Министерство образования и науки Российской Федерации
САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

А.И. Конюхов

Основы теории информации и кодирования.

Учебное пособие по курсу "Теория информации и кодирования".
Рекомендовано УМ физического факультета СГУ в качестве учебного пособия для студентов, обучающихся по направлению 03.03.03 «Радиофизика» и 03.03.02 «Физика».

Оглавление

Введение

В настоящем учебном пособии рассматриваются основы теории информации, основы теории кодирования, связанной с передачей информации по каналам связи. Данное пособие служит теоретическим базисом для специализированных курсов, связанных с обработкой информации. Содержание пособия соответствует программе курса «Теория информации и кодирования» для студентов высших учебных заведений, обучающихся по специальностям направления 03.03.03 «Радиофизика» и 03.03.02 «Физика». Пособие состоит из четырёх теоретических глав. В первой главе содержатся общие сведения по основам теории информации. Во второй главе рассматриваются методы сжатия информации. Третья глава посвящена краткому изложению теории корректирующего кодирования в системах передачи информации по каналам связи и в четвёртой главе рассматривается помехоустойчивое кодирование. Приведены примеры использования рекуррентных кодов.

Изложение теоретических глав написано в предположении, что студенты знакомы с основными понятиями из курса высшей математики. Студенты должны иметь навыки самостоятельной работы с учебными пособиями и монографической учебной литературой, умение решать физические и математические задачи, требующие применения дифференциального и интегрального математического аппарата, умение производить приближенные преобразования аналитических выражений. Данное пособие знакомит студентов с базовыми понятиями теории информации и кодирования (информационными характеристиками дискретных и непрерывных источников сообщений и основными принципами хранения информации и передачи информационных сообщений по каналам связи). При подготовке пособия использованы теоретические положения, отраженные в научных статьях, в учебной литературе и монографиях.

1. Базовые понятия теории информации.

В чисто технических автоматических и автоматизированных системах информация чаще всего используется для выработки управляющих воздействий. При обращении информации в системах можно выделить отдельные этапы. Так как материальным носителем информации является сигнал, то реально это будут этапы обращения и преобразования сигналов. Информацию можно рассматривать как сведения (знания), полученные в результате моделирования (описания) реального мира или его исследуемой части, являющиеся объектом некоторых операций: передачи, распределения, преобразования, хранения или непосредственного использования. Дальнейшее изложение требует определение некоторых устоявшихся понятий и терминов теории информации, которых необходимо строго придерживаться в дальнейшем. Информация содержится в сообщениях, которые генерируются источниками сообщений. Источник сообщений – любой процесс, объект или явление, который обладают способностью изменять свое состояние во времени или в пространстве. Символ источника сообщений – это любое мгновенное состояние источника сообщений. Сообщение – любая конечная последовательность символов. Алфавит источника сообщений – все множество различных символов, генерируемых источником сообщений. Объем алфавита источника сообщений – число различных символов, генерируемых источником сообщений. Дискретный источник сообщений – источник сообщений, обладающий конечным алфавитом. Непрерывный источник сообщений – источник сообщений, обладающий бесконечным алфавитом.

Для практического использования понятия информации необходимо научиться ее измерять. Для количественного измерения информации вводится такое понятие как энтропия. Энтропия – широко используемый в естественных и точных науках термин. Впервые введён в рамках термодинамики как функция состояния термодинамической системы, определяющая меру необратимого рассеивания энергии. В статистической физике энтропия характеризует вероятность осуществления какого-либо макроскопического состояния. Кроме физики, термин широко употребляется в математике: теории информации и математической статистике. Для энтропии (чаще в математике) встречается также название шенноновская информация или количество информации по Шеннону. Термин «энтропия» заимствован из термодинамики, где она характеризует среднюю неопределенность состояния системы молекул вещества. В теории информации этот термин введен в 1948 г. американским ученым К. Шенноном и далее более строго определен советскими математиками А.Я. Хинчиным и А.Н. Колмогоровым [1–5]. Рассмотрим энтропию для дискретного и непрерывного источника сообщений.

1.1 Дискретный источник сообщений.

Предположим, что источник сообщений может в каждый момент времени случайным образом принять одно из конечного множества возможных состояний. Такой источник называют дискретным источником сообщений. При этом принято говорить, что различные состояния реализуются вследствие выбора их источника.

Каждому состоянию источника U ставиться в соответствие условное обозначение в виде знака. Совокупность знаков $u_1, u_2, \ldots, u_i, \ldots, u_N$ соответствующих всем N возможным состояниям источника называют его алфавитом, а количество состояний N объемом алфавита. Формирование таким источником сообщений сводиться к выбору им некоторого состояния u_i и выдачи соответствующего знака. Таким образом, под *элементарным дискретным сообщением* будем понимать символ u_i выдаваемое источником, при этом в течение некоторого времени T источник может выдать дискретное сообщение в виде последовательности элементарных дискретных сообщений, представляющей сбой набор символов u_i (например, u_5, u_1, u_3) каждый из которых имеет длительность t_i секунд. В общем случае необязательно одинаковую для различных i. Такая модель источника сообщений соответствует реальной ситуации имеющей место в телеграфии (t_i=*const*) и передаче данных (t_i=*const*).

В каждом элементарном сообщении содержится для его получателя определенная информация. Определяя количественную меру этой информации, мы совершенно не будем учитывать ее смыслового содержания, так же ее значения для конкретного получателя. Очевидно, что при отсутствии сведений о состоянии источника имеется неопределенность относительно того, какое сообщение u_i из числа возможных им выбрано, а при наличии этих сведений данная неопределенность полностью исчезает. Естественно количество информации содержащейся в дискретном сообщении измерять величиной исчезнувшей неопределенности.

Меру этой неопределенности можно рассматривать и как *меру* количественной информации. Мера должна удовлетворять ряду естественных условий, одним из них является необходимость ее монотонного возрастания с увеличением возможности выбора, т.е. объема алфавита источника N. Кроме того, желательно, чтобы вводимая мера обладала свойством аддитивности заключающемся в следующем: если 2 независимых источника с объемами алфавита N и M рассматривать как один источник, одновременно реализующий пары состояний n_i и m_i то в соответствии с принципом аддитивности полагают, что неопределенность объединенного источника равна сумме неопределенностей исходных источников.

1.2 Энтропия источника сообщений.

В более общем случае, когда вероятности различных состояний источника не одинаковы степень неопределенности конкретного состояния зависит не только от объема алфавита источника, но и от вероятности этого состояния. В такой ситуации количество информации, содержащееся в одном дискретном сообщении u_k целесообразно определить как функцию вероятности появления этого сообщения $P(u_k)$ [6,7].

Количество информации в сообщении тем больше, чем оно более неожиданно. Если источник выдает последовательность зависимых между собой элементарных сообщений, то наличие предшествующих сообщений может изменить вероятность последующего а, следовательно, и количество информации в нем.

Количество информации является случайной величиной, поскольку сами сообщения являются случайными. Его распределение вероятностей определяется распределением вероятностей сообщений в данном ансамбле для цифровой характеристики всего ансамбля или источника сообщения используется математическое ожидание количества информации в отдельных сообщениях называемое энтропией [6].

$$H(U) = M\left\{\log\frac{1}{p(u_1)}\right\} = \sum_{i=1}^{M} P(u_1)\cdot\log\left(\frac{1}{P(u_1)}\right)$$

Чем больше энтропия источника, тем больше степень неожиданности выдаваемых им сообщений в среднем, т.е. тем более неопределенным является ожидание сообщений. Впервые эта мера была предложена Клодом Шенноном. Предполагающая мера была названа энтропией не случайно. Дело в том, что вид формулы совпадает с полученным ранее результатом Вольцманом выражением для энтропии термодинамической системы.

Из формулы Шеннона следуют важные выводы:

1. Увеличение меры Шеннона свидетельствует об уменьшении энтропии (увеличении порядка) системы;
2. Уменьшение меры Шеннона свидетельствует об увеличении энтропии (увеличении беспорядка) системы.

Положительная сторона формулы Шеннона – ее отвлеченность от смысла информации. Кроме того, в отличие от формулы Хартли, она учитывает различность состояний, что делает ее пригодной для практических вычислений. Основная отрицательная сторона формулы Шеннона – она не распознает различные состояния системы с одинаковой вероятностью.

1.3 Свойства энтропии.

1) Энтропия любого дискретного ансамбля не отрицательна

$$H(U) \geq 0$$

Равенство нулю возможно лишь в том случае, когда источник генерирует одно единственное сообщение с вероятностью *P=1* в этом случае вероятности других сообщений равны нулю. Не отрицательность следует из того, что количество информации в каждом из возможных сообщений источника не отрицательно.

2) Максимально возможное значение энтропии дискретного источника с объемом алфавита N равно $\log N$ и достигается в том случае, когда все его сообщения равновероятны.

3) Энтропия объединения нескольких независимых статистических источников сообщений равна сумме энтропии исходных источников - свойство аддитивности энтропии. Не теряя общности, ограничимся рассмотрением объединения u и z с объемами алфавита соответственно N и M. Под объединением двух источников u и z понимают обобщенный источник сообщений $(u \cdot z)$ характеризующейся совместными $P(u_i z_j)$ всех возможных комбинаций, состояния u_i - источника u, z_i - источника z.

Энтропия обобщенного источника будет равна:

$$H(U \cdot Z) = \sum_{i=1}^{N} \sum_{j=1}^{M} P(u_i, z_j) \log \frac{1}{P(u_i, z_j)} = I(Y, X)$$

Энтропия источника независимых сообщений. До сих пор определялось количество информации, содержащееся в отдельных сообщениях. Вместе с тем во многих случаях, когда требуется согласовать канал с источником сообщений, таких сведений оказывается недостаточно. Возникает потребность. в характеристиках, которые, бы позволяли оценивать информационные свойства источника сообщений в целом. Одной из важных характеристик такого рода является среднее количество информации, приходящееся на одно сообщение.

Зная взаимную информацию, связанную с парой событий (x_i, y_i), которые являются возможной реализацией двух случайных величин X и Y, мы можем получить среднее значение взаимной информации простым взвешиванием $I(x_i, y_i)$ с вероятностью появления этой пары и суммированием по всем возможным событиям. Таким образом получим

$$I(X,Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_i)\log\frac{P(x_i, y_i)}{P(x_i)P(y_j)} = I(Y,X)$$

как среднюю взаимную информацию между X и Y. Видно, что $I(X,Y) = 0$, когда X и Y статистически независимы и $P(x_i, y_j) = P(x_i)P(y_i)$. Важным свойством средней взаимной информации является то, что $I(X,Y) \geq 0$. Аналогично определим среднюю собственную информацию, обозначенную $H(X)$:

$$H(X) = \sum_{i=1}^{n} p(x_i)\, I(x_i) = -\sum_{i=1}^{m} P(x_i) \cdot P(x_i)$$

Если X представляет собой алфавит возможных символов источника, $H(X)$ представляет среднюю собственную информацию на символ источника, и её называют энтропией источника. В частном случае, когда символы источника равновероятны, $P(x_i) = 1/n$ для всех i, и, следовательно,

$$H(X) = -\sum_{i=1}^{n}\frac{1}{n}\, log\frac{1}{n} = \log n$$

В общем случае $H(X) \leq \log n$ при любых заданных вероятностях символов источника. Другими словами, энтропия источника максимальна, когда выходные символы равновероятны.

Рассмотрим двоичный источник, который выдаёт последовательности независимых символов, причём выходной кодовый символ «0» с вероятностью q, а символ «1» с вероятностью $1 - q$. Энтропия такого источника

$$H(X) = H(q) = -q\, log_q - (1-q)$$

Функция $H(q)$ показана на рис.1. Видно, что максимальное значение функции энтропии имеет место при $q = 1/2$, причём $H(1/2) = 1$ бит. Среднее значение условной собственной информации называется условной энтропией и определяется как

$$I(X|Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_i)\log\frac{1}{P(x_i|y_j)}$$

Мы интерпретируем $I(X|Y)$ как неопределённость X (дополнительную информацию, содержащуюся в X) после наблюдения Y. Комбинации вышеуказанных формул дают соотношение:

$$I(X,Y) = H(X) - H\ (X|Y) = H(Y) - H(Y|X)$$

Из условия $I(X,Y) \geq 1$ следует, что $H(X) \geq H(X|Y)$ и $H(Y) \geq H(Y|X)$, причём равенство имеет место тогда, и только тогда, когда X и Y статистически независимы. Если мы интерпретируем $H(Y|X)$ как среднее значение неопределённости (условной собственной информации) X после наблюдения Y и $H(x)$ как среднее значение априорной неопределённости (собственной информации), т.е. имевшейся до наблюдения, тогда $I(X,Y)$ определяет взаимную информацию (уменьшение среднего значения неопределённости, имеющейся относительно X после наблюдения Y). Так как $H(X) \geq H(X|Y)$, то ясно, что при условии наблюдения Y энтропия H(x) не увеличится.

