут 0. Операция деления $x^iR(x)$ на P(x) продолжается до такого значения i, при котором $R_1(n)$ будет исправимой комбинацией, но не более чем *n* раз. Отсутствие признака исправляемости комбинации указывает на то, что степень искажения принятой комбинации превышает исправляющую возможность кода.

4) Если после i шагов остаток $R_1(x)$ окажется исправимой комбинацией, то для исправления ошибок необходимо принятую комбинацию сложить по модулю 2 с полученным остатком, начиная с j= (*n* - i) разряда.

Существует еще один способ исправления ошибок кодом Файра. Пусть в канале связи имеет место одиночный пакет ошибок, непревосходящий b. При этом на выходе канала связи будет получен полином: $F(x)=f(x)+x^IB(x)$

где f(x)- кодовый полином, кратный P(x);

B(x) - полином в степени, соответствующей пакету ошибок;

I- номер разряда кодового слова, определяющий начало пакета ошибок.

Декодирование разбивается на следующие этапы:

1) Производится вычисление остатков от деления принятого полинома F(x) на полиномы g(x) и (x^c+1) при этом получаются остатки $B_1(x)$ и $B_2(x)$, соответственно.
2) Производится умножение $B_1(x)$ и $B_2(x)$ на последовательные степени x и вычисление на каждом таком шаге остатков от деления результатов на полиномы g(x) и (x^c+1), соответственно. При этом каждый раз производится сравнение новых остатков.
3) Этап второй осуществляется до тех пор, пока вышеназванные остатки не совпадут. Полученный в результате совпадения остаток определяет вид пакета ошибок. Для нахождения местоположения пакета ошибок (I) может быть использован аппарат теории сравнений. Пусть i и j - показатели степеней x, при которых соответствующие остатки совпадают, тогда
$I = -(iD_1 + jD_2) \bmod n$
Если два числа a и b дают один и то же остаток при делении на число *n*, то говорят что a и b сравнены по модулю *n* и обозначают $a = b \bmod n$ где:
$D1 = 1 \bmod m$; $D1 = 0 \bmod c$;
$D2 = 1 \bmod c$; $D2 = 0 \bmod m$;

Контрольные вопросы:

1. КодыФайра – это…
2. ОбъяснитепринципкодированиякодаФайра
3. ОбъяснитепринципдекодированиякодаФайра
4. Чему равна длина кода Файра?
5. Как определяется образующий полином кода Файра?

Лекция №13. Каскадные коды.

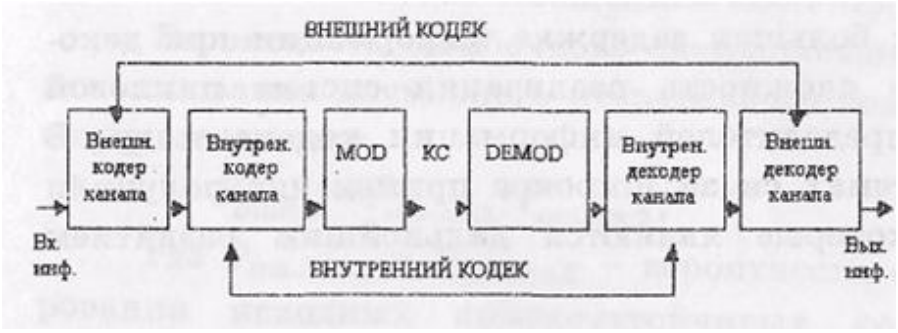
Модификация и комбинирование кодов

Реальные каналы связи, особенно каналы связи на основе стандартных каналов тональной частоты, являются каналами связи с группированием ошибок, причем длина пакетов ошибок может находиться в пределах от десятков двоичных символов до нескольких сотен двоичных символов. Кроме того, в защитных интервалах между пакетами ошибок имеются случайные ошибки. Для коррекции ошибок такой структуры требуются мощные помехоустойчивые коды, а это значит, необходимо использовать помехоустойчивые коды с очень большой длиной кодовых последовательностей и с высокой, избыточностью. Такие коды обладают

высокой сложностью реализации и большой задержкой информации при декодировании.

Для коррекции ошибок данной структуры был разработан способ кодирования информации, который обеспечивает требуемую верность передачи информации при меньшей сложности реализации кодека и задержки информации при декодировании. Сущность данного способа кодирования информации состоит в каскадировании двух или более кодов, т.е. в использовании нескольких уровней или ступеней кодирования и декодирования информации. При этом на каждом уровне кодирования могут использоваться либо одинаковые по типу и корректирующей способности коды, либо разные.

Наиболее распространенной схемой построения каскадных кодов является двухкаскадная или двухступенчатая схема:



В качестве внешнего кода чаще всего используются коды Рида-Соломона, корректирующие пакетные ошибки, а внутренним кодом могут быть различные циклические и сверточные коды, корректирующие случайные ошибки. В реальных системах связи в качестве внутреннего кода используются сверточные коды с алгоритмом декодирования Витерби.

Кодирование и декодирование информации производится следующим образом. Во внешнем кодере передаваемая информация кодируется кодом, рассчитанным для коррекции пакетных ошибок. Чаще всего используются недвоичные коды Рида-Соломона. Далее символы кодовых последовательностей внешнего кода кодируются внутренним кодом. С выхода внутреннего кодера кодовые символы каскадного кода поступают на вход модулятора и далее передаются в канал связи. На приемной стороне первоначально производится обработка информации внутренним декодером, а затем внешним декодером. С целью повышения корректирующей способности внутреннего кода к выходу внутреннего кодера подключается перемежитель кодовых символов. Сущность перемежения кодов состоит в рассредоточении ошибок, входящих в пакет по различным кодовым словам кодов, исправляющих случайные ошибки.

В беспроводных каналах связи такую достоверность практически не возможно получить без применения помехоустойчивого кодирования.

Однако применение мощных кодов с высокой исправляющей способность ограничено высокой сложностью реализации оптимальных декодеров, обеспечивающих минимальную вероятность ошибочного декодирования кодовых блоков. При выборе методов кодирования и главным образом методов декодирования, руководствуются многими факторами, которые делятся на три основные группы, взаимосвязь между которыми показана на рис. 1.

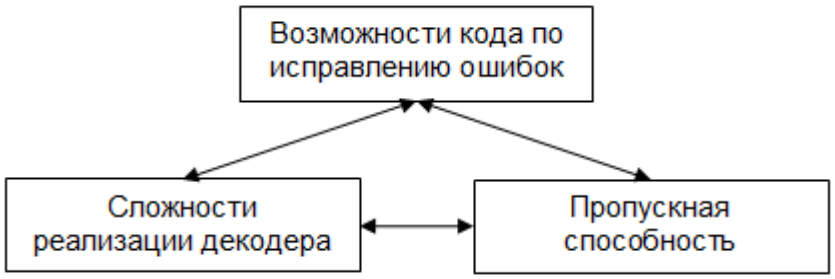


Рис. 1. Взаимосвязь между параметрами кодовых конструкций

Под сложностью реализации понимают аппаратные и программные затраты, стоимость микросхем и микропроцессоров, стоимость памяти для хранения данных и т.п. Под пропускной способностью, в данном контексте, понимают не только объемы полезной информации и избыточности, но и объемы служебной информации. Подобные сведения необходимы для установления и поддержания синхронизации передатчика и приемника, а также для управления элементами звена передачи данных. На практике чаще всего используются составные или каскадные коды.

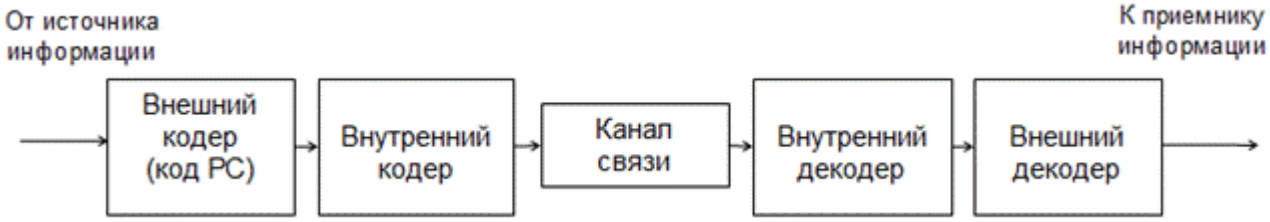


Рис. 2. Канал связи с использованием каскадного кода

Возможны различные варианты реализации каскадного принципа кодирования. Первоначально последовательность из $K_{инф} = K$ двоичных символов, являющихся информационными разбивается на $k_2 = k_{внеш}$ подблоковпо $k_1 = k_{вн}$ символов в каждом. Эти подблоки рассматриваются над двоичным полем Галуа степени расширения k_1, образуя группу информационных символов внешнего кода. Множество всех таких символов определяется $q = 2^{k_1}$. Внешний код формирует на

основе k_2 проверочные символы. Если в качестве внешнего кода используется код РС, то корректирующие возможности такого кода определяются выражением $d_2 = n_2 - k_2 + 1$. Проверочные символы этого кода являются элементами поля $GF\left(2^{k_1}\right)$. Все q-ичные символы комбинации кода (n2, k2) кодируются внутренним (n1, k1) кодом. В результате получается двоичный блоковый код длины $n_1 \times n_2$ содержащий $k_1 \times k_2$ информационны х двоичных символов с общим минимальным расстоянием $d_1 \times d_2$, где d_1 минимальное расстояние внутреннего кода. На рис. 1.19 представлена схема образования слова каскадного кода на основе кода РС.

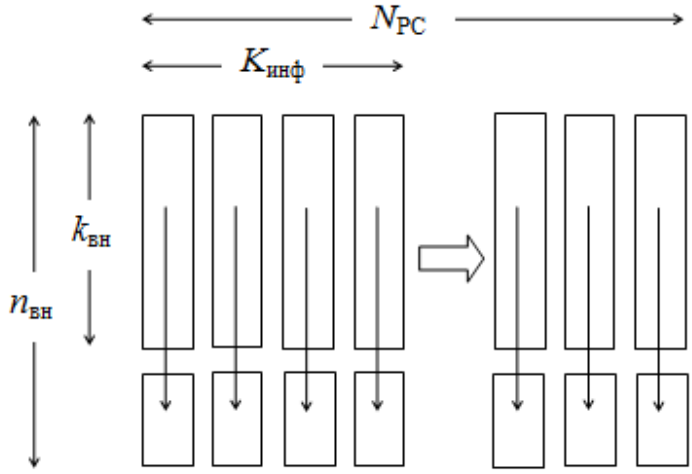


Рис. 3. Схема образования слова каскадного кода

Достоинством каскадных кодов является то, что они позволяют заменить декодирование длинного (n1, n2, k1, k2) кода декодированием двух значительно более коротких кодов – внутреннего двоичного (n1, k1) кода и внешнего (n2,k2) кода. Это позволяет говорить о линейном росте сложности декодера в зависимости от кратности исправляемых ошибок. Каскадные коды позволяют реализовать достаточно большое значение d, поэтому их применение имеет смысл в каналах с группирующимися ошибками.