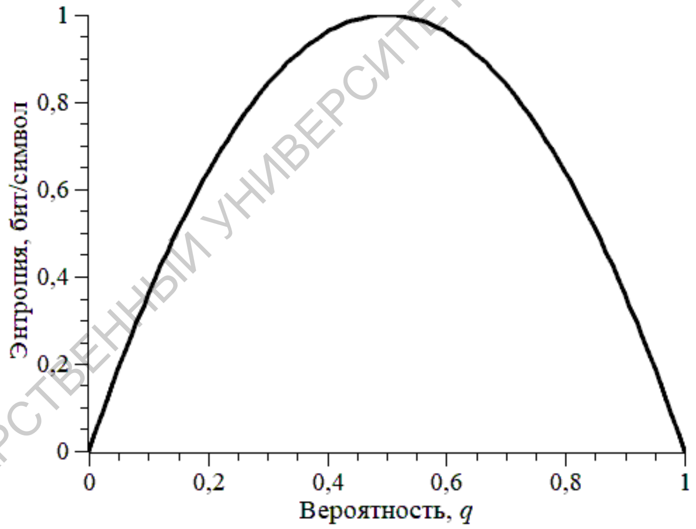

Рис.1 Энтропия двоичного источника.

1.4 *Энтропия дискретного источника сообщений.*

Для большинства реальных источников сообщения имеют разные **вероятности**. Например, в тексте буквы А, О, Е встречаются сравнительно часто, а Щ, Ы – редко. Согласно экспериментальным данным, для букв русского алфавита характерны безусловные вероятности, сведенные в табл. 1.

Таблица 1. Безусловные вероятности букв русского алфавита

буква	вероятность	буква	вероятность	буква	вероятность
пробел	0,175	М	0,026	Ч	0,012
О	0,090	Д	0,025	Й	0,010
Е	0,072	П	0,023	Х	0,009
А	0,062	У	0,021	Ж	0,007
И	0,062	Я	0,018	Ю	0,006
Т	0,053	Ы	0,016	Ш	0,006
Н	0,053	З	0,016	Ц	0,004
С	0,045	Ь,Ъ	0,014	Щ	0,003
Р	0,040	Б	0,014	Э	0,003
В	0,038	Г	0,013	Ф	0,002
Л	0,035	К	0,028		

При разных вероятностях сообщения несут различное количество $I(x_i)$ информации . При решении большинства практических задач необходимо знать среднее количество информации, приходящееся на один элемент сообщения. Это среднее количество информации при общем числе элементов сообщения источника n и числе символов алфавита m равно:

$$H(x) = \frac{I(x,n)}{n} = -\sum_{i=1}^{m} p(x_i) \cdot log_2 p(x_i) \quad \textit{(бит/сообщение)} \tag{1}$$

Физически энтропия $H(x)$ выражает среднюю неопределенность состояния источника сообщений и является объективной информационной характеристикой источника. Энтропия всегда положительна и принимает максимальное значение при равновероятных сообщениях:

$$H_{max}(X) = -\sum_{i=1}^{m} \frac{1}{m} \cdot log_2\left(\frac{1}{m}\right) = log_2(m) \tag{2}$$

Минимальное значение энтропии $H_{min}(X) = 0$ соответствует случаю, когда одна из вероятностей $p(x_i) = 1$, а остальные равны нулю, т.е. имеется полная определенность.

Для источника с зависимыми сообщениями энтропия тоже вычисляется как математическое ожидание количества информации на один элемент этих сообщений. Следует заметить, что полученное в этом случае значение энтропии будет меньше, чем для источника независимых сообщений. Это следует из того, что при наличии зависимости сообщений неопределенность выбора уменьшается и, соответственно, уменьшается энтропия. Так, в тексте после сочетания "чт" вероятнее всего, что третьей буквой будет "о" и маловероятно появление в качестве третьей буквы "ж" или "ь". В среднем, сочетание "что" несет меньше информации, чем эти буквы в отдельности.

Наиболее широкое применение в дискретных системах передачи информации получили двоичные источники. Двоичные источники характеризуются передачей только двух возможных сообщений. Причем, если вероятность передачи одного из них $p(x_1)$, то вероятность передачи другого $p(x_2) = 1 - p(x_1)$.

Определим энтропию двойного источника. Из формулы (1) получим:

$$H(X) = -\sum_{i=1}^{2} p(x_i) \cdot log_2 p(x_i) = p(x_1) \cdot log_2\, p(x_1) - p(x_2) \cdot log_2\, p(x_2)$$

$$= -p(x_1) \cdot log_2\, p(x_1) - [1 - p(x_1)]$$

График зависимости представлен на рис. ниже. Как следует из графика, энтропия двоичного источника изменяется в пределах от нуля до единицы. Энтропия равна нулю, когда вероятность передачи одного из символов равна нулю или единице, т.е. передается только одно сообщение. Получение же одного единственно возможного сообщения никакой новой информации не дает. Энтропия двоичного источника будет максимальна, если существует наибольшая неопределенность, т.е. $p(x_2) = p(x_1) = 0{,}5$. При этом $H(X) = log_2 m = 1$.

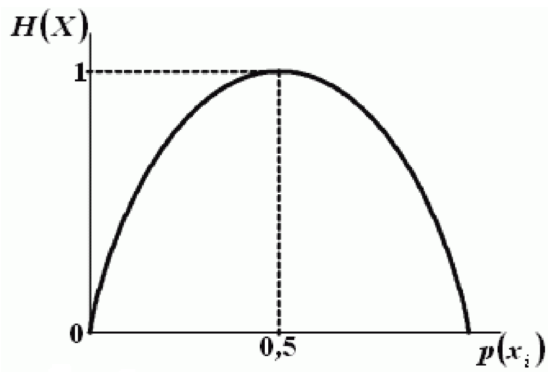

Рис.2. Зависимость энтропии от вероятности символов

1.5 Энтропия непрерывных сообщений.

В приведенных выше рассуждениях предполагалось, что все возможные значения (символы) непрерывного сообщения равновероятны, однако это не всегда справедливо. Как правило, символы непрерывного сообщения $x(t)$ обладают некоей плотностью распределения вероятности *p(x)*, которая характеризует вероятность попадания символов непрерывного сообщения $x(t)$ в интервал Δx, примыкающий к точке x. И, если плотностью распределения вероятности значений непрерывного сообщения *p(x)*, не постоянна, то и вероятность появления отдельных символов непрерывного сообщения различна. Следовательно, в соответствии с формулой (1.4), количество информации, которое может нести отдельный символ непрерывного сообщения, так же не постоянно и будет зависеть от значений непрерывного сообщения $x(t)$ Поэтому важно выяснить, как зависят информационные характеристики непрерывного сообщения от присущего ему закона распределения его символов.

Положим, что непрерывное сообщение обладает известной функцией плотности распределения вероятности его символов *p(x)*. Выберем интервал Δx_k. и обозначим середину этого интервала через x_k. Будем рассматривать все символы непрерывного сообщения, попадающие в интервал Δx_k , как k-тый символ дискретного сообщения. Вероятность появления этого символа дискретного сообщения (P_k) будет определяться выражением [8-10]:

$$P_k = P\left(x_k - \frac{\Delta x_k}{2} \le x \le x_k + \frac{\Delta x_k}{2}\right) = \int_{x_k - \frac{\Delta x_k}{2}}^{x_k + \frac{\Delta x_k}{2}} p(x)\,dx$$

Энтропию полученного таким образом дискретного сообщения ($H_Д$), на основании формулы Шеннона (1.8), можно записать в виде:

$$H_Д = -\sum_k P_k \log P_k = -\sum_k \int_{x_k - \frac{\Delta x_k}{2}}^{x_k + \frac{\Delta x_k}{2}} p(x)\,dx \cdot \log \int_{x_k - \frac{\Delta x_k}{2}}^{x_k + \frac{\Delta x_k}{2}} p(x)\,dx \qquad (3)$$

Предполагая, что функция *p(x)* вместе со своей производной непрерывная, имеем:

$$\int_{x_k - \frac{\Delta x_k}{2}}^{x_k + \frac{\Delta x_k}{2}} p(x)\,dx \approx p(x_k)\cdot \Delta x_k .$$

Исходное непрерывное сообщение можно рассматривать как предел сформированного дискретного сообщения при $\Delta x_k \to 0$ (т.е. при стремлении объёма алфавита сформированного дискретного сообщения к бесконечности). Поэтому, подставив приведенное выше выражения в (1.14) и перейдя к пределу, получим выражение для энтропии непрерывного сообщения:

$$H(X) = -\lim_{\Delta x_k \to 0} \left\{ \sum_k p(x_k)\Delta x_k \log\left[p(x_k)\Delta x_k\right] \right\}$$

Заменяя в этом выражении логарифм произведения суммой логарифмов сомножителей, преобразуем это выражение к виду

$$H(X) = -\lim_{\Delta x_k \to 0} \left\{ \sum_k p(x_k)\log p(x_k)\,\Delta x_k \right\} - \lim_{\Delta x_k \to 0} \left\{ \sum_k p(x_k)\Delta x_k \log \Delta x_k \right\} =$$

$$= -\int_{-\infty}^{+\infty} p(x)\log p(x)\,dx - \lim_{\Delta x_k \to 0} \sum_k p(x_k)\Delta x_k \log \Delta x_k \qquad (4)$$

Обозначим первое слагаемое в этом выражении как H_x, а второе - как H_Δ. Легко показать, что слагаемое H_Δ обращается в бесконечность при $\Delta x_k \to 0$ независимо от вида функции плотности распределения вероятности символов непрерывного сообщения ($p(x)$).

Действительно, если принять, что интервалы Δx_k одинаковы, т.е. $\Delta x_k = \Delta x$ для всех значений k, то для второго слагаемого в выражении (4) имеем:

$$H_\Delta = -\lim_{\Delta x \to 0} \left(\sum_k p(x_k)\Delta x \log \Delta x \right) = -\lim_{\Delta x \to 0} \left(\log \Delta x \sum_k p(x_k)\Delta x \right) =$$

$$= -\lim_{\Delta x \to 0} \left(\log \Delta x \sum_k p(x_k)\Delta x \right) = -\lim_{\Delta x \to 0} \log \Delta x \int_{-\infty}^{+\infty} p(x)\,dx$$

С учетом того, что $\int_{-\infty}^{+\infty} p(x)\,dx = 1$, получаем

$$H_\Delta = -\lim_{\Delta x \to 0} (\log \Delta x) = \infty$$

На основании выше изложенного можно сделать вывод: непрерывные сообщения не имеют абсолютной меры энтропии. Их полная энтропия равна бесконечности. На практике при определении различных информационных характеристик непрерывных сообщений, которое в реальных условиях всегда действует на фоне шума, приходится вычислять разность энтропий сообщения и шума. При этом бесконечно большие слагаемые H_Δ в энтропии сообщения и энтропии шума взаимно уничтожаются, так как они не зависят от законов распределения символов сообщения и шума. Таким образом, за меру энтропии непрерывного сообщения можно принять выражение

$$H_x = -\int_{-\infty}^{+\infty} p(x)\log p(x)\,dx.$$

Вычисленную таким образом энтропию называют *относительной* или *дифференциальной*, так как она определяет не полную энтропию непрерывного сообщения, равную, как было показано, бесконечности, а только ту ее часть (H_x), которая зависит от закона распределения символов непрерывного сообщения. И под энтропией непрерывного сообщения, как правило, понимают именно дифференциальную энтропию H_x.

1.6 Избыточность источника сообщений.

Избыточными в источнике являются сообщения, которые несут малое, иногда нулевое, количество информации. Наличие избыточности означает, что часть сообщений можно и не передавать по каналу связи, а восстановить на приеме по известным статистическим связям. Так и поступают при передаче телеграмм, исключая из текста союзы, предлоги, знаки препинания, поскольку они легко восстанавливаются по смыслу телеграммы на основании известных правил построения фраз.

Количественно избыточность оценивается коэффициентом избыточности:

$$\chi = \frac{H_{max}(X) - H(X)}{H_{max}(X)} = 1 - \frac{H(X)}{H_{max}(X)}$$

где $H(X)$ – энтропия источника; $H_{max}(X) = log_2 m$ – максимальная энтропия источника с алфавитом из m сообщений.

Избыточность при передаче сообщений имеет свои положительные и отрицательные стороны. Увеличение избыточности приводит к увеличению времени передачи сообщений, излишней загрузке каналов связи. За определенный промежуток времени по каналу передается меньшее количество информации, чем это возможно; поэтому одной из задач теории информации и техники кодирования является задача сокращения избыточности.

Однако при увеличении избыточности появляется возможность повышения помехоустойчивости передачи сообщений. Так, избыточность текста позволяет исправлять отдельные ошибки или восстанавливать пропущенные буквы или даже слова в телеграмме. У русского и всех европейских языков избыточность с учетом всех статистических зависимостей букв примерно одинакова $\chi = 0.5$. Она сформировалась в результате длительной, общественной практики на основе требований исправления искажения слов и фраз под воздействием различных мешающих факторов. Для систем связи устанавливается компромиссное значение избыточности, которое обеспечивает заданную скорость и надежность передачи сообщений.

2. Сжатие дискретной информации

2.1 Словарный метод кодирования

Практически все словарные методы кодирования принадлежат семье алгоритмов из работы двух израильских ученых - Зива и Лемпела, опубликованной в 1977 году [7, 15]. Сущность их состоит в том, что фразы в сжимаемом тексте заменяются указателем на то место, где они в этом тексте уже ранее появлялись. Это семейство алгоритмов называется методом Зива-Лемпела и обозначается как *LZ*-сжатие. Этот метод быстро приспосабливается к структуре текста и может кодировать короткие функциональные слова, так как они очень часто в нем появляются. Новые слова и фразы могут также формироваться из частей ранее встреченных слов. Декодирование сжатого текста осуществляется напрямую - происходит простая замена указателя готовой фразой из словаря, на которую тот указывает. На практике LZ-метод добивается хорошего сжатия, его важным свойством является очень быстрая работа декодера. (Когда мы говорим о тексте, то предполагаем, что кодированию подвергается некоторый вектор данных с конечным дискретным алфавитом, и это не обязательно текст в буквальном смысле этого слова.) Большинство словарных методов кодирования носят имя авторов идеи метода Зива и Лемпела, и часто считают, что все они используют один и тот же алгоритм кодирования. На самом деле разные представители этого семейства алгоритмов очень сильно различаются в деталях своей работы.

Все словарные методы кодирования можно разбить на две группы. Методы, принадлежащие к первой группе, находя в кодируемой последовательности цепочки символов, которые ранее уже встречались, вместо того чтобы повторять эти цепочки, заменяют их указателями на предыдущие повторения. Словарь в этой группе алгоритмов в неявном виде содержится в обрабатываемых данных, сохраняются лишь указатели на встречающиеся цепочки повторяющихся символов. Все методы этой группы базируются на алгоритме, разработанном и опубликованном, как уже отмечалось, сравнительно недавно - в 1977 году Абрахамом Лемпелем и Якобом Зивом, - LZ77. Наиболее совершенным представителем этой группы, включившим в себя все достижения, полученные в данном направлении, является алгоритм LZSS, опубликованный в 1982 году Сторером и Шимански. Процедура кодирования в соответствии с алгоритмами этой группы иллюстрируется рис. 3.

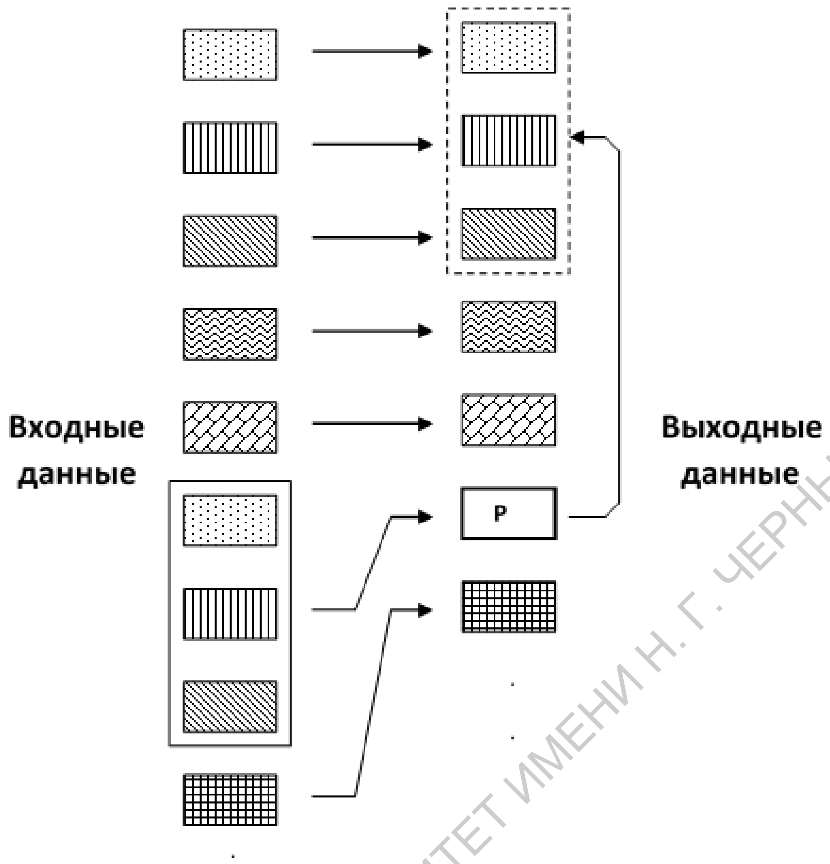

Рис.3. Процедура кодирования по алгоритму первой группы.

Алгоритмы *второй группы* в дополнение к исходному словарю источника создают словарь фраз, представляющих собой повторяющиеся комбинации символов исходного словаря, встречающиеся во входных данных. При этом размер словаря источника возрастает, и для его кодирования потребуется большее число бит, но значительная часть этого словаря будет представлять собой уже не отдельные буквы, а буквосочетания или целые слова. Когда кодер обнаруживает фразу, которая ранее уже встречалась, он заменяет ее индексом словаря, содержащим эту фразу. При этом длина кода индекса получается меньше или намного меньше длины кода фразы. Все методы этой группы базируются на алгоритме, разработанном и опубликованном Лемпелем и Зивом в 1978 году, - LZ78. Наиболее совершенным на данный момент представителем этой группы словарных методов является алгоритм LZW, разработанный в 1984 году Терри Вэлчем. Алгоритмы второй группы несколько проще в объяснении их работы, поэтому начнем рассмотрение принципа действия LZ-кодеров с алгоритма LZW. Рассмотрим в самом общем виде работу LZW-кодера и декодера.

2.2 Процедура кодирования LZW.

Процесс сжатия выглядит достаточно просто. Мы последовательно считываем символы входного потока (строку) и проверяем, есть ли в уже созданной нами таблице такая строка. Если строка есть, то считываем следующий символ, а если такой строки нет, - заносим в выходной поток код для предыдущей найденной строки, заносим ее в таблицу и начинаем поиск снова. Пусть на вход кодера поступает последовательность символов вида /WED/WE/WEE/WEB, при этом размер алфавита входных символов *dim A = 255*.

Схема сжатия выглядит следующим образом:

Вход(снмволы)	Выход(коды)	Новые коды и соответствующие строки
AV	/	256 = AV
E	W	257 = WE
D	E	258 = ED
/	D	259 = D/
WE	256	260 = AVE
/	E	261 = E/
WEE	260	262 = AVEE
AV	261	263 = EAV
EB	257	264 = WEB
/	B	265 = B/
WET	260	266 = AVET
<EOF>	T	

В результате получим выходной код **/WED<256>E<260><261><257>B**.

Как при этом изменилась длина выходного кода в сравнении с входным? Если для двоичного кодирования строки **/WED/WE/WEE/WEB** длиной в 15 букв и размером алфавита **dim A = 255** нам понадобилось бы **15*log$_2$ 255 = 15x8 = 120 бит**, то для двоичного кодирования выходной строки кодера **/WED <256> E<260> <261> <257> B** длиной в 10 новых символов с алфавитом в 264 буквы - **10*9 = 90 бит.**

LZW-декодер, обрабатывая входной поток закодированных данных, восстанавливает из него исходные данные. Так же, как и алгоритм сжатия, декодер добавляет новые строки в словарь всякий раз, когда находит во входном потоке новый код. Все, что ему остается сделать, -это преобразовать входной код в выходную строку символов и отдать ее на выход кодера. Схема работы LZW-декодера выглядит следующим образом: строка на входе кодера - **/WED<256>E<260><261><257>B.**

Вход	СТАРЫЙ КОД	СТРОКА	СИМВОЛ	Новый вход таблицы
новый код		Выход		
/	/	/		
W	/	W	W	256 = АУ
E	W	E	E	257 = WE
D	E	D	D	258 = ED
256	D	/W	/	259 = D/
E	256	E	E	260 = ЛУЕ
260	E	/WE	/	261 = E/
261	260	E/	E	262 = АУЕЕ
257	261	WE	W	263 = ЕЛУ
B	257	B	B	264 = WEB
260	B	/WE	/	265 = B/
T	260	T	T	266 = /WET

Самым замечательным качеством этого способа сжатия является то, что *весь словарь новых символов передается декодеру без собственно передачи*. В конце процесса декодирования декодер имеет точно такой же словарь новых символов, какой в процессе кодирования был накоплен кодером, при этом его создание было частью процесса декодирования. Работа кодера/декодера семейства LZ77 - первой опубликованной версии LZ-метода - выглядит несколько иначе. В алгоритме LZ77 указатели обозначают фразы в окне постоянного размера, предшествующего позиции кода. Максимальная длина заменяемых указателями подстрок определяется параметром ***F*** (обычно это от 10 до 20). Эти ограничения позволяют LZ77 использовать "скользящее окно" из ***N*** символов. Из них первые ***N-F*** были уже закодированы, а последние ***F*** составляют упреждающий буфер. При кодировании символа в первых ***N-F*** символах окна ищут самую длинную, совпадающую с этим буфером строку. Она может частично перекрывать буфер, но не может быть самим буфером. Найденное наибольшее соответствие затем кодируется триадой [***i, j, a***] где ***i*** есть его смещение от начала буфера, ***j*** - длина соответствия, ***a*** - первый символ, не соответствующий подстроке окна. Затем окно сдвигается вправо на ***j +1*** символ и готово к новому шагу алгоритма. Привязка определенного символа к каждому указателю гарантирует, что кодирование будет выполняться даже в том случае, если для первого символа упреждающего буфера не будет найдено соответствие. Объем памяти, требуемый кодеру и декодеру, ограничивается размером окна. Количество бит, необходимое для представления смещения (***i***) в триаде, составляет [***log(N-F)***]. Количество символов (***j***), заменяемых триадой, может быть закодировано [***logF***] битами.

Декодирование осуществляется очень просто и быстро. При этом поддерживается тот же порядок работы с окном, что и при кодировании, но в отличие от поиска одинаковых строк он, наоборот, копирует их из окна в соответствии с очередной триадой.

2.4 Дифференциальное кодирование

Работа дифференциального кодера основана на том факте, что для многих типов данных разница между соседними отсчетами относительно невелика, даже если сами данные имеют большие значения. Например, нельзя ожидать большой разницы между соседними пикселами цифрового изображения. Следующий простой пример показывает, какое преимущество может дать дифференциальное кодирование (кодирование разности между соседними отсчетами) в сравнении с простым кодированием без памяти (кодированием отсчетов независимо друг от друга).

Просканируем 8-битовое (256-уровневое) цифровое изображение, при этом десять последовательных пикселов имеют уровни:

144, 147, 150, 146, 141, 142, 138, 143, 145, 142.

Если закодировать эти уровни пиксел за пикселом каким-либо кодом без памяти, использующим 8 бит на пиксел изображения, получим кодовое слово, содержащее 80 бит.

Предположим теперь, что прежде чем подвергать отсчеты изображения кодированию, мы вычислим разности между соседними пикселами. Эта процедура даст нам последовательность следующего вида:

144, 147, 150, 146, 141, 142, 138, 143, 145, 142.

144, 3, 3, - 4, - 5, 1, - 4, 5, 2, -3.

Исходная последовательность может быть легко восстановлена из разностной, простым суммированием (дискретным интегрированием):

144, 144+3, 147+3, 150-4, 146-5, 141+1, 142-4, 138+5, 143+2, 145-3

144, 147, 150, 146, 141, 142, 138, 143, 145, 142.

Для кодирования первого числа из полученной последовательности разностей отсчетов, как и ранее, понадобится 8 бит, все остальные числа можно закодировать 4-битовыми словами (один знаковый бит и 3 бита на кодирование модуля числа).

Таким образом, в результате кодирования получим кодовое слово длиной $8+9\times4=44$ бита или почти вдвое более короткое, нежели при индивидуальном кодировании отсчетов.

Метод дифференциального кодирования очень широко используется в тех случаях, когда природа данных такова, что их соседние значения незначительно отличаются друг от друга, при том, что сами значения могут быть сколь угодно большими.

Это относится к звуковым сигналам, особенно к речи, изображениям, соседние пиксели которых имеют практически одинаковые яркости и цвет и т.п. В то же время этот метод совершенно не подходит для кодирования текстов, чертежей или каких-либо цифровых данных с независимыми соседними значениями.

2.5 Сжатие видеоданных

Передача цифрового видео от источника (видеокамера или записанный видеоролик) к получателю (видеодисплей) вовлекает в разработку целую цепь различных компонентов и процессов. Ключевыми звеньями этой цепи являются процесс компрессии (кодирования) и декомпрессии (декодирования), согласно которым несжатый цифровой видеосигнал сокращается до размеров, подходящих для его передачи и хранения, а затем восстанавливается для отображения на видеодисплеи. Продуманная разработка процессов компрессии и декомпрессии может дать существенное коммерческое и техническое преимущество продукта, обеспечив лучшее качество видеоизображения, большую надежность и гибкую приспособляемость по сравнению с конкурирующими решениями.
Скорости передачи данных в сетях, а также емкости жестких дисков, флэш-памяти и оптических накопителей постоянно растут. Имея в виду снижение цены передачи и хранения бита информации, не сразу становится очевидным, почему необходимо видеосжатие и его улучшение. Видеосжатие имеет два важных преимущества. Во-первых, оно дает возможность использовать цифровое видео в среде передачи и хранения видеоконтента, которая не поддерживает несжатое («сырое») видео. Например, пропускная способность современного Интернета недостаточна для обращения с несжатым видео в реальном масштабе времени даже при низкой частоте кадра и малом его размере. Цифровой многослойный видеодиск DVD может вместить всего несколько секунд несжатого видео с разрешением и частотой кадров, обеспечивающими обычное телевизионное качество, поэтому использование DVD было бы абсолютно непрактичным без применения аудио и видеосжатия. Во-вторых, видеосжатие делает более эффективным использование ресурсов при передаче и хранении видеоданных. Если доступен высокоскоростной канал, то более привлекательным представляется решение, позволяющее передавать сжатое видео высокого разрешения вместо несжатого видео низкого разрешения. Несмотря на постоянный рост емкости устройств хранения

информации и пропускной способности каналов передачи данных, представляется весьма вероятным, что сжатие видео будет оставаться существенным компонентом мультимедийных сервисов еще многие годы.