Другое преимущество каскадных кодов состоит в том, что при исправлении ошибок внутренним кодом можно использовать не только различные конструктивные методы исправления независимых ошибок, но и оптимальные переборные методы, если (n1,k1) маломощный код. Свойство может быть использовано при декодировании блочных кодов, методом кластерного анализа. Этот алгоритм декодирования подобен

декодированию по списку, когда в кластер (список) входят наиболее вероятные образцы ошибок.

Сложность декодера как функция числа исправляемых кодом ошибок в системе с каскадным кодированием растет линейно тогда, как при использовании обычных кодов эта зависимость носит экспоненциальный характер.

Основная причина такого эффекта заключается в том, что при декодировании комбинаций внутреннего кода он не исправляет ошибки, а формирует стирания при обнаружении ошибки.

Стертые позиции восстанавливаются кодом РС, и поскольку стирания достаточно хорошо указывают на ошибочные позиции, корректирующие возможности кода используются не поиск ошибок, а на исправление стертых позиций.

Коды РС сроятся над конечными полями. Как было отмечено, такое поле может быть образовано для любого простого p и обозначается как GF(p). Понятие GF(p) обобщается на поле из pm элементов, именуемые полем расширением поля GF(p) степени расширения GF (pm). Поле GF (pm) содержит в качестве подмножества все элементы GF(p). Символы из поля расширения GF (2m) используются при построении кода РС.

Общепринятым считается представление кода РС через параметры (n_2, k_2, t_2) и некоторое $m > 2$, здесь t_2 – число, исправляемых кодом ошибок, при этом $(n_2, k_2) \Rightarrow (2^m - 1, 2^{n_2} - 1 - 2t_2)$. Генерирующий полином для кода РС имеет вид:

$$g(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \ldots + \alpha_{2t-1} x^{2t-1} + \alpha_{2t} x^{2t}.$$

Общее число информационных символов кода РС над двоичным полем GF (2m) оценивается выражением:

$$K = (2^m)^{k_2}.$$

Предположим что $k_2 = 2$, а m = 3, тогда K = 64. При этом образуется $2^{(k_2-1)m}$ групп комбинаций, в которых на первом месте среди информационных разрядов систематического кода РС будет находиться один и тот же представитель поля GF (2m) от 0 до α^6. Пример такого разбиения кода РС представлен в Приложении Б.

Заметно, что разряд X5 всех кодовых комбинаций (таблица представляет код РС с n2 = 7, k2 = 2) определяют конкретную группу комбинаций, которую назовем кластером. Номера кластеров определим как степень

примитивного элемента базового поля на месте разряда X5. Порождающий полином кода определен как:

$$g(x) = (x - \alpha) \cdot (x - \alpha^2) \cdot (x - \alpha^3) \cdot (x - \alpha^4) \cdot (x - \alpha^5) =$$

$$= x^5 + x^4\alpha^2 + x^3\alpha^3 + x\alpha^4 + \alpha,$$

здесь учтено, что операции сложения и вычитания в двоичном поле равнозначны. Кроме общеизвестных свойств кода РС выделим несколько важных с точки зрения последующих рассуждений.

Свойство 1. Любой систематический код РС в своем составе имеет 2^{k_2-1} комбинаций, состоящих из n2 одинаковых q-ичных элементов. Так же как двоичный групповой код содержит чисто единичный элемент (единичную комбинацию), q-ичный код должен содержать комбинации, состоящие из одинаковых элементов поля GF (2m), например, $\alpha^3\alpha^3\alpha^3\alpha^3\alpha^3\alpha^3\alpha^3$. В рамках параметров рассматриваемого кода первые пять символов являются проверочными, а последние два символа являются информационными разрядами. В [88] доказывается, что аналогичная картина сохраняется и для несистематических кодов РС (см. Приложение Б). Рассмотренное свойство может быть использовано для реализации метода синхронного накопления данных и декодирования кодовых комбинаций мажоритарным методом (за счет применения кодов повторителей), причем применение таких кодов является обязательным условием в процедуре реализации обобщенных каскадных кодов.

Свойство 2. Все множество V кодовых комбинаций кода РС для каждого разряда x^i порождающего полинома содержит одинаковое число элементов из поля $GF(2^m)$. Другими словами каждый элемент поля распределен по каждому разряду Xi общего множества кодовых комбинации кода РС с одинаковой плотностью. Например, в рассматриваемом коде РС (7,2,6), для разряда X0 элемент α или любой другой элемент базового поля повторяется только q раз. Исключение составляют только те разряды, которые совпадают с номером кластера. Следствием данного свойства является тот факт, что искажение символа в каждом разряде кодовой комбинации может произойти с вероятностью $p = (q-1/q)$.

Свойство 3. В любой комбинации систематического (несистематического) кода РС, не отвечающей свойству 1, отсутствует один из элементов поля, который заменяется нулевым элементом поля. Это свойство вытекает из определения длины кодовой комбинации кода РС, определяемой как $n = 2^m - 1$, и свойства цикличности.

На основании представленных свойств множества кодовых комбинаций кода РС возможна оценка верхней границы для вероятности ошибочного декодирования комбинаций такого кода. В качестве предварительного замечания отметим, что любой код с метрикой Хемминга d_{min} способен исправить $d_{min} = 2t + s$ ошибок и стираний, здесь t – число ошибок в кодовой комбинации, а s – число стираний. При исправлении стираний целесообразно принять значение t равным единице. Это связано с тем, что среди символов с ИДС равных максимальной оценке с определенной долей вероятности не исключены ошибочные символы. Принимая $t = 1$, получаем некоторый запас по коррекции стираний. Тогда $d_{min} = 2 + s$ и окончательно $s = d_{min} - 2$. Это число стираний, исправляемых кодом и обеспечивающее запас корректирующей способности в случае возникновения не выявленной ошибки. Пусть внутренним кодом обнаруживаются ошибки и q-ичные символы (подблоки) с обнаруженными ошибками стираются, если число стертых подблоков больше $n_2 - k_2$, то стирается вся комбинация кода РС, а если число стираний меньше или равно $n_2 - k_2$, то стирания исправляются кодом РС. Внешний код не обнаруживает ошибку в случае, если нестертые q-ичные символы совпадут в соответствующих местах с символами одной из кодовых комбинаций, отличной от переданной.

В такой конструкции проявляются два очень важных свойства. Первое из них заключается в том, что при использовании адаптивных режимов в условиях высокого качества канала связи может быть повышена скорость кода за счет выкалывания проверочных символов, относящихся к проверочным разрядам внешнего кода, т.е. выкалывание (перфорация) символов, относящихся к проверкам проверок.

Рис. 4. Конструкция слова каскадного кода

Вторым положительным свойством конструкции слова каскадного кода является возможность применения напрямую процедуры перемежения символов непосредственно к матрице, с помощью которой это слово представляется. Процедура перемежения символов заключается в предварительном заполнении информационными разрядами матрицы памяти некоторой размерности. Если запись данных от источника информации в указанную матрицу осуществляется по строкам, то после ее заполнения считывание данных в канал связи выполняется по столбцам. Это делается для того, чтобы противостоять наиболее сложному виду помехи, которая в канале связи проявляется в виде группирующихся ошибок. Если известна средняя вероятность ошибки на символ в данном типеканала связи p_s, то внутри пачки ошибок значение этого параметра $p_{гр}$, при этом $p_{гр} >> p_s$. Подобные устройства описаны в и имели название декоррелятора ошибок, что более точно отражает их суть. Приемник последовательно фиксирует столбцы данных и записывает их в виде столбцов в матрицу памяти аналогичную по размерности матрице на передаче. После заполнения матрицы приема, данные в декодер списываются построчно.

Поскольку в канале связи группирующаяся помеха воздействовала на символы столбца, то при построчном считывании комбинаций из матрицы приема в декодер в каждой такой комбинации будет ограниченное число ошибок, которые могут быть обнаружены и исправлены. Применение подобных устройств связано с задержкой данных при их обработке $t_{дек}$, которая оказывается не столь велика при высоких рабочих частотах процессоров приемников. Пусть матрица перемежителя (деперемежителя) имеет размерность $2^9 \times 2^9$ и рабочая частота процессора составляет 2 ГГц, следовательно, время заполнения матрицы данными займет около 0.5 мс, а с учетом задержек на передаче и приеме около 1 мс. Применение перемежителей в современных системах передачи данных связывается с ТК, где им в соответствии с постулатами К. Шеннона отводится роль элемента случайного кодирования.

Контрольные вопросы:

1. Каким образом производится кодирование и декодирование информации в каскадных кодах?
2. В чем сущность перемежения кодов?
3. В чем достоинство каскадных кодов?

4. Опишите свойства конструкции слова каскадного кода?
5. Какая наиболее распространенная схема построения каскадных кодов?

Лекция №14. Турбо коды

Турбо-код — параллельный каскадный блоковый систематический код, способный исправлять ошибки, возникающие при передаче цифровой информации по каналу связи с шумами. Синонимом турбо-кода является известный в теории кодирования термин — каскадный код (англ. *concatenatedcode*) (предложен Д. Форни в 1966 году).
Турбо-код состоит из каскада параллельно соединённых систематических кодов. Эти составляющие называются компонентными кодами. В качестве компонентных кодов могут использоваться свёрточные коды, коды Хемминга, Рида — Соломона, Боуза — Чоудхури — Хоквингема и другие. В зависимости от выбора компонентного кода турбо-коды делятся на свёрточные турбо-коды (англ. *TurboConvolutionalCodes, TCC*) и блоковые коды-произведения (англ. *TurboProductCodes, TPC*).
Турбо-коды были разработаны в 1993 году и являются классом высокоэффективных помехоустойчивых кодов с коррекцией ошибок, используются в электротехнике и цифровой связи, а также нашли своё применение в спутниковой связи и в других областях, в которых необходимо достижение максимальной скорости передачи данных по каналу связи с шумами в ограниченной полосе частот.