Сигнал, несущий определенную информацию, можно сжать путем удаления из него имеющейся избыточности. Избыточность – это компоненты данных, без которых можно обойтись для верного изображения исходной информации. Многие типы данных содержат в себе статистическую избыточность. Такие данные можно эффективно сжимать, используя компрессию без потерь. К сожалению, сжатие без потерь применительно к видео дает относительно небольшой выигрыш. Поэтому для достижения высокой эффективности сжатия приходится применять сжатие с потерями. При сжатии видео с потерями используется несколько типов избыточности:

1) когерентность областей изображения - малое изменение цвета изображения в соседних пикселях (свойство, которое эксплуатируют все алгоритмы сжатия изображений с потерями);
2) избыточность в цветовых плоскостях - используется большая важность яркости изображения для восприятия
3) подобие между кадрами - использование того факта, что на скорости 25 кадров в секунду, как правило, соседние кадры изменяются незначительно.

Использование подобия между кадрами в самом простом и наиболее часто используемом случае означает кодирование не самого нового кадра, а его разности с предыдущим кадром. Для видео типа "говорящая голова" (передача новостей, видеотелефоны) большая часть кадра остается неизменной и даже такой простой метод позволяет значительно уменьшить поток данных. Более сложный метод заключается в нахождении для каждого блока в сжимаемом кадре наименее отличающегося от него блока в кадре, используемом в качестве базового. Далее кодируется разница между этими блоками. Этот метод существенно более ресурсоемкий.

Видеокодек кодирует исходную видеопоследовательность в сжатой форме, а также декодирует сжатую видеопоследовательность, производя цифровую видеокопию, которая или совпадает, или близка к исходной видеопоследовательности.

Кодек преобразует исходный видеоряд с помощью определенной модели. Модель кодирования – это эффективное кодированное представление видеоданных, с помощью которого можно реконструировать эти данные с определенной степенью точности. В идеале модель должна представлять последовательность с наименьшим числом бит и наибольшей возможной точностью. Эти две цели (высокое качество и эффективность сжатия) обычно противоречат друг другу, так как высокая степень сжатия видеоданных предполагает существенное снижение качества на выходе декодера.

Видеокодек состоит из следующих основных функциональных блоков:

1. блока преобразования цветового пространства;
2. блока устранения временной статистической взаимосвязи (схожести соседних по времени кадров видеопотока между собой);
3. блока устранения пространственной статистической взаимосвязи между соседними пикселами на кадре;
4. блока энтропийного кодера.

Цветовые пространства.

В цветовом пространстве RGB пикселы цветного изображения представляются с помощью трех чисел, указывающих относительное соотношение красного (Red), зеленого (Green) и голубого (Blue) цветов (три основные компоненты видимого света) [9, 10]. Любой цвет можно получить с помощью комбинации красного, зеленого и голубого цветов в соответствующей пропорции. Пространство RGB хорошо приспособлено для фиксирования и показа цветных изображений. Цветные электроннолучевые трубки CRTs (Cathode Ray Tubes) и жидкокристаллические дисплеи отображают RGB-изображения, отдельно освещая красные, зеленые и голубые компоненты каждого пиксела в соответствии с интенсивностью каждого из них. Если смотреть на экран с расстояния обычного зрителя, то различные компоненты сливаются в единый «правильный цвет». В силу этого, цветовое пространство RGB применяется в компьютерной технике, однако при таком подходе практически нет возможности эффективного сжатия изображений, т.к. для правильного отображения, все три компоненты должны быть представлены с одинаковым разрешением. Следующее цветовое пространство предлагает выход из этой ситуации.

YCbCr. Известно, что органы зрения человека менее чувствительны к цвету предметов, чем к их яркости (светимости). В цветовом пространстве RGB все три цвета считаются одинаково важными, и они обычно сохраняются с одинаковым разрешением. Однако можно отобразить цветное изображение более эффективно, отделив светимость от цветовой информации и представив ее с большим разрешением, чем цвет. Цветовое пространство YCbCr и его вариации (иногда их обозначают YUV) является популярным методом эффективного представления цветных изображений. Буква Y обозначает компоненту светимости, которая вычисляется как взвешенное усреднение компонент R, G и B по следующей формуле:

$$y = k_r R + k_g G + k_b B \tag{5}$$

где k обозначает соответствующий весовой множитель.

Цветовая информация может быть представлена компонентами цветовых разностей, т.е. каждая из этих компонент представляет собой разность между компонентами R, G и B и компонентой светимости Y. Таким образом:

$$\begin{gathered} Cb = B - Y \\ Cr = R - Y \\ Cg = G - Y \end{gathered} \tag{6}$$

Таким образом, цветное изображение полностью описывается компонентой светимости *Y* и тремя хроматическими составляющими. Возникает резонный вопрос – до преобразования имелось три компоненты составляющие изображение – стало четыре. Однако, зная две из трех хроматических составляющих, можно легко вычистить четвертую, так как сумма *Cb* + *Cr* + *Cg* является постоянной. Для описания изображения выбираются составляющие *Cb* и *Cr*. Преимущества такого способа представления изображений состоит в том, что можно не сжимая компоненту светимости Y, сжать световые составляющие, представив их с меньшим разрешением, что и осуществляется в алгоритме JPEG на втором шаге сжатия. Перед тем, как отображать картинку на экране, требуется произвести обратное преобразование из YCbCr в RGB. Формулы для прямого и обратного преобразования выглядят следующим образом:

$$Y = k_r R + (1 - k_b - k_r)G + k_b B$$

$$C_b = \frac{0.5}{1 - k_b}(B - Y)$$

$$C_r = \frac{0.5}{1 - k_r}(R - Y)$$

$$R = Y + \frac{1 - k_r}{0.5} C_r$$

$$G = Y - \frac{2k_b(1 - k_b)}{1 - k_b - k_r} C_b - \frac{2k_r(1 - k_r)}{1 - k_b - k_r} C_r$$

$$B = Y + \frac{1 - k_b}{0.5} C_B$$

Рекомендация ITU-T с идентификатором BT.601 предлагает коэффициенты k_b = 0.114 и k_r = 0.229. С этими коэффициентами получаем следующие формулы:

$$Y_{601} = 0.299R' + 0.587G' + 0.114B'$$

$$Cb = -0.172R' - 0.339G' + 0.511B' + 128$$

$$Cr = 0.511R' - 0.428G' - 0.083B' + 128$$

Данные формулы используются для кодирования восьмибитного сигнала RGB с диапазоном возможных значений от 16 до 235, то есть 16 соответствует полностью белому, а 236 – полностью черному цвету. Это сделано в целях улучшения передачи изображений и видеопоследовательностей по линиям передачи (телевидение). При таких значениях промежутки от 0 до 15 и от 237 до 256 содержит шум, который при преобразовании отбрасывается, что позволяет улучшить шумовые характеристики изображения.

При использовании в компьютерной технике необходимость в этом отпадает, и диапазон значений сигнала RBG является полным – от 0 до 256. В этом случае используются следующие формулы преобразования:

$$Y_{601} = 0.257R' + 0.504G' + 0.098B' + 16$$
$$Cb = -0.148R' - 0.291G' + 0.439B' + 128$$
$$Cr = 0.439R' - 0.368G' - 0.071B' + 128$$

В этом случае полностью белый цвет – 0, полностью черный – 256. В контексте использования данного кодирования в телевещании говорят о «суперчерном» и «супербелом» цветах. Данный вариант преобразования YCbCr имеет название YCbCr: SDTV (Soft Definition Television). Существует также еще один вариант преобразования.

Недавно появившийся стандарт телевещания HDTV (High Definition Television) использует несколько иные формулы перехода из RGB в YCbCr при диапазоне RGB от 16 до 236:

$$Y_{709} = 0.213R' + 0.715G' + 0.072B'$$
$$Cb = -0.117R' - 0.394G' + 0.511B' + 128$$
$$Cr = 0.511R' - 0.464G' - 0.047B' + 128$$

и при полном диапазоне от 0 до 256:

$$Y_{709} = 0.183R' + 0.614G' + 0.062B' + 16$$
$$Cb = -0.101R' - 0.338G' + 0.439B' + 128$$
$$Cr = 0.439R' - 0.399G' - 0.040B' + 128$$

Другие цветовые пространства. Однако, цветовое пространство YCbCr – не единственное, использующее для передачи изображения компоненту светимости и две компоненты цветоразности. На данный момент существуют следующие цветовые пространства:

YUV. Используется стандартами PAL (Phase Alternation Line), NTSC (National Television System Committee) и SECAM (Sequential Color with Memory). Цветовое пространство YCbCr было разработано в рамках рекомендации ITU-R BT.601 на основе именно этого цветового пространства. Формулы перехода из RGB в YUV выглядят следующим образом:

$$Y' = 0.299 * R' + 0.587 * G' + 0.114 * B'$$
$$U = -0.147 * R' - 0.289 * G' + 0.436 * B'$$
$$V = 0.615 * R' - 0.515 * G' - 0.100 * B'$$

YIQ – это цветовое пространство также разработано на основе YUV и опционально используется в NTSC. Здесь «I» – «inphase» – синфазный сигнал, «Q» – «quadrature» – квадратура. Формулы перехода из RGB:

$$Y' = 0.299 * R' + 0.587 * G' + 0.114 * B'$$
$$I = -0.596 * R' - 0.275 * G' - 0.321 * B'$$
$$Q = 0.212 * R' - 0.523 * G' - 0.311 * B'$$

Photo YCC (торговая марка Eastman Kodak Company) – было разработано для кодирования изображения на носителях Photo CD. Целью было создания цветового пространства, независимого от устройства отображения. Формулы перехода из RGB:

$$Y = 0.213 * R' + 0.419 * G' + 0.081 * B'$$
$$C_1 = -0.131 * R' - 0.256 * G' + 0.387 * B' + 156$$
$$C_2 = 0.373 * R' - 0.312 * G' - 0.061 * B' + 137$$

Форматы сэмплирования

Формат 4:4:4 подразумевает, что все три компоненты (У, СЬ и Сr) имеют одинаковое разрешение и, следовательно, сэмплы (отсчеты) всех компонентов присутствуют в каждом пикселе. Число в пропорции означает относительную долю каждой компоненты при сэмплировании в горизонтальном направлении, т.е. для каждой из четырех компонент яркости отбирается по четыре хроматические компоненты. Сэмплирование по формату 4:4:4 означает полную точность в передаче хроматических компонент.

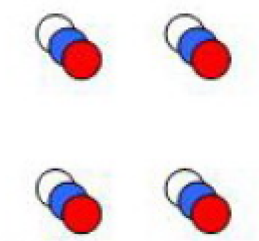

Рис.4. - Семплирование 4:4:4 (сохраняется по 4 пиксела в яркостной и цветоразностных компонент).

При сэмплировании по формуле 4:2:2 (этот формат иногда обозначается YUY2) хроматические компоненты по вертикали имеют одинаковое разрешение с яркостью, а по горизонтали они имеют половину от разрешения яркости. Числа 4:2:2 означают, что на каждые четыре сэмпла яркости Y по горизонтали отбирается только две компоненты СЬ и две компоненты Cr. Формат 4:2:2 используется для высококачественного цветного видео.

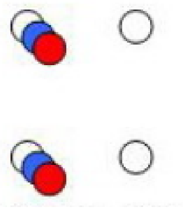

Рис. 5. - Семплирование 4:2:2 (сохраняется 4 пиксела яркости и по две – цветоразных)

В популярном формате семплирования 4:2:0 (YV12) каждая компонента СЬ и Cr имеет и по вертикали и по горизонтали половину разрешения по сравнению с Y. Пропорция 4:2:0 выглядит несколько странной, поскольку эти числа не имеют обычной интерпретации, а само это выражении просто является данью исторической традиции, когда под этим «кодом» подразумевался именно этот формат сэмплирования, который отличается от форматов 4:4:4 и 4:2:2. Цветное сэмплирование 4:2:0 широко используется во многих потребительских приложениях, таких как видеоконференции, цифровое телевидение и диски DVD. Поскольку хроматические компоненты отбираются в четыре раза реже компонент яркости, то пространство 4:2:0 YCbCr требует в два раза меньше сэмплов по сравнению с форматом видео 4:4:4 (или RGB).

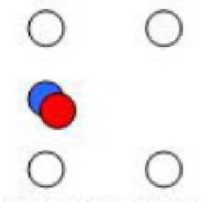

Рис. 6. - Семплированиеи 4:2:0 (Сохраняется 4 пиксела яркости и по одному – для цветоразностных компонент)

Motion JPEG и MPEG.