Структура турбо-кода

Согласно Шеннону, наилучшим кодом является код, который передает сообщение за бесконечно большое время, формируя в каждый момент времени случайные кодовые элементы. У приёмника есть бесконечные версии сообщения, искажённого случайным образом. Из этих копий декодер должен выбрать копию, наиболее близкую к переданному сообщению. Это представляет собой теоретически идеальный код, который может исправить все ошибки в сигнале. Турбо-код является шагом в этом направлении. Ясно, что мы не должны посылать информационное сообщение в течение бесконечного времени. Для приемлемой работы достаточно удвоить или утроить время передачи, что обеспечит довольно приличные результаты для каналов связи.
Особенностью турбо-кодов является параллельная структура, состоящая из рекурсивных систематических сверточных (RSC) кодов, работающих

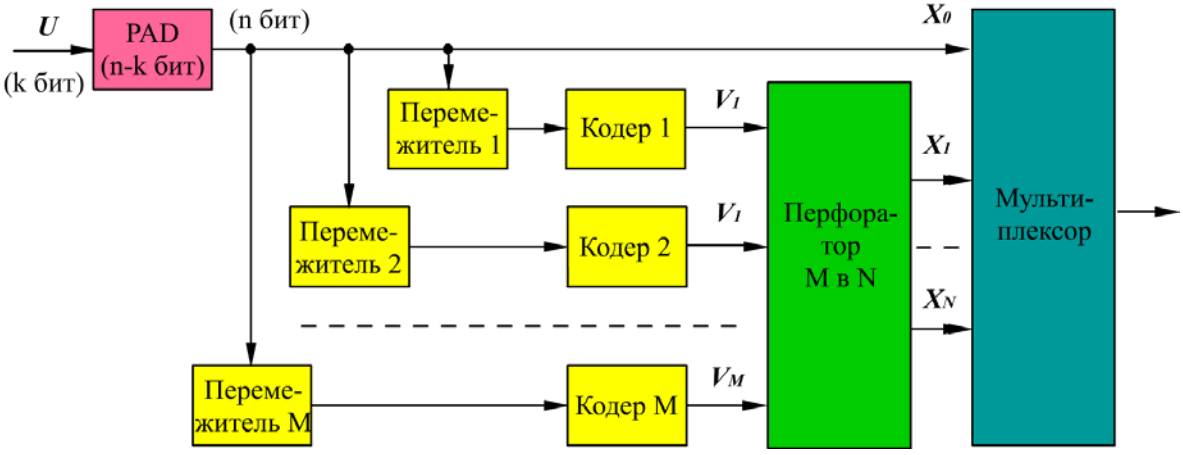
параллельно и использующих создание *случайной* версии сообщения. Параллельная структура использует два или больше кодов RSC, каждый с различным перемежителем. Цель перемежителя состоит в том, чтобы предложить каждому кодеру некоррелированную или *случайную* версию информации, в результате чего паритетные биты каждого RSC становятся независимыми.

В турбо-кодах блоки имеют длину порядка нескольких Кбит. Цель такой длины состоит в том, чтобы эффективно рандомизировать последовательность, идущую на второе кодирующее устройство. Чем длиннее размер блока, тем лучше его корреляция с сообщением первого кодера, то есть корреляция мала.

Существует несколько схем турбо-кодов:

- PCCC — в случае конкатенации параллельных сверточных кодов
- SCCC — схема с последовательным соединением сверточных кодов, коды *SCCC* имеют высокие характеристики при больших отношениях сигнал/шум
- TPC — турбо-код-произведение, использует блочные коды вместо сверточных; два различных блочных кода (обычно коды Хемминга) соединены последовательно без промежуточного перемежителя. Так как два кода независимы и работают в рядах и колонках, что само по себе обеспечивает достаточно хорошую рандомизацию, то применение перемежителя не требуется.

Кодирование



На рис. 1 представлена общая структурная схема M-блочного турбо-кодера. Сначала на вход формирователя пакетов PAD (англ. *PacketAssembler/Disassembler*) поступает блок данных U длиной k бит. В формирователе пакетов к данным

прибавляется ещё $(n-k)$ дополнительных бит служебной информации, соответствующих используемому стандарту формирования пакета и включающих в себя символы его начала и окончания[https://ru.wikipedia.org/wiki/%D0%A2%D1%83%D1%80%D0%B1%D0%BE-%D0%BA%D0%BE%D0%B4 - cite_note-ref_-_4-4](). То есть получается пакет X_0, состоящий из n бит.

Далее последовательность бит X_0 поступает параллельно на M ветвей, содержащих последовательно соединённые перемежитель и компонентный кодер. Таким образом X_0используется в качестве входных данных сразу всеми компонентными кодерами.

Перемежение в турбо-кодах

В перемежителях по псевдослучайному закону происходит перемешивание поступающих бит. В отличие от посимвольного прямоугольного перемежителя, используемого в кодах Рида-Соломона, в турбо-кодах используется перемежение отдельных бит, которое подобно случайным перестановкам. Причём впоследствии, при операциях декодирования этот закон перемежения будет считаться известным. Полученные последовательности поступают на входы кодеров.

Задача перемежителя — преобразовать входную последовательность так, чтобы комбинации бит X_0, соответствующие кодовым словам с низким весом (весом называется число ненулевых бит кодового слова) на выходе первого кодера, были преобразованы в комбинации, дающие кодовые слова с высоким весом на выходах остальных кодеров. Таким образом кодеры получают на выходе кодовые слова с различными весами. При кодировании формируются кодовые слова так, чтобы получалось максимально возможное среднее расстояние между ними (расстоянием между двумя кодовыми словами называется число бит, в которых они различаются). Из-за того что кодовые блоки формируются из почти независимых частей, на выходе турбо-кодера среднее расстояние между кодовыми словами больше, чем минимальное расстояние для каждого компонентного кодера, а следовательно растёт эффективность кодирования.

Перестановка для каждой указанной длины блока k задается определенным переупорядочиванием целых чисел $1, 2..., k$ как предусмотрено следующим алгоритмом (ECSS-E-ST-50-01C)[5].

$k = 8 * k_0$, где $k_0 =$ одному из следующих значений : $223, 223 * 2, 223 * 4, 223 * 5$, в зависимости от необходимой глубины перемежителя

Следующие операции выполняются для значений от $s = 1$ до $s = k$, чтобы получить адреса перестановки $\boldsymbol{\pi}(s)$. В уравнениях ниже, $\lfloor \boldsymbol{x} \rfloor$ обозначает наибольшее целое число, меньше или равное $\boldsymbol{x}$, и $\boldsymbol{p}_q$ обозначает одно из следующих четырёх простых чисел: $p_0 = 31, p_1 = 37, p_2 = 43, p_3 = 47$,

$$m = (s-1) \mod 2$$

$$i = \lfloor \frac{s-1}{2 * k_0} \rfloor$$

$$j = \lfloor \frac{s-1}{2} \rfloor - i * k_0$$

$$q = (19 * i + 1) \mod 4$$

$$c = (p_q * j + 21 * m) \mod k_0$$

$$\boldsymbol{\pi}(s) = 2 * (q + 4 * c + 1) - m$$

Интерпретация перестановки чисел такова, что $\boldsymbol{s}$-й бит, переданный перемежителем, является $\boldsymbol{\pi}(s)$-м битом входного информационного блока. Деперемежитель осуществляет запись принятого бита по вычисленному адресу.

Кодовая скорость

Кодовая скорость — отношение длины кодового блока на входе к длине преобразованного кодового блока на выходе кодера.

В отсутствие перфоратора (см. рис. 1) исходная последовательность X_0 мультиплексируется с последовательностями проверочных бит $V_1, \ldots, V_M$, образуя кодовое слово, подлежащее передаче по каналу. Тогда значение кодовой скорости на выходе турбо-кодера

$$R = \frac{k}{n(M+1)}$$

Для увеличения кодовой скорости применяется выкалывание (перфорация) определённых проверочных битов выходной последовательности. Таким образом кодовая скорость возрастает до

$$R = \frac{k}{n(N+1)}$$

где $N < M$, причём N может быть дробным, если число оставшихся после перфорации проверочных бит не кратно n

Если учесть, что турбо-коды оперируют с блоками большой длины с $k > 10000$, то $k \approx n$, и кодовая скорость равна

$$R = \frac{1}{N+1}$$

Из приведённых формул видно, что с помощью перфоратора, выкалывая разное число проверочных бит, возможно регулирование кодовой скорости. То есть можно построить кодер, адаптирующийся к каналу связи. При сильном зашумлении канала перфоратор выкалывает меньше бит, чем вызывает уменьшение кодовой скорости и рост помехоустойчивости кодера. Если же канал связи хорошего качества, то выкалывать можно большое число бит, вызывая рост скорости передачи информации

Декодирование

Алгоритм декодирования по максимуму апостериорной вероятности (MAP) При осуществлении декодирования с исправлением ошибок существенен анализ априорной и апостериорной вероятностей прихода верного кодового слова. Априорной называется информация, которой обладает декодер до прихода кодового слова, а апостериорной называется информация, полученная после обработки кодового слова.
В своей работе Берроу предлагает для использования в турбо-декодерах алгоритм максимума апостериорной вероятности (англ. *Maximumof A-posterioriProbability, MAP*), также известный под названием алгоритма Бала (Bahl — Cocke — Jelinek — Raviv (BCJR)). Алгоритм Бала дает «мягкую» оценку достоверности декодированного бита. То есть предъявляет на выходе степень доверия результату декодирования. В противоположность «жёсткой» структуре, при которой на выходе декодера формируется лишь наиболее вероятное значение декодированного бита («0» или «1»), при вынесении «мягкого» решения используется более подробная дискретизация выходного сигнала, характеризующая вероятность корректного приема бита. Благодаря использованию «мягких» решений в турбо-декодерах оказывается эффективным использование нескольких итераций декодирования. Апостериорная информация, полученная о кодовом слове на выходе первой итерации декодирования,

поступает на вход блока следующей итерации и является для него уже априорной вероятностью. Такой подход позволяет улучшать качество декодирования от итерации к итерации. Таким образом, изменяя число итераций декодирования, можно адаптировать декодер к текущему состоянию канала передачи и достичь требуемой вероятности ошибки на бит

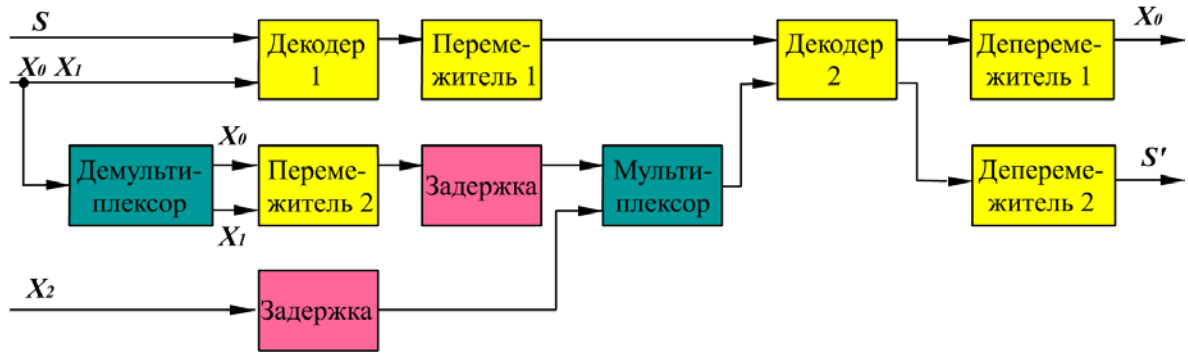
Логарифмическое отношение правдоподобия (LLR)

Рассмотрим информационный бит как бинарную переменную u_k, то есть — значение u в момент времени k. Его логарифмическое отношение правдоподобия (LLR) определено как логарифм отношения его основных вероятностей.

$$L(u_k) = \ln \frac{Pr(u_k = 1)}{Pr(u_k = 0)}$$

Эта метрика используется в большинстве систем исправления ошибок с помощью помехоустойчивого кодирования и называется логарифмическим отношением правдоподобия или LLR. Она немного лучше, чем линейная метрика, так как, например, логарифм облегчает обработку очень маленьких и очень больших значений. Если вероятности приёма «0» и «1» равны, метрика равна 0.