Motion JPEG — покадровый метод видеосжатия, основной особенностью которого является сжатие каждого отдельного кадра видеопотока с помощью алгоритма сжатия изображений JPEG. Motion JPEG использует внутрикадровое сжатие с потерями на основе дискретного косинусного преобразования (ДКП). Эта математическая операция преобразует каждый кадр/поле изображения из пространственной области в частотную.

Психовизуальная модель, основанная на особенностях восприятия изображений человеком, использует обычно грубое квантование высокочастотной составляющей изображения и более аккуратное квантование низкочастотной составляющей, снижая тем самым точность передачи резких переходов яркости и оттенков цвета. Квантованные коэффициенты ДКП без потерь упаковываются в выходной битовый поток с использованием кодов Хаффмана либо с помощью арифметического кодирования. Почти все программные реализации MJPEG позволяют пользователям контролировать степень сжатия (а также другие параметры) и достигать компромисса качества изображения и размера файла. При аппаратных решениях параметры кодирования, как правило, предварительно выбраны и зафиксированы.

Заголовок каждого кодированного MJPEG обычно соответствует стандарту JPEG, однако, допустимыми являются некоторые несоответствия стандарту. Так, например, в нём может отсутствовать маркер DHT, определяющий таблицы для хаффмановского декодирования. В этом случае в процессе декодирования следует использовать таблицы, приведённые в разделе K.3 стандарта JPEG (CCITT Rec. T.81).

В MJPEG применяется схема только внутрикадрового сжатия (по сравнению с более сложными схемами вычисления с межкадровым сжатием). В то время, как современные видеоформаты с межкадровым сжатием, такие как MPEG1, MPEG2, H.264/MPEG-4 AVC и им подобные, достигают в среднем степени сжатия 1:50 и более, отсутствие в MJPEG межкадрового сжатия, как правило, не позволяет получать коэффициенты сжатия, превосходящие 1:20, в зависимости от допустимости пространственных искажений в декодированных кадрах видеопоследовательности. Так как кадры сжимаются независимо друг от друга, MJPEG требует меньше вычислительных ресурсов и оперативной памяти на этапе кодирования. Однако, декодирование MJPEG может оказаться более затратным, чем при использовании межкадрового сжатия, поскольку, во-первых, предполагает полное декодирование в MJPEG каждого макроблока изображения, тогда как при использовании схем с межкадровым сжатием часть макроблоков, помеченных как «skip», не декодируется, а берётся из предыдущих кадров. Во-вторых, время выполнения процедур хаффменовского декодирования и обратного ДКП зависит от информационной насыщенности декодируемого макроблока изображения, которая при отсутствии межкадрового сжатия оказывается значительно большей, чем при его наличии (в первом случае декодируется полное изображение, во втором — разностное, то есть не само изображение, а лишь его отличие от предсказанного по предыдущим кадрам).

При внутрикадровой схеме сжатия в MJPEG качество изображения зависит непосредственно от статической (пространственной) сложности каждого видеокадра. Кадры с большими гладкими переходами или монотонными областями хорошо сжимаются, но при слишком высоких степенях сжатия содержат, помимо оригинальных деталей, видимые артефакты сжатия в виде блоков размером 8x8 пикселей, несколько отличающиеся по яркости и оттенку цвета. Появление их связано с грубым квантованием низкочастотных коэффициентов ДКП. Кадры, имеющие сложные текстуры, тонкие кривые линии, помимо артефактов блочности содержат также артефакты, проявляющиеся в виде шума вокруг тонких линий и на резких границах (так называемый эффект Гиббса), связанные с грубым квантованием высокочастотных коэффициентов ДКП.

Основным преимуществом видеосжатия Motion JPEG является простота реализации, что делает MJPEG подходящим для реализации в устройствах с ограниченными вычислительными ресурсами. Чрезвычайно быстрый нелинейный видеомонтаж — если какой-либо кадр берётся целиком (без изменений) из одного MJPEG-источника, его можно записать в выходной MJPEG-поток как есть, без декодирования-сжатия.

При высоком битрейте MJPEG даёт качественные стоп-кадры, что позволяет его использовать, например, в системах видеонаблюдения (там это нужно, чтобы, например, выяснить номер проехавшего автомобиля или подробно рассмотреть лицо преступника). Однако при отсутствии межкадрового сжатия достижение заданного битрейта требует использования большего, чем в случае MPEG, покадрового сжатия, что приводит появлению заметных артефактов сжатия. Недостатками MJPEG являются более низкий коэффициент сжатия по сравнению с потоковыми методами сжатия, например, MPEG-4 и проявляющиеся при высоких степенях сжатия артефакты.

MPEG (Moving Picture Experts Group) – целое семейство стандартов сжатия цифровой информации, разработанное и стандартизированное одноименной экспертной группой специалистов, сформированной организацией ISO в далеком 1988 году.

Первым плодом их творения стал исходный стандарт видео и аудио компрессии ***MPEG-1***, а в 1993 году при участии компаний JVC и Philips, была разработана его спецификация Video CD (VCD), которая и известна многим пользователям. Из названия видно, что VCD является форматом для хранения сжатого видео со звуком на обычных компакт дисках. Использование для кодирования алгоритмов MPEG-1 позволяет получать видеопоток шириной до 1,5 Мбит в секунду с разрешением кадра 352x288 точек для PAL или 352x240 для NTSC, после чего на обычном CD может уместиться 74 минуты видео со звуком качества VHS (как у обычного видеомагнитофона).

В 1995 году увидел свет популярнейший стандарт MPEG-2, который впоследствии получил широкое распространение в цифровых видеодисках DVD, а так же при передаче сигнала кабельного и спутникового телевидения. Качество картинки здесь значительно выше, чем у предшественника: при 25 кадрах в секунду, разрешение составляет 720x576 точек для системы PAL, а для системы NTSC – 720x480 при 30 кадрах/с. При этом, средняя максимальная ширина потока равна 9,8 Мбит/с, что практически в 7 раз выше, чем у Video CD. Еще одним неоспоримым преимуществом MPEG-2 является возможность сохранения пятиканальной аудиодорожки (Dolby Digital 5.1 и DTS) . Максимальная емкость двухслойного DVD диска (DVD-9) составляет 8,5 Гбайт, на который можно записать до трех часов видео с полным качеством. Вместе с MPEG-2, приблизительно в тоже время, начал разрабатываться новый стандарт MPEG-3, предназначенный для кодирования аудио и видеопотоков в телевидении высокой чёткости со скоростью передачи данных от 20 до 40 Мбит/с. Но довольно скоро выяснилось, что для этих задач можно использовать несколько модифицированную версию стандарта MPEG-2, после чего все дальнейшие разработки MPEG-3 были прекращены и на сегодняшний день этот стандарт не используется. Стоит отметить, что довольно часто термин «MPEG-3» многие пользователи ассоциируют с популярной технологией сжатия звука MP3. Но это в корне не верно, так как ее правильное название - MPEG-1 Audio Layer 3 [12, 16].

Наконец, в 1998 году появилось новое семейство форматов сжатия видео – ***MPEG-4***. Разрабатывалось оно с целью улучшения качества картинки при низкой скорости потока. Прежний стандарт MPEG-2, рассчитанный на высокий битрейт, с этой задачей справиться не мог, так что алгоритмы сжатия пришлось серьезно модифицировать. Так же MPEG-2 не походит и для хранения видео высокой четкости (HD) с разрешениями от 1280x720 (720p) до 1920x1080 пикселей (1080i или 1080p), которое все больше и больше набирает популярность. На сегодняшний день MPEG-4 является основным стандартом сжатия мультимедиа контента, и хотя DVD списывать со счетов еще рано, практически все современные фото и видеокамеры снимают в HD-качестве. Так что для сохранения видео с таких устройств на компьютер, в любом случае придется ориентироваться на кодеки семейства MPEG-4x [11].

3. Применение корректирующего кодирования в системах связи

Приведем несколько типичных примеров использования эффективного кодирования в системах связи. Один из типичных случаев характерен для систем, в которых требуются очень мощные коды. Под мощными кодами понимаются коды с длинными блоками и большим кодовым расстоянием. Обычный метод реализации таких кодов состоит в *каскадировании* двух или более простых кодов. Еще одной интересной проблемой является кодирование для каналов, в которых ошибки возникают не независимо, а пакетами.

3.1 Каскадные коды

Каскадные коды были впервые предложены в качестве метода практической реализации кода с большой длиной блока и высокой корректирующей способностью [13, 14]. Эта цель достигается введением нескольких уровней кодирования, обычно – двух. Основную идею каскадного кодирования с двумя уровнями иллюстрирует рис. 4.

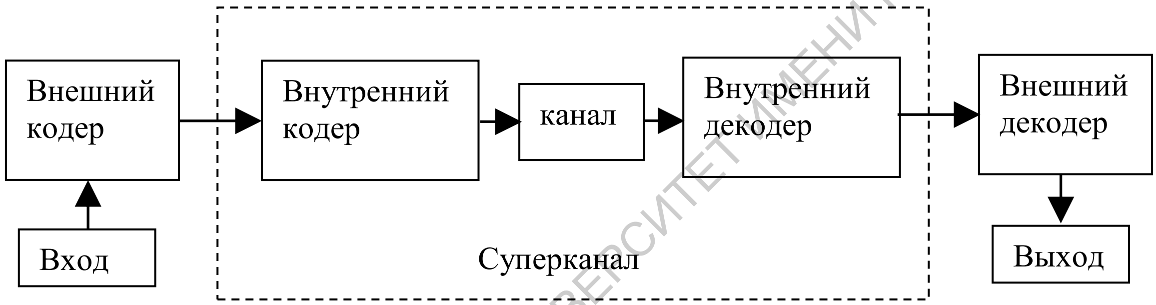

Рис. 4. Каскадное кодирование.

В этой схеме комбинацию внутреннего кодера, канала и внутреннего декодера иногда называют *суперканалом*, аналогично, комбинацию внешнего и внутреннего кодеров - *суперкодером*, а комбинацию внутреннего и внешнего декодеров – *супердекодером*.

Длина каскадного кода получается равной $N1 = N \cdot n$ двоичных символов, где N - длина внешнего кода, а n - длина внутреннего кода. При этом информационная длина кода составляет $K1 = K \cdot k$ двоичных символов, а скорость кода $R1 = R \cdot r$. Несмотря на то, что общая длина кода получается большой и, соответственно, значительно возрастает его исправляющая способность, его декодирование может выполняться с помощью двух декодеров, рассчитанных на длины составляющих его кодов n и N. Это позволяет многократно снизить сложность декодера в сравнении с тем, если бы такая исправляющая способность достигалась одноуровневым кодированием.

Простейшей иллюстрацией к каскадному кодированию является итеративный код, рассмотренный ранее. Этот код состоит из простых кодов с проверкой на четность (по строкам и столбцам), но в то же время обладает исправляющей способностью. На практике, конечно, используются гораздо более сложные коды.

Обычно внешнее кодирование выполняется *блочными кодами*, а внутреннее *сверточными кодами*. Построенный каскадный код эквивалентен линейному двоичному коду с минимальным расстоянием $d>d_1d_2$.

Использование каскадных кодов позволяет сделать скорость передачи сколь угодно близкой к пропускной способности канала. Они во многих отношениях наиболее перспективны среди известных блоковых помехоустойчивых кодов.

Каскадное кодирование широко применяется на практике, в частности, при помехоустойчивом кодировании речевой информации в системе сотовой связи формата *GSM*.

Все рассмотренные ранее методы кодирования и примеры расчета их эффективности относились к каналам без памяти, то есть к каналам, в которых вероятность ошибки постоянна и не зависит от времени. На практике же ошибки обычно группируются так называемыми *пакетами*. При постоянной средней вероятности ошибок на большом интервале времени значение $P_{ош}$ на отдельных коротких интервалах может значительно превышать среднее значение – $P_{ош\ ср}$. Использование традиционных методов кодирования-декодирования потребовало бы в этом случае применения сложных кодов с большой исправляющей способностью и, соответственно, большой избыточностью.

В каналах с памятью ошибки могут возникать группами и их оказывается слишком много для исправления в конкретном временном интервале. Примером являются каналы с замираниями, когда сигнал поступает на приемник по двум или более путям различной длины. В итоге суммарный сигнал оказывается искаженным. Таким эффектом обладают каналы мобильной беспроводной связи. Во многих системах появление групповых ошибок могут стимулировать импульсные помехи.

Одно из возможных решений в таких случаях заключается в использовании достаточно простого кода, рассчитанного на исправление одиночных ошибок, вместе с парой устройств, выполняющих чередование (*интерливинг* – “*interleaving*” перемежение) закодированных символов перед их передачей в канал и восстановление (дечередование) после приема. При такой обработке кодовой и принятой последовательностей ошибки на входе декодера распределяются более равномерно.

Чередование битов кодированного сообщения перед передачей и обратная операция после приема приводят к рассеянию пакета ошибок во времени и они становятся для декодера случайно распределенными. Таким образом происходит как бы превращение канала в канал без памяти, и, следовательно, появляется возможность эффективно использовать коды с коррекцией случайных ошибок, рассмотренные ранее. Структурная схема системы с чередованием показана на рис. 5.