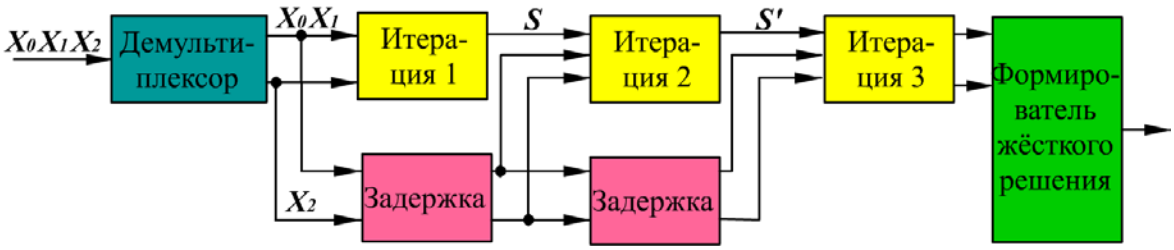
Одна итерация итеративного турбо-декодера при двухкаскадном кодировании



На рис. 2 для простоты понимания представлен вариант схемы одной итерации турбо-декодирования при двухкаскадном кодировании. Эта схема несложно обобщается на случай произвольного количества каскадов кодирования.

Декодер для одной итерации содержит каскадное соединение двух элементарных декодеров, каждый из которых, основываясь на критерии максимума апостериорной вероятности, выносит «мягкое» решение о переданном символе. На первый декодер первой итерации с выхода демодулятора поступают «мягкие» решения символов последовательностей X_0 и X_1. Таким образом на выходе первого декодера появляется оценка информационного символа, которая после последующего перемежения попадает на вход второго декодера и является для него априорной информацией. Используя «мягкое» решение о последовательности X_2, второй декодер формирует свою оценку

Трёхитерационный турбо-декодер при двухкаскадном кодировании



С выхода каждой итерации решение переходит на вход следующей. Организация работы трёхитерационного турбо-декодера показана на рис. 3. От итерации к итерации происходит уточнение решения. При этом каждая итерация работает с «мягкими» оценками и на выход отдает также «мягкие». Поэтому такие схемы получили название декодеров с мягким входом и мягким выходом (англ. *SoftInputSoftOutput (SISO)*). Процесс декодирования прекращается либо после выполнения всех итераций, либо когда вероятность ошибки на бит достигнет требуемого значения. После декодирования из полученного «мягкого» решения производится окончательное «жёсткое»

Преимущества и недостатки турбо-кодов

Преимущества

Среди всех практически используемых современных методов коррекции ошибок турбо-коды и коды с низкой плотностью проверок на чётность наиболее близко подходят к границе Шеннона, теоретическому пределу максимальной пропускной способности зашумленного канала. Турбо-коды позволяют увеличить скорость передачи информации, не требуя увеличения мощности передатчика, или они могут быть использованы для уменьшения требуемой мощности при передаче с

заданной скоростью. Важным преимуществом турбо-кодов является независимость сложности декодирования от длины информационного блока, что позволяет снизить вероятность ошибки декодирования путём увеличения его длины

Недостатки

Основной недостаток турбо-кодов — это относительно высокая сложность декодирования и большая задержка, которые делают их неудобными для некоторых применений. Но, например, для использования в спутниковых каналах этот недостаток не является определяющим, так как длина канала связи сама по себе вносит задержку, вызванную конечностью скорости света.

Ещё один важный недостаток турбо-кодов — сравнительно небольшое кодовое расстояние (то есть минимальное расстояние между двумя кодовыми словами в смысле выбранной метрики). Это приводит к тому, что, хотя при большой входной вероятности ошибки (то есть в плохом канале) эффективность турбо-кода высока, при малой входной вероятности ошибки эффективность турбо-кода крайне ограничена. Поэтому в хороших каналах для дальнейшего уменьшения вероятности ошибки применяют не турбо-коды, а LDPC-коды.

Хотя сложность используемых алгоритмов турбо-кодирования и недостаток открытого программного обеспечения препятствуют внедрению турбо-кодов, в настоящее время многие современные системы используют турбо-коды.

Применение турбо-кодов

Компании FranceTelecom и TelediffusiondeFrance запатентовали широкий класс турбо-кодов, что ограничивает возможность их свободного применения и, в то же время, стимулирует развитие новых методов кодирования таких, как, например, LDPC.

Турбо-коды активно применяются в системах спутниковой и мобильной связи, беспроводного широкополосного доступа и цифрового телевидения. Турбо-коды утверждены в стандарте спутниковой связи DVB-RCS. Турбо-коды также нашли широкое применение в мобильных системах связи третьего поколения (стандарты CDMA2000 и UMTS)

Контрольные вопросы:

1. Результатом чего являются турбо-коды?

2. С помощью чего осуществляется декодирование турбо-кода?
3. Поясните смысл перемежения для блоковых кодов?
4. Каковыпараметрыправилперемежения?
5. Изобразите перемежение символов и блочное перемежение

Лекция №15. Весовой спектр кода

Как известно весовой спектр кода- это есть множество чисел M(ω),где M(ω) – число кодовых комбинаций весов в кодовом множестве (ω =0,n). То есть вес кодовой комбинации определяется суммой ее ненулевых составляющих .Для кодов у которых число кодовых комбинаций не велико (то есть сравнительно не большое число *к*- информационных разрядов) весовой спектр можно определить непосредственным перебором этих $2^{к}$ разрешенных комбинаций. Подобные коды небольшой длинны ,обычно БЧХ коды ,используются в первой ступени каскадного кода ,для исправления ошибок небольшой кратности при низком качестве дискретного канала .Известны два способа получения разрешенных кодов комбинаций циклического кода :

1. Умножением кодовой комбинации Q(x) простого кода на одночлен x^r и добавлением к этому произведению остатка R(x) ,полученного в результате деления произведения $Q(x)*x^r$на образующий полином P (x)

$$F(x) =Q(x)*x^r+R(x) \qquad (1)$$

2. умножение кодовой комбинации Q(x)простого кода на образующий полином

$$F(x) =Q(x)*P(x) \qquad (2)$$

Первый способ позволяет построить систематический код ,в котором информационные символы находятся на первых *к*-позициях, проверочные – на остальных r-позициях . При использовании второго способа получения несистематический код, в котором в явном виде нет ни информационных ни проверочных символов. Однако, с точки зрения помехоустойчивости систематический и несистематический коды совершенно идентичны ,так как кодовые множества остаются теми же ,а меняется лишь порядок ,по которому каждому информационному слову ставится в соответствии кодовое слово . Правильность этого утверждения можно проследить на примере кода построенного с использованием первого и второго способов ,наглядно изображенного на рис. 1

С целью автоматизации построения разрешенных комбинаций кода первым способом могут быть использованы те или иные алгоритмы метода

программной реализации процедур кодирования информации циклическими кодами:

a) деление кодовой информации по частям;
b) непосредственноеделениекодовойинформации;
c) матричныйметод;
d) табличныйметод;

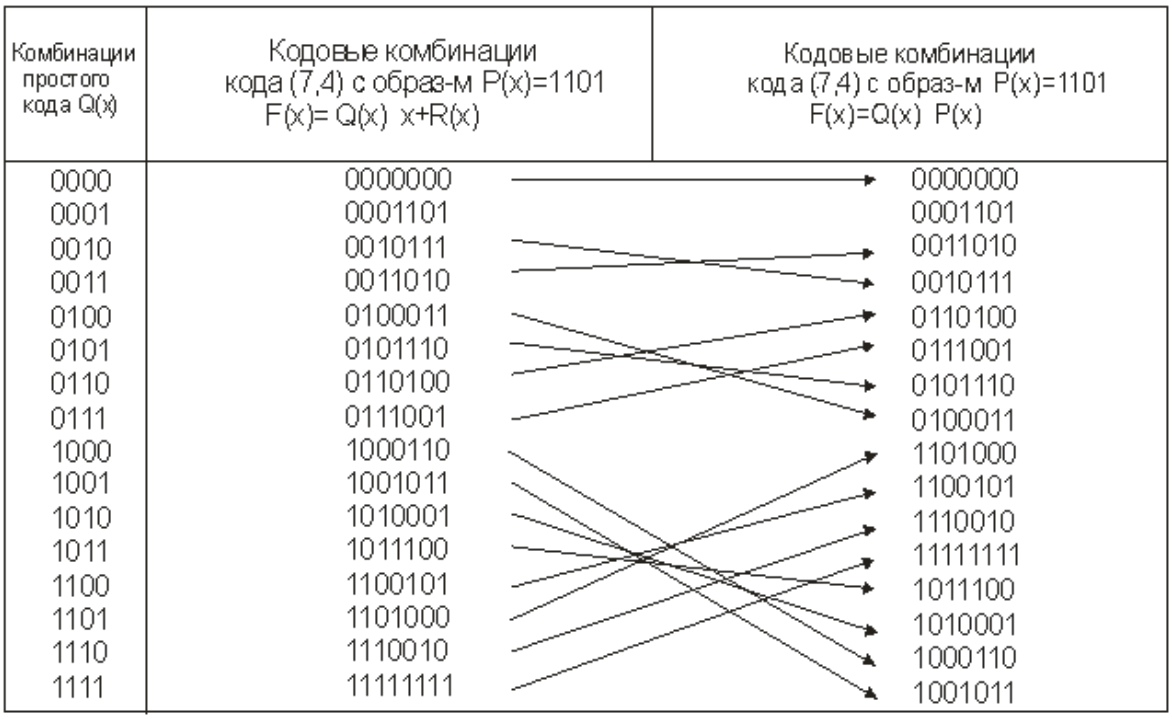


Рис 1 Взаимосвязь кодовых комбинаций, полученных разными способами

Согласно алгоритмов соответствующих методов кодирования оценим сложность программной реализации. Для этого необходимо произвести расчет загрузки, объема памяти и требуемого времени при кодировании. Количество операций, выполняемых в процессе обработки кодовой комбинации, определяется как:

$A_0=A_п+A_цK$ (3)

где A_0 – общее количество операций, необходимых для кодирования;

$A_п$ – число операций, выполняемых 1раз;

$A_ц$ – число операций, выполняемых в цикле;

К – количество циклов, меняющиеся в зависимости от метода кодирования.

Рассмотрим сложность реализации метода кодирования делением кодовой комбинации по частям. Рассчитанные данные величин загрузки ЭВМ и необходимого времени кодирования в зависимости от длины кодовой комбинации и количества проверочных разрядов приведены в таблице 1

Таблица 1. Зависимость основных параметров метода деления КК по частям от длины кода

Длина кода параметры	7,4	15,11	31,26	63,57	127,120	255,247
Загрузка (опер/бит)	15,25	14,45	14,19	14,08	14,04	14,02
Время мс	0,061	0,159	0,369	0,803	1,685	3,46

Из данных таблицы видно, что данный метод удобен для кодирования больших длин кодов, так как он не требует дополнительных затрат памяти на кодовую таблицу.

Рассчитанные данные величин загрузки ЭВМ и необходимого времени кодирования в зависимости от длины кодовой комбинации и количества проверочных разрядов для метода кодирования непосредственным делением полинома на образующий полином.