Устройство чередования в этой схеме переупорядочивает (переставляет) символы передаваемой последовательности некоторым детерминированным образом в пределах нескольких блоков. Требуемый промежуток чередования определяется длительностью пакета ошибок. С помощью устройства восстановления производится обратная перестановка, восстанавливающая исходный порядок следования символов.

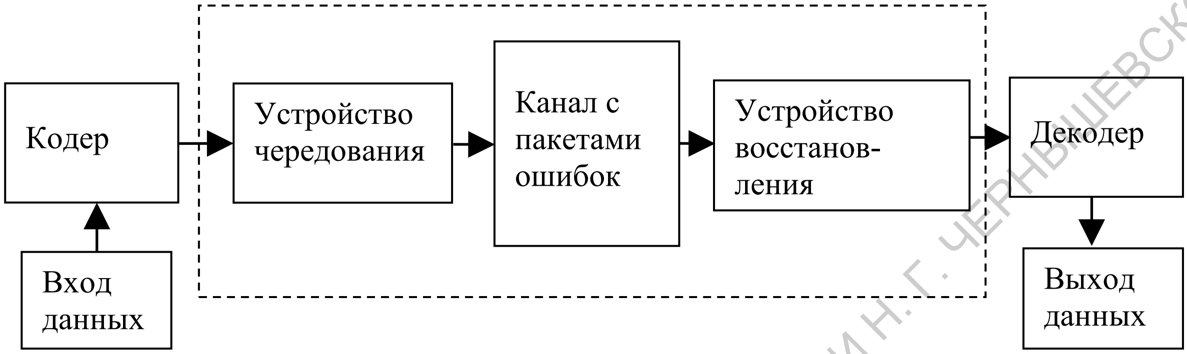

Рис. 5. Структурная схема системы с чередованием.

Следует отметить, что имеется два способа чередования-восстановления. Первый способ – периодическое чередование. Он проще, но при изменении характера помех может оказаться неустойчивым. Более сложное – псевдослучайное чередование, которое обладает при нестационарных ошибках гораздо большей устойчивостью.

3.2 Периодическое чередование

При периодическом чередовании функция перестановок периодична с некоторым периодом. Типичное блочное устройство чередования работает следующим образом. Кодовые символы записываются в матрицу, имеющую N строк и M столбцов построчно, а читаются из нее по столбцам. На приемной стороне операция выполняется в обратном порядке: запись производится по столбцам, а чтение - по строкам. При этом происходит восстановление исходного порядка следования символов. Естественно, что процедуры перемежения и дечередования должны быть синхронизированы.

При таком перемежении достигается следующее: любой пакет ошибок длиной $m \leq M$ переходит на выходе устройства восстановления в одиночные ошибки, каждая пара которых разделена не менее чем N символами. Правда, при этом любая периодическая с периодом *M одиночная ошибка превращается в пакет*, но вероятность такого преобразования очень мала, хотя и существует [17-20].

Проиллюстрируем алгоритм периодического чередования на следующем примере. Пусть передается 7 кодовых слов, каждое из которых состоит из 7 кодовых символов. Допустим, наш код способен исправлять однобитовые ошибки в любой 7-символьной последовательности. Если пакет ошибок имеет длительность, равную длине одного кодового слова (7 символов), то он способен уничтожить информацию в одном или двух кодовых словах. На рис. 6 приводится один из возможных алгоритмов чередования. Вначале в результате перемешивания битов кодовые символы перемешиваются (Рис 6б)). Полученный поток преобразуется в модулированный сигнал и передается по каналу. В результате импульсной помехи образуется групповая ошибка длиной в 7 символов. В процессе приема восстанавливается исходный порядок битов. Так как ошибки перераспределились по одной на каждое слово, поток легко декодируется любым традиционным способом.

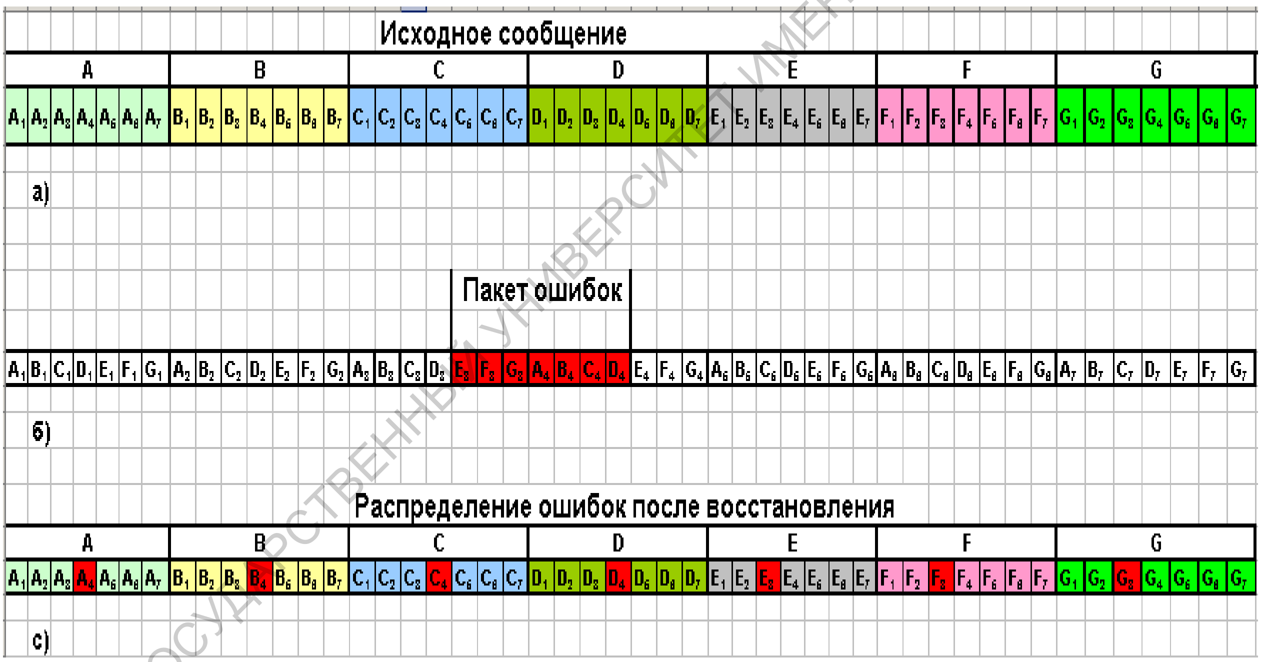

Рис. 6. Пример процедуры чередования битов: а) исходные кодовые слова, содержащие по 7 символов, б) при передаче сообщения, подвергнутого битовому чередованию, возникла группа ошибок длительностью 7 символов, в) пакет ошибок перераспределяется после восстановления, в результате каждое слово имеет всего по одной ошибке.

Чтение битов по горизонтали (исходные кодированные слова)

Чтение битов по вертикали (реализация чередования битов)

A_1	A_2	A_3	A_4	A_5	A_6	A_7
B_1	B_2	B_3	B_4	B_5	B_6	B_7
C_1	C_2	C_3	C_4	C_5	C_6	C_7
D_1	D_2	D_3	D_4	D_5	D_6	D_7
E_1	E_2	E_3	E_4	E_5	E_6	E_7
F_1	F_2	F_3	F_4	F_5	F_6	F_7
G_1	G_2	G_3	G_4	G_5	G_6	G_7

Рис. 7. Реализация процедуры чередования.

При псевдослучайном чередовании блоки из L символов записываются в память с произвольной выборкой, а затем считываются из нее псевдослучайным образом. Порядок перестановок, одинаковый для устройств чередования и восстановления, можно записать в ПЗУ и использовать его для адресации.

Как и для периодического чередования, существует вероятность того, что ошибки будут следовать таким образом (синхронно с чередованием), что одиночные ошибки будут группироваться в пакеты. Но такая вероятность чрезвычайно мала (если, конечно, это не организованная помеха и противник не знает порядка перемежения). Случайное же совпадение порядка следования перестановок при перемежении и импульсов помехи при достаточной длине L практически невероятно.

3.3 Адаптивные корректирующие коды

Существенным недостатком многих корректирующих кодов является их слабая приспособленность к изменяющимся условиям передачи информации. Избыточность таких кодов постоянна и выбирается обычно из соображений обеспечения требуемой верности при наихудших условиях передачи. Если избыточность кода привести в соответствие с реальным состоянием канала в контролируемом интервале времени, можно существенно повысить эффективность использования каналов без снижения достоверности. Эта идея лежит в основе адаптивных корректирующих кодов [17].

Различают методы *адаптивного декодирования*, когда в зависимости от числа ошибок в принимаемых кодовых комбинациях изменяют структуру или параметры алгоритмов декодирования и функции схем декодеров, и методы *адаптивного кодирования,* когда наряду с этим изменяют структуру или параметры кодов, алгоритмов кодирования и схем кодеров.

Для построения схем адаптивного кодирования требуется канал обратной связи, по которому на передающую сторону направляют информацию о качестве канала и об условиях приема. Обнаружение ошибок адаптивными кодами систем без обратной связи позволяет обеспечить практически любую заданную достоверность при относительно невысокой сложности оборудования, но часть информации теряется, так как комбинации с обнаруженными ошибками потребителю не выдаются. Исправление ошибок также позволяет обеспечить достоверность передачи, но при отсутствии потерь информации. Платой за это является значительное увеличение длины кодовых комбинаций – до десятков тысяч разрядов, а также существенное усложнение аппаратуры. Недостатком систем без обратной связи является и то, что передатчик не получает никаких подтверждений о том, как принята информация приемником. Системы без обратной связи находят применение в случаях, когда канал обратной связи невозможно организовать, или когда недопустимы задержки при передаче информации. К таким системам относятся, например, некоторые системы спутниковой связи.

Наиболее широкое распространение получили системы с обратной связью, в которых повышение достоверности достигается обнаружением ошибок на приемном конце и повторением только неправильно принятых комбинаций. Адаптивное управление повторением информации существенно приближает избыточность кода к информационному пределу.

Основными задачами, которые решают при построении систем с адаптивными алгоритмами кодирования и декодирования, являются:

- разработка методов и аппаратуры контроля состояния каналов;
- оптимизация использования полученной информации о состоянии канала для изменения способа кодирования, параметров сигналов и т.п.;
- отыскание таких алгоритмов кодирования и декодирования, при которых системы становятся инвариантными относительно статистических особенностей реальных каналов и позволяют добиться оптимальной избыточности.

Теория адаптивного корректирующего кодирования интенсивно развивается, так как она позволяет более полно учесть реальные условия передачи информации.

4. Помехоустойчивое кодирование

В настоящее время теория кодирования имеет важное широкое практическое применение как средство экономной, удобной, быстрой, а также надежной передачи сообщений по линиям связи с различного вида шумами (телефон, телеграф, радио, телевидение, компьютерная, космическая связи и т. д.). Подлинный взрыв развития теории связи начался в послевоенные годы, с 1948–1949 гг., с появлением классических работ Клода Шеннона и Норберта Винера. Труды Н. Винера были порождены исследованиями военного времени по автоматическому управлению огнем, труды К. Шеннона знаменитые "Математическая теория связи" и "Связь при наличии шума" – исследованиями по шифрованию сообщений и их передачи по секретным каналам связи. Математические модели Н. Винера и К. Шеннона довольно сильно различались: сигнал по Н. Винеру может обрабатываться после воздействия шумом, по К. Шеннону сигнал можно обрабатывать как до, так и после передачи по каналу связи с шумами. В силу этого и других различий, Винеровские труды легли в основу теории автоматического управления, Шенноновские труды оказались основополагающими для задач эффективного использования каналов связи. Таким образом, с 1949 г., с фундаментальных работ К. Шеннона, началось бурное развитие теории кодирования как отдельной научной дисциплины, а также развитие таких тесно с нею связанных научных дисциплин, как сжатие информации и криптология [17-20].

Управление правильностью передачи информации выполняется с помощью помехоустойчивого кодирования. Есть коды, обнаруживающие ошибки, и корректирующие коды, которые еще и исправляют ошибки. Помехозащищенность достигается с помощью введения избыточности, дополнительных битов. В симплексных каналах связи устраняют ошибки с помощью корректирующих кодов. В дуплексных - достаточно применения кодов, обнаруживающих ошибки. Это основные методы, используемые в информационных сетях.

Шеннон показал, что с каждым каналом связано измеряемое в битах в секунду и называемое пропускной способностью канала число C, имеющее следующее значение. Если требуемая от системы связи скорость передачи информации R (измеряемая в битах в секунду) меньше C, то, используя коды, контролирующие ошибки, для данного канала можно построить такую систему связи, что вероятность ошибки на выходе будет сколь угодно мала. В самом деле, из шенноновской теории информации следует тот важный вывод, что построение слишком хороших каналов является расточительством; экономически выгоднее использовать кодирование. Фактически в работе Шеннона утверждается, что мощность сигнала, шум в канале и полоса частот ограничивают лишь скорость передачи, а не ее точность. Шеннон, однако, не указал, как найти подходящие коды, а лишь доказал их существование. В пятидесятые годы много усилий было потрачено на попытки построения в явном виде классов кодов, позволяющих получить обещанную сколь угодно малую вероятность ошибки, но результаты были скудными. В следующем десятилетии решению этой увлекательной задачи уделялось меньше внимания; вместо этого исследователи кодов предприняли длительную атаку по двум основным направлениям.