Таблица 2. Зависимость основных параметров метода с непосредственным делением КК от длины кода

Длина кода параметры	7,4	15,11	31,26	63,57	127,120	255,247
Загрузка (опер/бит)	8,7	7,38	7,26	7,17	7,15	7,05
Время мс	0,105	0,237	0,567	1,227	2,613	5,451

Из данных таблицы видно, что загрузка с увеличением длины кода уменьшается, а время необходимое для кодирования увеличивается. Метод кодирования делением по частям требует дополнительных затрат памяти на кодовую таблицу, причем ее объем тем больше, чем больше длина l разрядного блока.

При методе кодирования с использованием определяющей матрицы строится определяющая матрица $A_{n,m}$. Для ее получения требуется найти проверочные элементы дополнительной матрицы. Проверочные элементы находят последовательным делением на образующий полином единицы с приписанными справа нулями и запоминанием строк дополнительной матрицы с промежуточными остатками.

В таблице 3 приведены рассчитанные данные величин загрузки ЭВМ и необходимого времени кодирования в зависимости от длины кодовой комбинации и количества проверочных разрядов для матричного метода.

Таблица 3. Зависимость основных параметров
матричного метода от длины кода

Длина кода параметры	7,4	15,11	31,26	63,57	127,120	255,247
Загрузка (опер/бит)	3,75	2,63	2,26	2,12	2,05	2,028
Время мс	0,075	0,089	0,099	0,121	0,247	0,601

Из данных данной таблицы видно, что матричный метод не требует больших затрат машинного времени. Однако необходимый объем памяти для таблиц растет с увеличением длины кода.

При реализации табличного метода в память ЭВМ записывается матрица состоящая из все возможных информационных комбинаций расположенных в порядке возрастания натуральных чисел от 0 до $2^к$ с соответствующими им проверочными разрядами. Проверочные разряды подсчитываются заранее по правилу кодирования данного кода. При этом достаточно знать номер строки таблицы, на которую указывают значение кодовой комбинации, чтобы определить закодированное слово. Рассчитанные для табличного метода данные величин объема памяти, загрузки и времени в зависимости от длины кода приведены в таблице 4.

Из данных таблицы видно, что этот метод кодирования требует минимальных затрат памяти, загрузки и машинного времени при коротких длинах кода. При увеличении дли кода резко увеличивается необходимый объем памяти для таблиц.

Результаты анализа вышеприведенных таблиц показывают, что при небольших длинах кода наиболее удобным в отношении сложности реализации является матричный метод, так как он требует небольшой загрузки ЭВМ и небольшого времени кодирования. Так же хорошие показатели в отношении загрузки и времени обеспечивает метод деления по частям путем изменения величины l. Однако при использовании кодов большой длины оба метода становятся малоэффективными, так как требуют значительных затрат памяти ЭВМ на таблицы. При реализации длинных кодов лучшим является метод непосредственного деления, не требующий дополнительных затрат памяти ЭВМ на таблицы.

Таблица 3.4Зависимость основных параметров
табличного метода от длины кода

Длина кода папараметры	7,4	15,11
Объем памяти (ячеек)	15	23
Загрузка (опер/бит)	1,25	0,45
Время мс	0,015	0,015

Более приемлемым с точки зрения сложности реализации является второй способ построения разрешенных кодовых комбинаций. Его применение упрощается при использовании формулы произведения многочлена Q(x) на неприводимый многочлен P(x).

В основу разработки алгоритма определения весового спектра на основе кодирования был положен способ получения кодовой комбинации циклического кода путем деления разрешенной кодовой комбинации на образующий полином.

Начинается алгоритм с объявления переменных и резервирования памяти (блок 2). В этом блоке определяем какие переменные и массивы будут

необходимы для записи и хранения промежуточных данных и конечного результата.

Затем необходимо ввести исходные данные: образующий полином P(x) длину информационного полинома k (блок 3).

Длина информационной части необходима для определения количества переборов, т.е. всех возможных комбинаций кодового слова (блок 4). В блоке 5 происходит вычисление проверочных разрядов кодовой комбинации r. Затем вычисляется многочлен x^r (блок 6). В блоке 7 идет перемножение двух многочленов x^r и $Q_i(x)$. Этот промежуточный результат запоминается памятью и проверяется условие все ли возможные комбинации информационной последовательности мы просчитали (блок 8). Следующим шагом является деление полученного результата на образующий полином P(x) (блок 9). Это нужно для определения остатка R(x), формирование которого происходит в блоке 10. Остаток от деления R(x) находится для каждой комбинации и поэтому проверяется условие есть ли еще кодовые комбинации (блок 11). В блоке 12 осуществляется формирование кодовой комбинации с проверочными символами F(x). Получив кодовую комбинацию можно приступать к подсчету весов (блок 13). Теперь перейдем к печати значений весов (блок 14).Структурная схема алгоритма приведена на рис 3.2

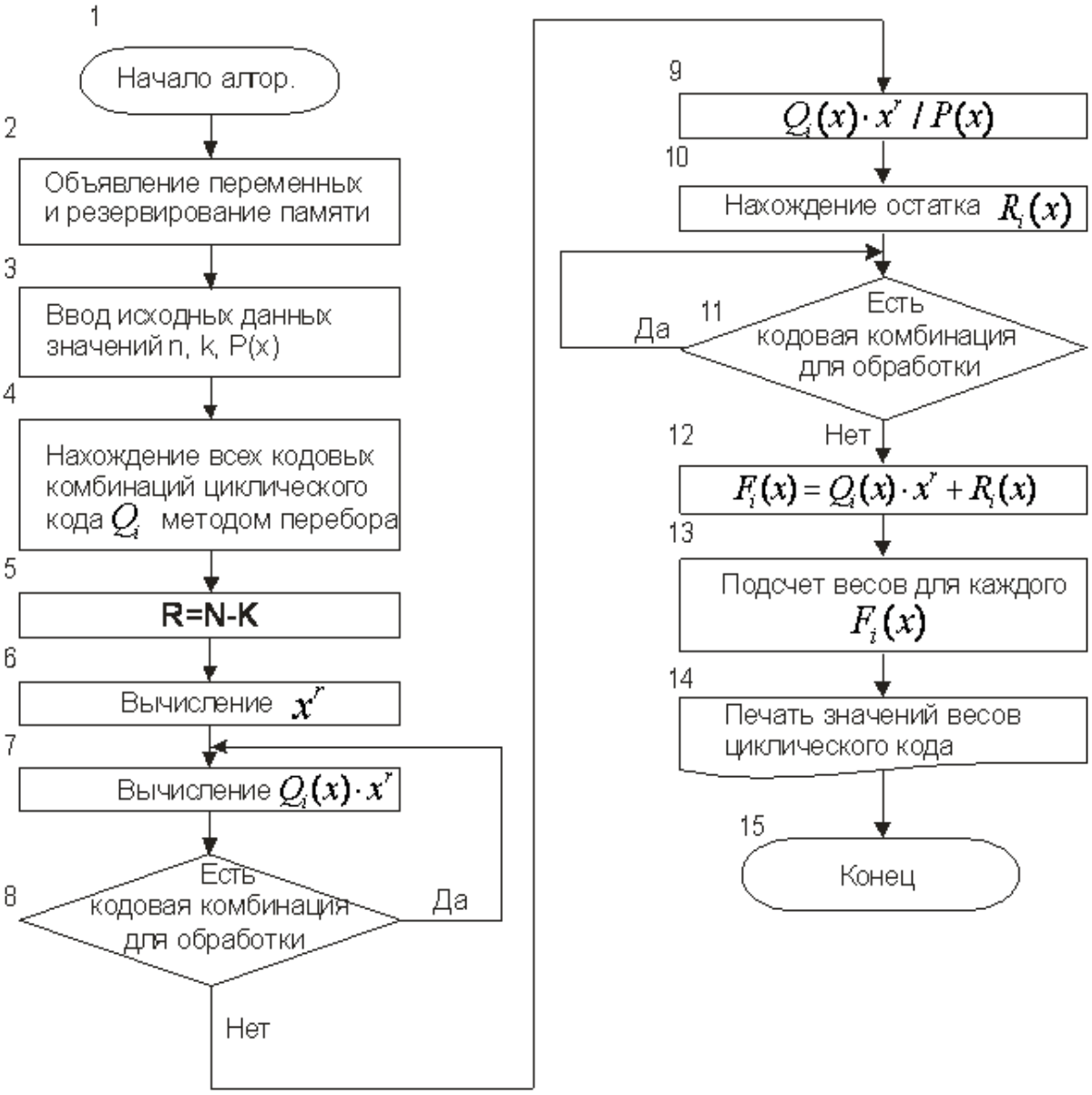


Рис. 2 Структурная схема алгоритма подсчета весов

В основу разработки алгоритма определения весового спектра на основе умножения полиномов был положен способ получения кодовой комбинации циклического кода умножением двух многочленов : образующего и информационного.

Начинается алгоритм с объявления переменных и резервирования памяти (блок 2). В этом блоке определяется какие переменные и массивы будут необходимы для записи и хранения промежуточных данных и конечного результата (c1, c2, c3)

Затем необходимо ввести исходные данные: образующий полином (50) и длину информационного полинома (№1) (блок3).

Длина информационной части необходима для определения количества переборов, т.е. всех возможных комбинаций кодового слова. Кроме

определения количества приборов нужно определить начальные значения c1 и c2 (блок 4)

Эти переменные будут необходимы для записи промежуточных результатов. В блоке 5 происходит умножение двух многочленов и получения кодовой комбинации циклического кода 51. Получив кодовую комбинацию можно приступать к подсчету весов (блок 6).Прежде чем закончить вычисления необходимо определить все ли возможные комбинации информационной последовательности мы просчитали (блок 7). Принцип перебора – это операция циклического сдвига комбинации влево. Если осуществлен полный перебор, то перейдем к печати значений весов (блок 9). Если же нет, то определим следующее значение c1 (блок 8),и повторим процесс вычисления весов кодовой комбинации ции. На этом процесс нахождения весов можно считать законченным (блок 10).

Контрольные вопросы:

1. Весовойспектркода – это…
2. Какие два способа известны для получения разрешения кодовых комбинаций?
3. Объяснитесущностьпринципаперебора?

Лекция №16. Декодирование с мягким решением

Декодирование блочного кода может быть осуществлено посредством жёсткого или мягкого принятия решения, а на выходе из декодера мы получим жёсткие или мягкие данные. В декодировании с жёстким принятием решения, каждому принимаемому биту в демодуляторе приписывается значение 0 или 1, в зависимости от того, принимаемые данные с помехами больше или меньше порогового значения. В декодере применяется избыточнаяинформация, добавленная в кодере, для определения наличия ошибок и, по возможности, их исправления. Искомые выходные данные декодера - это исправленное кодовое слово.

Декодер с мягким принятием решения принимает не только бинарную величину 1 или 0, но также доверительная величина, связанная с заданным битом. Если модулятор определился, то биту присваивается значение 1, степень уверенности в нём высока. Если он менее определён, то он помещает более низкую доверительную величину. Декодер с мягким входом может выпускать данные с жёстким решением или данные с мягким решением. Например, декодирующее устройство Витерби принимает мягкую информацию из демодулятора и выпускает данные с жёстким

решением. Декодер может использовать мягкую информацию для определения твёрдого равенства заданного бита 0 или 1, бит на выпуске мы получим такое жёсткое решение.

Декодер с мягким входом и мягким выходом (SISO) принимает данные с мягким решением и выпускает данные с мягким решением. Для каждого бита кодового слова, декодер SISO оценивает степень уверенности в других битах кодового слова и, используя избыточную информацию кода, производит усовершенствованные мягкие данные для заданного бита.

RS-коды коррекции ошибок являются стандартным алгоритмом для упреждающей коррекции ошибок (FEC). RS-коды - это блочные коды, обладающие хорошей способностью исправления ошибок при применении как в программном, так и в аппаратном обеспечении. RS-коды - это коды с жёстким принятием решения. За RS-кодами следуют каскадныекодыВитерби (RSV), которые превосходят автономные RS-коды в отношении интенсивности появления ошибочных битов.

Концепция декодеров SISO применена в турбо-кодах. Турбо-код подаёт демодулированные данные с мягким решением в SISO-декодер. Выходныеданныеэтого декодера затем подаются в тот же самый (или другой) SISO-декодер. Затем операция повторяется снова. Этот процесс итерации продолжается до тех пор, пока не будет принято уверенное решение. Концепция подачи выходных данных обратно на вход аналогична турбонаддуву двигателя, поэтому турбо-код получил именно такое название.

Для эффективности применения турбо-кода, имеющиеся данные следует закодировать двумя (или более) различными кодами. Затем, при декодировании, каждый из кодов будет изменять степень уверенности в отношении каждого бита. При каждой итерации, все коду изменяют степень уверенности в данных, так что каждый код сопровождает немного отличные данные для каждой итерации. Каждый код снижает или повышает степень доверия к заданному биту, и вследствие этого изменяет величину жёсткости решения в отношении ошибочных битов. В конце концов, данные получат такой вид, что все коды будут только повышать уверенность во всех битах. Величины жёсткого решения на этом этапе приближаются к передаваемым данным.

Рассмотрим передачу двоичных сигналов по каналу с шумом, при использовании сверточных кодов. Существуют два подхода к декодированию помехоустойчивых кодов, основанных на полученной после демодуляции последовательности действительных чисел.

Декодирование с «жестким» решением (hard decision decoding). В этом случае каждому действительному числу сопоставляется 0 или 1, т.е. при

использовании жестких решений относительно принятых из канала величин происходят ошибки типа "инверсии" принимаемых символов. В этом случае при декодировании по принципу максимального правдоподобия используется расстояние Хэмминга.

Декодирование с «мягким» решением (soft-decision decoding). Принятые из канала величины квантуются на заданное число уровней, и уровни кодируются числами в некотором интервале, например, кодируются посредством величин -4, -3, -2, -1, 0, 1, 2, 3 при квантовании на 8 уровней. В этом случае при декодировании по принципу максимального правдоподобия используется обычное евклидово расстояние.

Вообще говоря, для выбора того или иного решения желательно знать статистику шума в канале связи. Кроме того, необходимо помнить, что декодирование с «жестким» решением имеет солидную теоретическую базу, которая гарантирует для заданного кода соответствующую корректирующую способность, Для «мягкого» решения такая теоретическая база практически отсутствует. С другой стороны можно ожидать, что при «мягком» решении количество ошибок, выявленных декодером, будет меньше, чем при «жестком» решении (за счет большего числа уровней), и соответственно один и тот же код сможет исправить большее количество ошибок, например, в канале с аддитивным белым гауссовым шумом (АБГШ). Целесообразность применения декодирования с «мягким» решением также может быть обоснована тем, что по своей природе шумовая компонента в задаче восстановления данных или приема сигналов является непрерывной, а не дискретной. Это означает, что принятые символы более естественно представляются (квантуются) действительными числами (соответствующими напряжению, току), а не двоичными символами из конечного поля GF(2m).

Алгоритм Витерби для «мягкого» решения не отличается от алгоритма для «жесткого» решения, за исключением того, что расстояния вычисляются не по Хэммингу, а как евклидово расстояние. В работах показано, что оптимальным при использовании мягкого решения является квантование сигнала на 8-16 уровней.

Декодирование по максимуму правдоподобия является важнейшей и наиболее сложной алгоритмической проблемой в теории кодирования. Известно, что, к примеру, для двоичного симметричного канала связи и произвольных линейных кодов эта проблема в общем случае является NP-полной. Более того, она остается таковой даже в том случае, когда допускается сколь угодно долгая предобработка кода . Тем не менее, к настоящему моменту разработано и изучено множество общих подходов к решению данной задачи, позволяющих уменьшить асимптотическую

сложность декодирования по сравнению с переборным методом. Все эти алгоритмы имеют сложность, зависящую экспоненциально от длины кода, но в отличие от переборного метода - с меньшим показателем экспоненты. Кроме того, они вполне пригодны для практического применения в связке с кодами средней длины (до 200 символов в блоке).

К их числу, например, относится алгоритм декодирования по так называемым информационным совокупностям. Другим примером решения задачи ML-декодирования является так называемый алгоритм «соседей нуля» (англ. zeroneighbors), предложенный Левитиным и Хартманом. Данный алгоритм относится к семейству градиентоподобных алгоритмов декодирования. Еще одним представителем названного семейства является алгоритм ML-декодирования методом минимальных слов, изученный в работе.

Контрольные вопросы:

1. За счет чего увеличивается объем передаваемой информации в современных системах связи?
2. В чем смысл декодирования с мягким решением?
3. Какая форма является наиболее общей формой фазы амплитудной модуляции?
4. Чему равна n-мерная функция распределения?
5. При каком условии сигнал считается принятым верно?

Лекция №17. Применение помехоустойчивых кодов в телекоммуникационных системах.

Одним из основных способов обеспечение достоверности передачи информации по каналам связи является использование помехоустойчивых кодов, обнаруживающих и исправляющих ошибки.

Применение того или иного кода зависит типа ошибок. Например, код Хэмминга служит для исправления ошибок, код БЧХ для исправления многократных одиночных независимых ошибок, код Файра для одиночных пакетов ошибок и код РС для многократных пакетов ошибок.

Циклические широко применяются в различных системах, таких как система космической связи, системы цифрового телевидения, в цифровом радиовещании, в сотовой и транкинговой связи, в системах передачи данных. Эти коды также используются для устранения ошибок в полупроводниковых ЗУ, в накопителях на магнитных дисках и лазерных дисках, позволяющих обеспечить высокую надежность хранения информации. Например в технологии АТМ используются коды БЧХ (BCH – Bose – Chaudhuri - Hoquenghem), в цифровой транкинговой связи стандарта

APCO 25 кодов Хэмминга, кодов Рида-Соломона и кодов Голея, а также циклических кодов контроля чётности (CRC - CyclicRedundancyCheck) в радиоканалах подвижной связи GSM PLMN блочных и свёрточных кодов, в цифровом телевидении и при записи информации на магнитные диски кодов PC.

В системах ATM очень успешно используются циклические коды БЧХ для обнаружения и исправления ошибок в заголовке. Кроме кодов БЧХ также используются другие коды, использование которых зависит от типов ошибок в каналах связи и которые могут исправлять различные комбинации ошибок в зависимости от избыточности. Тип ошибок во многом зависит от способа передачи информации и от физической природы канала.

В технологиях ATM из-за ошибок в основном происходит потеря ошибок, который называется "эффектом размножения". При этом эффекте из-за ошибок в заголовке информационный пакет может быть доставлен не тому получателю. Для защиты заголовка ячейки ATM наиболее целесообразным является использование кодов БЧХ. Эти коды с большим выбором длины и с широким спектром возможностей по исправлению ошибок при ограниченном количестве набора значений n, k, t.

В ячейке ATM заголовок составляет 5 октетов. Под поле контроля ошибок отведено 8 бит. Этого вполне достаточно для исправления ошибок и обнаружения 89% многобитовых ошибок. Каждый передатчик ATM ячеек подсчитывает значение поля контроля ошибок в заголовке для первых четырёх октетов заголовка и заносит результат в пятый октет (в поле контроля ошибок в заголовке). Значение поля определяется как остаток от деления (по mod2) произведения x^8 на содержимое заголовка ячейки (без поля контроля заголовка) на производящий полином x^8+x^2+x+1. Оборудование передатчика подсчитывает этот остаток и прибавляет к нему по mod2 фиксированную комбинацию 01010101. Эта сумма и записывается в поле контроля ошибок заголовка. Все эти вышеуказанные операции реализуются оборудованием приёма ячеек ATM с помощью адаптивного механизма.

После запуска приёмник находится в режиме коррекции. Если обнаружена однобитовая ошибка, то ячейка стирается. В обоих случаях приёмник переключается в режим детектирования. В этом состоянии приёмника каждая ячейка с обнаруженной одиночной или множественной ошибкой в заголовке стирается. Если ошибок в заголовке не обнаружено, то механизм переходит в состояние коррекции.

В цифровой транкинговой связи стандарта APCO25 основными методами кодирования являются:

- блочное кодирование;
- решёточное кодирование;
- перемежение;

При блочном кодировании информации используются следующие виды корректирующих кодов:
- коды Хэмминга;
- коды Рида-Соломона;
- коды Голея;
- циклические коды контроля чётности (CRC-коды).

Коды Хэмминга используются при кодировании речевых сообщений (речевых кадров, синхрослова шифрования, слова управления каналом связи), коды Рида-Соломона и Голея для преамбулы и маркёра конца сообщения, а коды контроля чётности используются в основном для кодирования данных и формируются путём вычисления остатка от деления исходного информационного блока, представленного в виде полинома, на порождающий полином и сложения по mod2 с определённым инверсным полиномом. Например, для кодирования речевой информации используются 3 вида кодов Рида – Соломона с параметрами (36, 20, 17), (24, 16, 9) и (24, 12, 13). Все они являются укороченными и получаются из кода длиной 63 путей вычеркивания левых наиболее информативных символов.

Блочное кодирование в APCO25 является систематическим, т.е. первые k символов кодового слова представляют собой повторение информационного блока, а последние (n - k) символов являются проверочными.

В стандарте APCO25 используются три разновидности кода Голея:
-стандартный код Голея с параметрами (23, 12, 7);
-расширенный(24, 12, 8);
-укороченный (18, 6, 8).

Стандартный код Голея генерируется порождающим полиномом:

$$G(x)=x^{11}+x^{10}+x^{6}+x^{5}+x^{4}+x^{3}+x^{2}+1,$$

который при записи в восмиричном виде можно представить числом 6165.

Расширенный код Голея (24, 12, 8) образуется добавлением к стандартному одного бита контроля чётности. Укороченный код Голея (18, 6, 8) получается вычислением левых наибольших шести битов из расширенного кода.

В радиоканалах подвижной связи GSM PLMN используется свёрточное и блочное кодирование с перемежением. Перемежение обеспечивает преобразование пакетов ошибок в одиночные. Свёрточное кодирование является мощным свойством борьбы с одиночными ошибками, а блочное кодирование используется для обнаружения

нескорректированных ошибок.

Блочный код (n, k, t) преобразует k информационных символов путём добавления символов чётности (n-k), а также корректировать t ошибочных символов.