Первое направление носило чисто алгебраический характер и преимущественно рассматривало блоковые коды. Первые блоковые коды были введены в 1950 г., когда Хэмминг описал класс блоковых кодов, исправляющих одиночные ошибки. Коды Хэмминга были разочаровывающе слабы по сравнению с обещанными Шенноном гораздо более сильными кодами. Несмотря на усиленные исследования, до конца пятидесятых годов не было построено лучшего класса кодов. В течение этого периода без какой-либо общей теории были найдены многие коды с малой длиной блока. Основной сдвиг произошел, когда Боуз и Рой-Чоудхури [1960] и Хоквингем [1959] нашли большой класс кодов, исправляющих кратные ошибки (коды БЧХ), а Рид и Соломон [1960] нашли связанный с кодами БЧХ класс кодов для недвоичных каналов. Хотя эти коды остаются среди наиболее важных классов кодов, общая теория блоковых кодов, контролирующих ошибки, с тех пор успешно развивалась.

Открытие кодов БЧХ привело к поиску практических методов построения жестких или мягких реализации кодеров и декодеров. Первый хороший алгоритм был предложен Питерсоном. Впоследствии мощный алгоритм выполнения описанных Питерсоном вычислений был предложен Берлекэмпом и Месси, и их реализация вошла в практику как только стала доступной новая цифровая техника. Второе направление исследований по кодированию носило скорее вероятностный характер. Ранние исследования были связаны с оценками вероятностей ошибки для лучших семейств блоковых кодов, несмотря на то, что эти лучшие коды не были известны. С этими исследованиями были связаны попытки понять кодирование и декодирование с вероятностной точки зрения, и эти попытки привели к появлению последовательного декодирования. В последовательном декодировании вводится класс неблоковых кодов бесконечной длины, которые можно описать деревом и декодировать с помощью алгоритмов поиска по дереву. Наиболее полезными древовидными кодами являются коды с тонкой структурой, известные под названием сверточных кодов. Эти коды можно генерировать с помощью цепей линейных регистров сдвига, выполняющих операцию свертки информационной последовательности.

В конце 50-х годов для сверточных кодов были успешно разработаны алгоритмы последовательного декодирования. Интересно, что наиболее простой алгоритм декодирования - алгоритм Витерби - не был разработан для этих кодов до 1967 г. Применительно к сверточным кодам умеренной сложности алгоритм Витерби пользуется широкой популярностью, но для более мощных сверточных кодов он не практичен.

В 70-х годах эти два направления исследований опять стали переплетаться. Теорией сверточных кодов занялись алгебраисты, представившие ее в новом свете. В теории блоковых кодов за это время удалось приблизиться к кодам, обещанным Шенноном: были предложены две различные схемы кодирования (одна Юстесеном, а другая Гоппой), позволяющие строить семейства кодов, которые одновременно могут иметь очень большую длину блока и очень хорошие характеристики. Обе схемы, однако, имеют практические ограничения. Между тем к началу 80-х годов кодеры и декодеры начали появляться в конструкциях цифровых систем связи и цифровых систем памяти.

4.1 Общие сведения о кодах и системах кодированной связи

Код есть форма представления сообщения, не зависящая от его физической сути. Это отличает код от сигнала, который определяет физическое представление сообщения (и кода) в системе связи. На практике, однако, часто связывают абстрактную (символьную) форму кода с физическими сигналами, называя код частотным, временным, фазовым, амплитудным. Код представляют совокупностью (кодовых) символов.

Если сообщения обладают внутренними корреляционными связями, т. е. если одно сообщение некоторым образом зависит от другого, как это обычно бывает при передаче текстов на естественных языках, то помехоустойчивость любого кода может быть повышена за счет статистических связей между сообщениями. Если эти связи слабые, или неизвестны, или их нельзя использовать для повышения помехоустойчивости, то в этом случае форма представления сообщения должна быть избыточной; в частности, число символов в коде сообщения увеличивают, а между кодовыми символами вводят искусственные корреляционные связи. Поэтому в некоторых случаях помехоустойчивые коды называют избыточными. Введение избыточности в код позволяет помимо обнаружения и исправления ошибок повысить энергетическую эффективность линии связи, обузить частотный спектр передаваемого сигнала, сократить время вхождения в связь путем повышения помехозащищенности тракта синхронизации, улучшить корреляционные свойства ансамбля сигналов, простыми средствами реализовать разнесенный прием. Вид помехоустойчивого кода зависит от структуры системы связи, обобщенная схема которой приведена на рис. 1. Всюду рассматриваем системы связи, передающие только дискретные сообщения. В современных системах передачи дискретных сообщений последние поступают на вход системы, как правило, от нескольких источников. Даже если внешний источник один, сама система связи содержит источник сигналов служебной связи, телеуправления и телесигнализации (ТУ-ТС). Скорость поступления сообщений от разных источников может быть как одинаковой, так и различной синхронной с собственной тактовой частотой ап-паратуры связи или асинхронной с ней. Блок уплотнения (БУ) объединяет сообщения, поступающие от разных источников, в единую последовательность, как правило, двоичных символов с тактовой частотой, соответствующей скорости передачи системы связи.

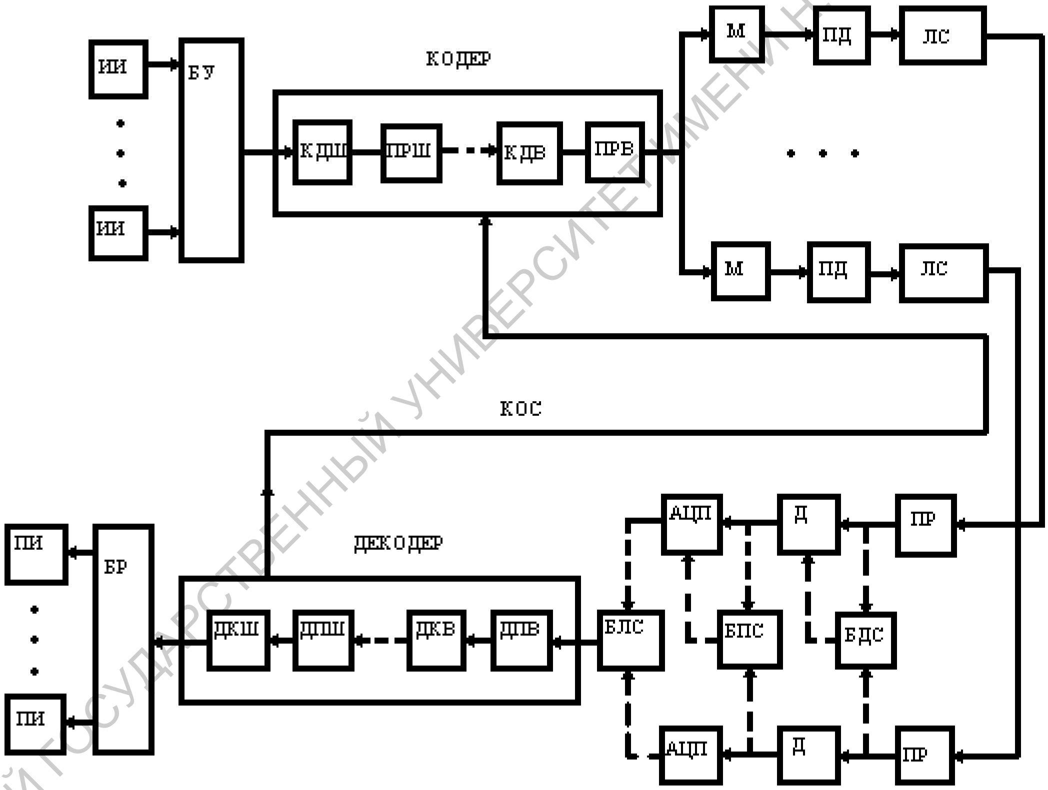

Рис. 7. Схема системы связи: ИИ - источник информации; БУ - блок уплотнения сообщений; КДШ, КДВ - кодеры внешний, внутренний; ПРШ, ПРВ - перемежители внешний, внутренний; М - модулятор; ПД - передатчик; ЛС - линия связи; ПР - приемник; Д - демодулятор; АЦП - аналого-цифровой преобразователь; БДС, БПС, БЛС - блоки додетекторного, последетекторного, логического сложения; ДПШ, ДПВ - деперемежители внешний, внутренний; ДКШ, ДКВ - декодер внешний, внутренний; БР-блок разуплотнения сообщений; ПИ-получатель информации; КОС - канал обратной связи

Схема на рис. 7 может видоизменяться в зависимости от конкретной реализации системы связи. В каналах действуют искажения сигналов, шумы, помехи, которые в дискретном канале проявляются в виде перехода одного значения символа в другое - ложное (событие, состоящее в появлении ошибки) или неиспользуемое (событие, которое называют стиранием). В зависимости от характера ошибок различают дискретные каналы: симметричный (все ложные значения символов равновероятны), асимметричный (некоторые ложные значения символов обладают большей вероятностью), без памяти (искажение символа не зависит статистически от искажения другого выходного символа), с памятью (искажение символа выходной последовательности зависит статистически от искажения другого символа той же последовательности), со стираниями (наряду с ошибками имеют место стирания символов).

Любой канал связи с ограниченными полосой частот, временем передачи и динамическим диапазоном (значений амплитуд) обладает конечной пропускной способностью. Теоретически пропускная способность - это максимальное число переданных двоичных единиц (бит) в единицу времени при сколь угодно малой вероятности ошибок. Реально получаемое число передаваемых бит в единицу времени называют скоростью передачи. При неограниченно малой вероятности ошибок скорость передачи всегда меньше пропускной способности. В канале с ошибками максимальное значение скорости получают путем использования помехоустойчивого кодирования. Последнее требует введения избыточности в передаваемый сигнал: по времени, частоте или амплитуде. Если код согласован с каналом, т. е. код позволяет исправлять наиболее вероятные ошибки, введенная избыточность становится оправданной. Если код не согласован с каналом, ошибки могут быть не только не исправлены, но и размножены кодом. В этом случае применение помехоустойчивого кодирования принесет не пользу, а вред. Для согласования кода с каналом связи необходимо иметь максимальный объем сведений о возможных мешающих влияниях в каналах.

4.2 Помехи в каналах связи

Мешающие влияния разделяют на шумы, помехи, замирания, искажения, ошибки (рис. 8). Обычно шумы имеют естественное происхождение; наиболее существенное влияние оказывает собственный шум приемника. Помехи могут быть также естественного происхождения (грозовые разряды, индустриальные помехи, влияние соседних радиосредств) и преднамеренные. Все разнообразие помех можно свести к шести основным типам: шумовым, импульсным, узкополосным (в пределе-синусоидальным), внутрисистемным, ретранслированным, имитационным. Шумовую помеху представляют в виде внешнего флуктуационного шума, увеличивающего интенсивность шума приемника.

Импульсные помехи (ИП) действуют в течение ограниченного времени; в зависимости от формы импульса различают шумовые (ограниченный во времени шум), видео- и синусоидальные (узкополосные) ИП. Импульс помехи может быть одиночным, однако чаще воздействует пакет ИП, который поражает элементы сигнала, искажая его временные характеристики.

Узкополосная помеха накрывает часть спектра передаваемого сигнала, искажая спектр и ухудшая как спектральные, так и корреляционные свойства сигнала.

Внутрисистемные помехи характерны для асинхронно-адресных систем связи, работающих в одной полосе частот с различением станций по форме адресных сигналов (кодов). Возникают помехи главным образом за счет неидеальности взаимокорреляционных функций адресных кодов.

Ретранслированная помеха создается в результате усиления и переизлучения переданного сигнала одной-двумя соседними станциями. Переизлученный и задержанный сигнал, попадая в приемник истинной станции, создает специфическую помеху, воздействующую тем сильнее, чем хуже корреляционные свойства передаваемых сигналов. Имитационная помеха (ИМП) близка по форме переданному сигналу; степень близости определяется числом передаваемых сигналов и их корреляционными свойствами. Часто ИМП называют также структурной или прицельной помехой. Название "прицельная помеха" становится оправданным при совпадении в приемнике фазы или средней частоты ИМП с фазой переданного сигнала или со средней частотой одного или нескольких частотных подканалов. В последнем случае помеху иногда называют сосредоточенной. В наземных радиолиниях причинами замираний, составляющих основную часть мешающих влияний естественного происхождения, служат многолучевость, метеоусловия, время года. Многолучевость вызывает быстрые замирания, метеоусловия и время года - медленные. Частотную селективность замираний определяют по снижению коэффициента частотной корреляции до значения 0,5 ... 0,6. Интервал частот, лежащий в пределах 1...0,5, называют полосой (интервалом) когерентности канала связи.