Одной из основных характеристик свёрточного кодирования является величина k, которая называется длиной кодового ограничения, и показывает, на какое максимальное число выходных символов влияет данный информационный символ. Так как сложность декодирования свёрточных кодов по наиболее выгодному, с точки зрения реализации, алгоритму Витерби возрастает экспоненциально с увеличением длины кодового ограничения, то типовые значения k малы и лежат в интервале 3^{-10}. Другой недостаток СК заключается в том, что они не могут обнаруживать ошибок. Поэтому в стандарте GSM для внешнего обнаружения ошибок используется блочный код на основе свёрточного кода (2, 1, 5). Наибольший выигрыш обеспечивают СК только при одиночных (случайных) ошибках в канале. В канале с замираниями, что имеет место в GSM PLMN, необходимо использовать СК совместно с перемежением.

В подвижной связи стандарта GSM используются следующие помехоустойчивые коды:

- циклический код (53, 50), с кодовым расстоянием $d_0 = 3$;
- сверточный код (2, 1);
- перемежение (в речевом режиме).
- код Файера $(x^{23} + 1)(x^{17} + x^9 + 1)$, k = 184; r = 40;
- сверточный код (2, 1).

Первые три кода имеют вероятность не обнаружения ошибок порядка 10^{-3} . Четвертый и пятый коды используется для передачи данных.

В системах CDMA используется следующие коды:

- сверточный код;
- каскадное кодирование;
- код Рида – Соломона → перемежение → сверточный код;
- Турбокодирование;
- специальное кодирование.

Сверточные коды используется для кодирования речи. Второй и третий коды для кодирования данных, четвертый и пятый коды используется для передачи данных.

Кодирование речи имеет ряд принципиальных особенностей: необходимо обеспечить интерактивную связь в режиме реального времени, при которой задержка, связанная с обработкой информации, не должна превышать допустимой величины.

Для этого на первом этапе производится декорреляция пакетов ошибок, в результате которой они преобразовываются в одиночные ошибки. На втором этапе сигнал обрабатывается с помощью классических методов борьбы со случайными ошибками, что приводит к их полному подавлению. Для борьбы с замираниями и возникновением, связанных с ними, пакетов ошибки служит процедура перемежения, которая состоит в перестановке символов кодируемой последовательности её модуляции и восстановлении исходной последовательности после демодуляции. Данная операция не вносит избыточности, а только изменяет порядок следования импульсов. Чем больше глубина перемежения, то есть максимальное расстояние, на которое разносятся соседние символы входящей последовательности, тем больше задержка.

Код БЧХ (63,44), используемый в системе спутникового цифрового радиовещания, позволяет исправить две или три ошибки, обнаружить и замаскировать 5 или 4 ошибки на каждый кодовый блок из 63 символов.

Повышение производительности вычислительных систем существенно увеличило объем хранимой и передаваемой информации. Недопустимость ошибок, а в ряде случаев сама природа данных, требует использование, как оборудования, так и программных процедур обнаружения и исправления ошибок обмена.

В системах цифровой записи и воспроизведения на компакт-дисках и магнитных лентах основную долю ошибок составляют ошибки типа «лотерея пакета». Для уменьшения влияния пакетов ошибок на качество записи и воспроизведения используют помехоустойчивое каскадное кодирование, являющееся эффективным средством борьбы с такими ошибками. Для того чтобы можно было обнаружить и исправить ошибки даже в самых неблагоприятных ситуациях, характеризующая способность кодов даже обнаружение и исправление ошибок выбирается с большим запасом. Кроме того, в цифровых магнитофонах записываемый поток имеет блочную структуру. Такая организация (формат) записи требует применения помехоустойчивых кодов блочной структуры.

В системах записи- воспроизведения на подвижные носители ошибки считывания обуславливаются царапинами и другими дефектами носителя, в устройствах полупроводниковой памяти замыканиями и отрывами межкомпонентных соединений и т.к.

В системах записи воспроизведения информации (как правило, двоичной) роль канала играет носитель информации: магнитная лента, диск, грампластинка, полупроводниковые запоминающие устройства.

В цифровых магнитофонах формата Prodigi (от англ. Professesional Digital), R – DAT (Rotary head Digital Audio Tape), S – DAT (Stationary head

Digital Audio Tape) используютсякодыРида – Соломона: двойнойкодС1(32,28) иС2(36,26), (40,32) и (29, 27).

Контрольные вопросы:

1. Какие коды используются в системе АТМ?
2. Какие коды используются при блочном кодировании?
3. При кодировании каких кодов используется Код Хэмминга?
4. Какие три разновидности кода Голея вы знаете?
5. В каких системах используется код БЧХ?

Лекция №18. Решетчатая кодовая модуляция. Многоуровневая кодовая модуляция.

Решетчатая кодовая модуляция (TCM)

Главная идея TCM, предложенная Унгербёком в 1976, состоит в том, чтобы реализовать *отображение через разбиения (декомпозицию) множества (сигнальных точек).* Для этого выбирается базовая структура решетки, ассоциированная с переходами на состояниях *конечного автомата,* и подмножества сигналов отображаются на ребра решетки. В системах, требующих высокой спектральной эффективности, допускается присваивание информационных (не кодированных) битов параллельным ребрам решетки.

Разбиение множества точек и отображение на решетку

Метки, приписываемые сигнальным точкам, определяются с помощью разбиения (декомпозиции) сигнального созвездия. На множестве2^{v}модуляционных точекSприменяется схемавложенных разбиений (древовидная декомпозиция) поvуровням. На i-ом уровне разбиения, $1 \le i \le v$,подмножество сигналов разбивается на два подмножества: $S_1(0), S_1(1)$, если$i=1$, и$S_1(b_i...b_{i-1}0)$ и $5S_i(b_i...b_{i-1}1)$,$i>1$, так, чтобы $расстояние d_i^2$между точками в каждом подмножестве было максимальным. *Битовый разряд* метки$b_i \in \{0,1\}$ ассоциируется с выбором подмножества,$S_j(b_1...b_{i-1}\ b_i)$на i-ом уровне разбиения. Этот процесс разбиения завершается полной *нумерацией* всех сигнальных точек. Каждая сигнальная точка получает свою (уникальную) метку (номер) изvбит$b_1b_2…b_v$, обозначаемую в дальнейшем $s(b_1b_2...b_v)$. Описанная процедура реализует *стандартное разбиение (по Унгербёку)* созвездия точек 2^{v}- ной модуляции.

При таком разбиении внутренние расстояния в подмножествах образуют неубывающую последовательность $\partial_1^2 \le \partial_2^2 \le \partial_v^2$. Результат соответствует *естественной нумерации* точек Л/-ФМ модуляции, т.е. двоичному представлению целых чисел, величина которых возрастает с переходом по часовой стрелке (или по счетчику). На рисунке 1 показана естественная нумерация точек 8-ФМ, дающая в результате $\partial_1^2 = 0{,}586$, $\partial_2^2 = 2$, $\partial_3^2 = 4$.

Унгербёк рассматривал кодер «как конечный автомат с заданным числом состояний и заданными переходами на множестве состояний». Он предложил несколько практических правил отображения подмножеств сигналов и точек на ребра решетки. Эти правила сводятся к следующим:

1. все подмножества должны появляться на решетке с одинаковой вероятностью.
2. входящие и исходящие переходы одного и того же состояния, должны быть приписаны подмножествам, находящимся на наибольшем Евклидовом расстоянии.
3. параллельные переходы присваиваются сигнальным точкам, разделенным наибольшим Евклидовым расстоянием (высшие уровни разбиения).

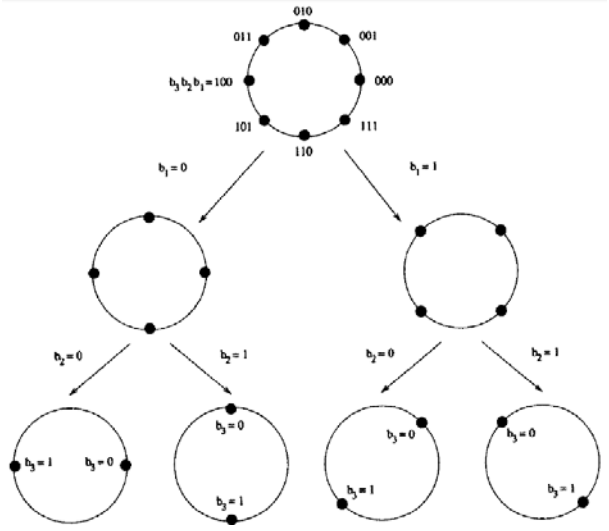


Рис.1. Естественное отображение для 8-ФМ созвездия.

Общая структура ТСМ кодера показана на рисунке 2. В общем случае решетчатой кодовой модуляции скорости *(v— 1)/v*структура решетки определяется сверточным кодом скорости*(k+1)/k.* Информационные символы, которые не кодируются, соответствуют *параллельным ребрам* на решетке.

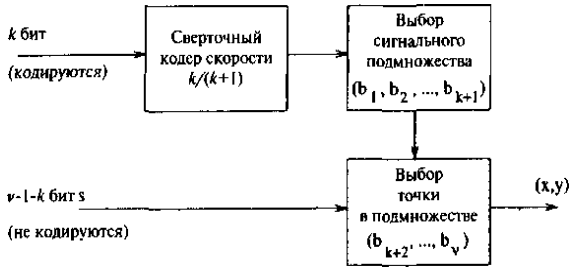


Рис. 2. Структура TCM кодера скорости (v—*1)/v*.

Декодирование по максимуму правдоподобия

Для выбора наиболее правдоподобных TCM последовательностей можно использовать алгоритм Витерби при условии, что генератор (вычислитель) реберных меток учитывает параллельные ребра. Кроме того, должен быть изменен блок выбора лучшего ребра и выживших не кодированных символов. Память выживших путей (или память обратного прохода) должна включать*(v— к* — 1) не кодированных двоичных символов в отличие от одного только бита в случае сверточного кода скорости 1/n. Важно заметить, что в случае 2^v-ичной ФМ (2^v-PSK) или 2^y-ичной КАМ (2^v-QAM)*корреляционные метрики* для *двумерных символов* имеют вид$x_p x_r$ $+y_p y_r$ где (x_p, y_p) представляет эталон сигнальной точки в созвездии, а $(x_r\ y_r)$ является принятой точкой.

Расстояние и вероятность ошибки

Помехоустойчивость TCM последовательностей можно анализировать так же, как для сверточных кодов. Это означает, что из диаграммы состояний TCM кодера может быть получен нумератор спектра весов. Единственная разница состоит в том, что теперь степени будут не целыми числами (соответствующими расстоянию Хемминга), а вещественными (соответственно расстоянию Евклида). Необходимо аккуратно учитывать факт наличия параллельных переходов на диаграмме состояний. Последнее означает, что модифицированная диаграмма состояний содержит два члена.

Многоуровневая кодовая модуляция (MCM)

В многоуровневой схеме кодирования, предложенной Имаи-Хирокава для множества 2V сигнальных точек созвездия используется v уровневая схема вложенных разбиений на два подмножества. Элементы кодовых слов v двоичных компонентных кодов C;, 1 < / < v, используются для индексации (нумерации) смежных классов на каждом уровне разбиения. Одним из преимуществ MCM конструкций является гибкость в согласовании Евклидовых расстояний на подмножествах сигнальных точек, д], i = 1, 2,

...,v, на каждом уровне разбиения с расстояниями Хемминга компонентных кодов.Уочмэн с соавторами предложил несколько правил конструирования, основанных на соображениях пропускной способности (применил цепное неравенство для взаимной информации). Более того, как показано в работах, многоуровневые коды с длинными компонентными кодами, такими как турбо коды или коды с низкой плотностью проверок, достигают пропускной способности канала.

Полезно так же отметить, что при выборе двоичных компонентных кодов разбиение на подмножества является дихотомическим, однако в общем случае компонентные коды могут быть выбраны над любым конечным полем соответственно схеме разбиения сигнального множества. Другим важным преимуществом многоуровневого кодирования является то, что декодирование (двоичного кода) может выполняться независимо на каждом уровне. Такое многоуровневое декодирование позволяет существенно снизить сложность по сравнению с оптимальным декодированием всего кода.

Конструкции и многоуровневое декодирование.

Обозначим C_i, $1 \le i \le v$, двоичный линейный блоковый $(n, k_i\ d_i)$ код. Обозначим $v = (v_{i1}, v_{i2}, ..., v_{in})$ кодовое слово кода C,. Рассмотрим код, образованный чередованием позиций подкодов, лг($|C_1|C_2|...|C_v|$), с кодовым словом вида

$v = (v_{11}v_{21}---v_{Vl}\ v_{12}v_{22}.--v_{v2}---v_{1n},v_{2n}---v_{vn})$

Каждый блок из v компонент вектора v является меткой (номером) сигнала на множестве 2^V модуляционных точек S. Тогда

$s(v) = (s(v_{11}\ v_{21} \bullet \bullet \bullet v_{v1}), i(v_{12}\ v_{22} \bullet \bullet \bullet v_{v2}), \bullet \bullet \bullet, s(v_{ln}\ v_{2n}, \bullet \bullet \bullet\ v_{vn}))$

является последовательностью сигнальных точек в S. Последовательности сигналов из множества S вида

$A = \{s(v): v \in \pi(|C_1|C_2|-|C_v|)\}$

образуют v уровневый модуляционный код над сигнальным множеством S или v уровневую конструкцию кодовой модуляции на множестве НУ сигналов. Такое же определение справедливо и для сверточных компонентных кодов.

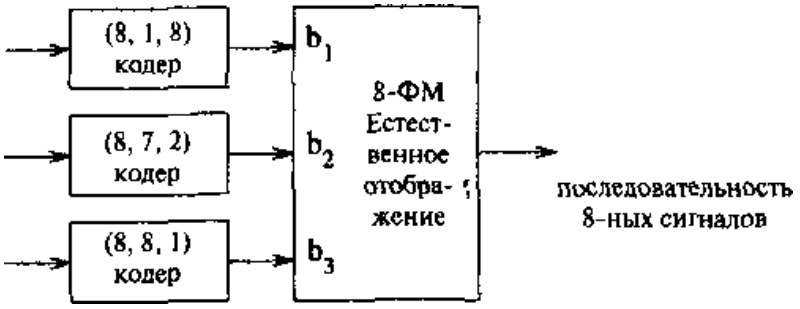


Рис. 5. Пример МСМ системы на сигналах 8-ФМ.

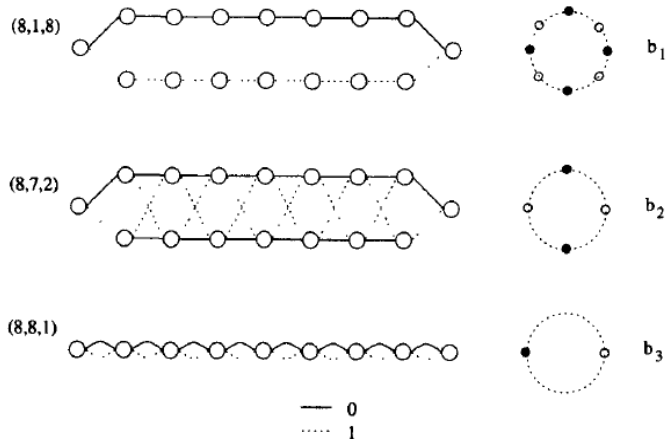
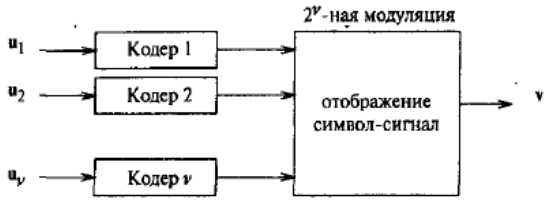


Рис. 6. Решетки компонентных кодов в примере МСМ на сигналах 8-ФМ.

Скорость, или спектральная эффективность, этой многоуровневой конструкции равна R = $(k_1+k_2+...+k_v)/n$(бит/сим- вол). Квадрат минимального Евклидова расстояния этой системы $D_c^{\ 2}$ (А) удовлетворяет оценке

(9.6) $$D_C^2(\Lambda) \geq \min_{1 \leq i \leq v} \{d_i \delta_i^2\}$$

Как уже упоминалось, одним из преимуществ многоуровневого кодирования является возможность применения многоступенчатого декодирования. На Рисунках 7 (а) и (б) показаны основные структуры, используемые для кодирования и декодирования многоуровневых кодов. Многоступенчатое декодирование приводит к снижению сложности (измеряемой, например, числом ребер на решетке декодирования) по сравнению с декодированием по максимуму правдоподобия (реализуемым, например, алгоритмом Витерби на полной решетке многоуровневого кода). Однако при многоступенчатом декодировании декодеры ранних уровней считают, что на старших уровнях кодирование не применяется. Это приводит к увеличению количества кодовых слов на минимальном расстоянии. Иначе говоря, увеличивается коэффициент ошибок или число ближайших соседей. Величина связанных с этим эффектом потерь зависит от выбора компонентных кодов и отображения символов на сигналы. В диапазоне вероятности ошибки порядка 10~2 ~ Ю-5 эта величина может достигать нескольких дБ.



(а) Многоуровневое кодирование

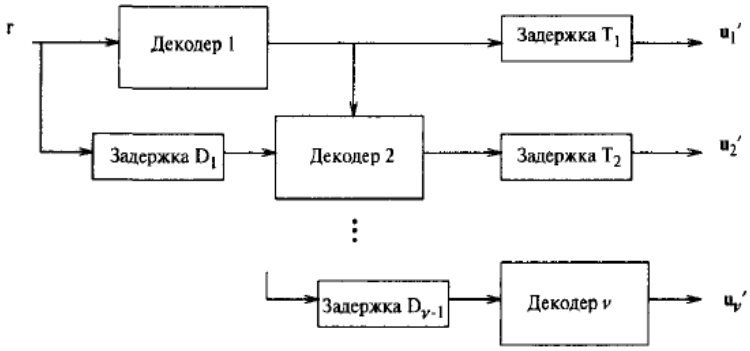


Рис. 7. Базовая структура кодера и декодера для систем многоуровневой кодовой модуляции.

Для кодов средней и большой длины предельная эффективность многоуровневой кодовой модуляции может достигаться с помощью гибридного подхода, когда на первых уровнях используются мощные турбо коды, а на остальных уровнях используются двоичные коды с жестким декодированием. Такие комбинации могут достигать очень высокой помехоустойчивости.

Неравная защита в системахмногоуровневой кодовой модуляции

Многоуровневая кодовая модуляция является удобной схемой создания неравной защиты от ошибок (UEP - unequal-error- protection), так как она обладает необходимой гибкостью в конструировании минимальных Евклидовых расстояний между кодовыми последовательностями на каждом уровне разбиений. Однако следует очень внимательно выбирать отображение между символами и сигналами с тем, чтобы не разрушить искомую способность к неравной защите. При смешанном разбиении некоторые уровни разбиения являются нестандартными, тогда как другиевыполняются поправилам, предложенным Унгербёком. Этим способом достигается хороший обмен между коэффициентами ошибок и Евклидовыми расстояниями по уровням конструкции. Для достижения неравной зашиты расстояния Евклида по уровням разбиения выбираются следующим образом

(1) $$d_1\delta_1^2 \geq d_2\delta_2^2 \geq \cdots \geq d_\nu\delta_\nu^2$$

Для 1≤i≤v обозначим v,(u;) кодовое слово кода C,, соответствующее информационному вектору и, размерности к, бит. Обозначим s = s(u) и s' = s(u') последовательности 2^v-ных сигналов для информационных векторов u = (u_1, u_2,..., u_v) и и' = (u'_1, $и'_2$, ..., u'_v), соответственно. Евклидово разделение [YI] между кодовыми последовательностями на г'-ом уровне декомпозиции для i = 1,2, ..., v определено как

s, =min{d(s,s'):$u_1 \neq u'_1$, $u_j = u'_j$,j<i} (2)

где $s_1 = d_1\delta_1^2$, $s_2 = d_2\delta_2^2$, …, $s_v = d_v\delta_v^2$. В канале с АБГШ система неравенств (1) обеспечивает снижение уровня защиты от ошибок для компонентных сообщений меньшего уровня.

Контрольные вопросы:

1. Опишите общую структуру ТСМ кодера?
2. Какие правила отображения подмножеств вывел Унгербёк?
3. Объяснитесмысл МСМ?
4. Что понимается под неравной защитой в МСМ

СПИСОК ЛИТЕРАТУРЫ:

1	Джураев Р.Х., Джаббаров Ш.Ю., Умирзаков Б.М., Хамраев Э.А Помехоустойчивые коды в телекоммуникационных системах. Учеб. пособие-ТУИТ, Ташкент 2013.
2	Р.Х. Джураев, Ш.Ю. Джаббаров, С.О.Махмудов «Теория информации и кодирования». Электронный конспект лекций/ТУИТ, с.128. Ташкент, 2015
3	Tracey Ho Network Coding: Introduction. Cambridge University Press, 2008
4	Abbas El Gamal, Young-Han Kim Network Information Theory. Cambridge University Press, 2011
5	Думачев В.Н. Теория информации и кодирования. Воронеж: Воронежский институт МВД России, 2012. – 200 с.
6	Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение – ТЕХНОСФЕРА – Москва, 2005.
7	Вернер М. Основы кодирования. Учебник для ВУЗов. ТЕХНОСФЕРА – Москва, 2006.
8	Блейхут Р. Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986.
9	Варгаузин В. Помехоустойчивое кодирование в пакетных сетях. Телемультимедия, 2005.
10	Золотарев В. и др. Помехоустойчивое кодирование. Методы и алгоритмы. М.: Горячая линия – Телеком, 2004.
11	Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи. М.: Радио и связь, 1987.
12	Плохов Е.М. Теория информации и кодирование. Учеб. пособие. Феникс, 2002.
13	Кудряшов Б.Д. Теория информации. – С.-Пб.: Питер, 2009. – 320 с
14	Котоусов А.С. Теория информации. – М.: Радио и связь, 2003. – 80 с