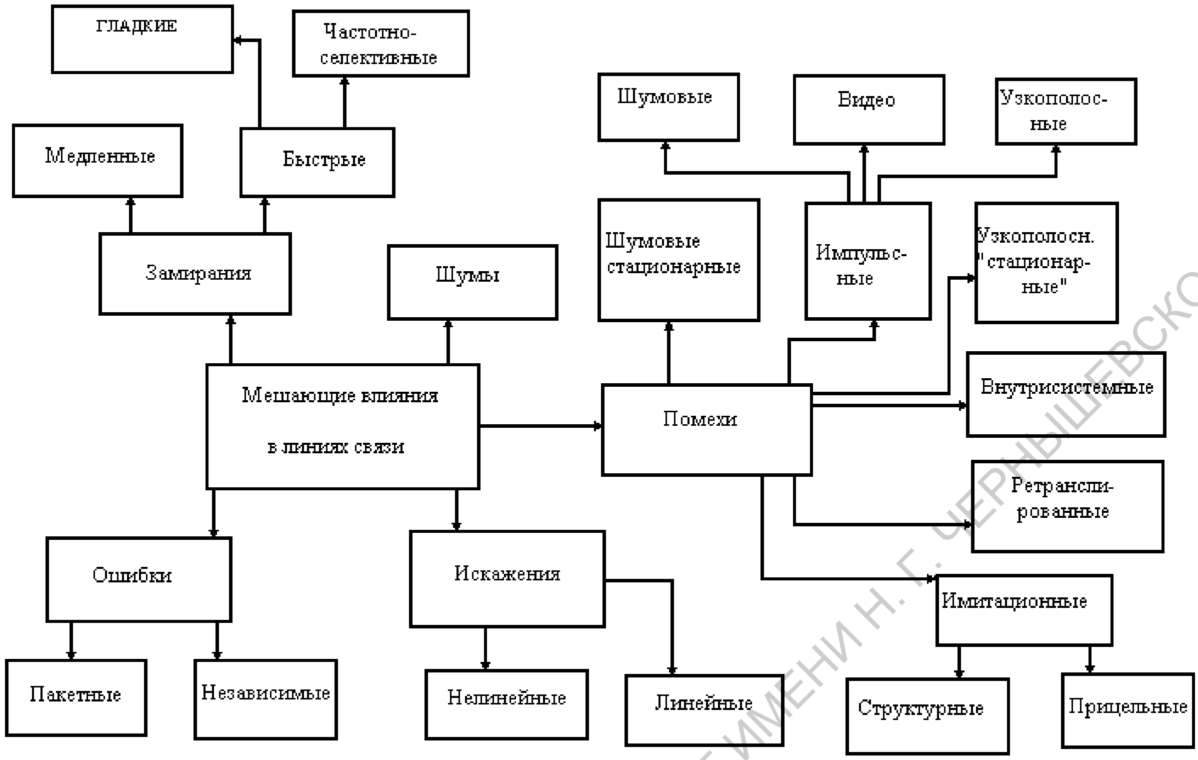

Рис. 8 Классификация мешающих влияний в линиях связи

При передаче сигнала по линии связи он искажается и воспроизводится с некоторой ошибкой. Причиной таких ошибок являются искажения сигналов в канале связи и помехи, воздействующие на сигнал. Искажения часто обусловлены известными характеристиками линии связи и тогда могут быть устранены путем надлежащей коррекции. Помехи заранее неизвестны и поэтому не могут быть полностью устранены. Они весьма разнообразны как по своему происхождению, так и по физическим свойствам. Можно дать следующую классификацию помех по месту их возникновения:

1. атмосферные помехи;
2. промышленные помехи (индустриальные помехи);
3. космические помехи;
4. электризационные помехи;
5. помехи посторонних каналов связи;
6. внутренние шумы.

Атмосферные помехи обусловлены электрическими процессами в атмосфере и, прежде всего, грозовыми разрядами. Энергия этих помех сосредоточена, главным образом, в области ДВ и СВ.

Промышленные помехи возникают из-за резких изменений тока в электрических цепях всевозможных электроустановок. К ним относятся помехи от электротранспорта, электрических моторов, медицинских установок, систем зажигания двигателей и т.д.

Космические помехи создаются радиоизлучением внеземных источников. Они создают общий шумовой фон и в наибольшей степени проявляются на ультракоротких волнах.

Помехоустойчивость дискретного канала связи определяется вероятностью ошибочного $P_{\text{ош}}$ приема сигналов. Передается один из двух сигналов $S_1(t)$ или $S_1(t)$. Формула, характеризующая вероятность ошибочного приема $S_1(t)$ (т.е. принятия решения о передаче $S_2(t)$, когда передавался $S_1(t)$), будет следующей

$$P_{\text{ош}} = \int_{-\infty}^{-0.5E_0} w(\eta)d\eta = 1 - F(\sqrt{\frac{E_0}{2N_0}}) = 1 - F(\sqrt{\frac{E_1 + E_2 - 2E_{12}}{2N_0}})$$

Из формулы следует, что вероятность ошибочного приема элементов двоичного сообщения тем меньше, чем больше эквивалентная энергия E_0 и чем меньше спектральная плотность мощности помех N_0.

Помехоустойчивое кодирование (англ. Error Correcting Coding, ECC) — процесс преобразования информации, предоставляющий возможность обнаружить и исправить ошибки, возникающие при передаче информации по каналам передачи данных. Процесс помехоустойчивого кодирования заключается во введении избыточности, т. е. для передачи информации используется код, у которого используются не все возможные комбинации, а только некоторые из них. Такие коды называют избыточными или корректирующими. Соответственно, процесс введения избыточности (преобразование информационных символов в кодовое слово) называется кодированием, а обратный процесс восстановления информации из кодового слова, возможно содержащего ошибки — декодированием. Еще одним важным свойством помехоустойчивого кодирования является эффект усреднения шума, который достигается за счет того, что избыточные символы зависят от нескольких информационных символов. В рамках цифровой системы передачи данных задачи кодирования и декодирования возложены на кодер и декодер соответственно.

В целом, способность помехоустойчивых кодов определять и исправлять ошибки — их корректирующие свойства — зависят от правил построения этих кодов и параметров кода (числа разрядов, избыточности и др.), а также от используемых алгоритмов декодирования. Основными параметрами, характеризующими корректирующие свойства кодов являются

1. Избыточность кода.
2. Кодовое расстояние.
3. Кратность гарантированно обнаруживаемых ошибок.
4. Кратность гарантированно исправляемых ошибок.

4.3 Избыточность корректирующего кода

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

$$r = n - k,$$

где n — число кодовых символов на выходе кодера, соответствующих k информационных символов на его входе.

Относительной избыточностью корректирующего кода называют величину $R_{\text{отн}} = \frac{r}{n} = \frac{(n-k)}{n} = 1 - \frac{k}{n}$.

С ней связана так называемая относительная скорость передачи информации или скорость кода, которая показывает, какую часть общего числа символов кодовой комбинации составляют информационные символы.

$\frac{k}{n} = 1 - R_{\text{отн}}$.

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

$$R = H \cdot \frac{k}{n}$$

Кодовое расстояние d или *расстояние Хемминга* 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.

$$10011 \oplus 11001 = 01010 \Rightarrow d = 2.$$

Кодовое расстояние может быть различным. Так, в первичном натуральном безызбыточном коде это расстояние для различных комбинаций может различаться от единицы до n, где n — значность/длина кода.

Для помехоустойчивого кода наиболее важным является *минимальное кодовое расстояние* d_{min} — наименьшее кодовое расстояние из всех между всеми парами кодовых комбинаций.

В безызбыточном коде все комбинации являются разрешенными, d_{min} = 1. Достаточно только исказиться одному символу, и будет ошибка в сообще- нии.

4.4 Классификация помехоустойчивых кодов

Помехоустойчивые коды классифицируются по различным признакам. Одной из основных классификаций является деление кодов на *блочные* и *непрерывные*.

Блочный (блоковый) код является кодом без памяти. Кодер блочного кода отображает подающийся на вход блок информационных символов длиной k в кодовую последовательность из n выходных символов. Термин «без памяти» указывает, что каждый блок из n символов зависит только от соответствующего блока из k символов и не зависит от других блоков.

Основыми параметрами блочных кодов являются *длина информационного блока* k, *длина кодового слова* n, *скорость кода* $\frac{k}{n}$ и *минимальное кодовое расстояние* d_{min} .

Непрерывные или древовидные коды — это коды, исправляющие ошибки, которые используют непрерывную, или последовательную, обработку информации короткими фрагментами (блоками). Чаще всего используются линейные древовидные коды, называемые сверточными. Кодер древовидного кода является устройством с памятью. На вход поступают наборы из k входных информационных символов, а на выходе появляются наборы из n кодовых символов. Каждый набор n кодовых символов зависит от текущего входного набора и от v предыдущих входных символов. Следовательно кодер должен содержать устройство памяти на $m = k + v$ входных символов. Параметр m часто называют длиной кодового ограничения кода.

Также непрерывные коды характеризуются скоростью кода $\frac{k}{n}$ и свободным расстоянием $d_{\text{св}}$.

Особое место в такой классификации занимают каскадные коды и турбо коды, представляющие из себя комбинации блочных и/или непрерывных кодов.

Другой подход к классификации делит коды на линейные и нелинейные. Линейные коды образуют векторное пространство. Два кодовых слова линейного кода при сложении по определенному правилу дают в результате третье кодовое слово.

Практически все применяемые на практике схемы кодирования основаны на использовании линейных кодов. Двоичные линейные блоковые коды часто называют *групповыми кодами*, так как их кодовые слова образуют математическую структуру, называемую *группа*.

По способу кодирования коды делятся на *систематические* и *несистематические*. В первом случае информационные символы передаются на выход декодера без изменения и к ним добавляются проверочные символы. В случае несистематического кодирования информационные символы в явном виде в кодовом слове отсутствуют.

Ещё одним вариантом деления помехоустойчивых кодов является разделение их на *коды, исправляющие случайные ошибки*, и коды, *исправляющие пакеты (пачки) ошибок*. Хотя для исправления пачек ошибок было разработано большое количество кодов с хорошими характеристиками, часто оказывается выгодным использовать коды, исправляющие случайные ошибки, совместно с устройствами перемежения/деперемежения. Также стоит отметить, что существуют алгоритмы декодирования, позволяющие использовать коды, рассчитанные для исправления случайных ошибок, для исправления пачек ошибок без использования перемежителей.

4.5 Линейные коды

Линейным (или групповым) двоичным кодом называется подмножество E^n, являющееся линейным подпространством (подгруппой) в E^n [17]. Произвольный линейный код с параметрами $[n, k, d]$ можно задать различными способами:

Пусть E^n обозначает n-мерное метрическое пространство всех двоичных векторов длины n с метрикой Хэмминга (ниже приведены примеры двумерных проекций E^n при малых n на рис. 1 и общепринятую модель E^n для произвольного n на рис. 8). Произвольное подмножество C пространства E^n называется двоичным кодом длины n, элементы кода называются кодовыми словами. В дальнейшем параметры линейного кода C длины n с кодовым расстоянием d будем обозначать через $[n, k, d]$, где k – размерность кода; для нелинейного кода C параметры будем обозначать через $(n, |C|, d)$.

Произвольный линейный код с параметрами $[n, k, d]$ можно задать различными способами: аналитически, с помощью одной формулы (такой способ задания линейного кода не всегда может найтись) посредством кодовой матрицы порядка $2^k \times n$ (строками матрицы являются кодовые слова); посредством порождающей матрицы порядка $k \times n$ (в строки записаны кодовые слова, образующие базу линейного кода); посредством проверочной матрицы – матрицы H такой, что для любого кодового слова $x=(x1, x2, \ldots, xn)$ выполняется

$$H\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = Hx^T = \mathbf{0}^{n-k},$$

здесь H – матрица порядка $(n-k)\times n$. Последнее уравнение задает $(n-k)$ проверочных уравнений. Очевидно, что такое представление линейного кода также является аналитическим.

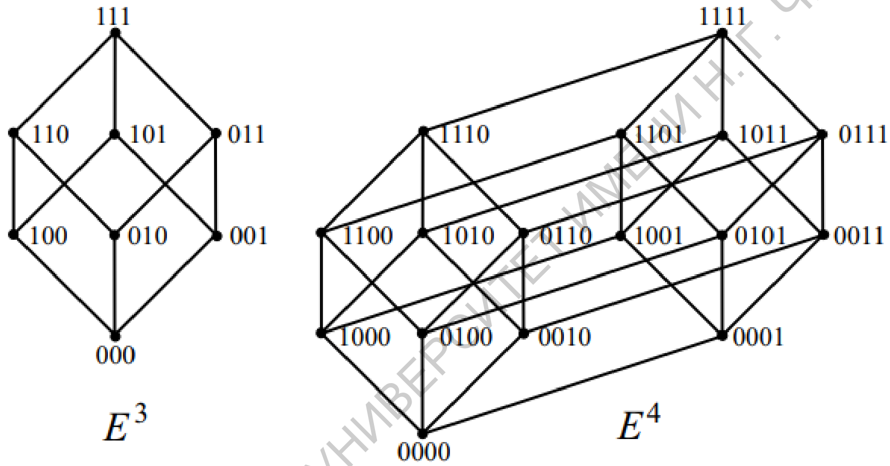

Рис.8. Двумерные проекции E^3 и E^4

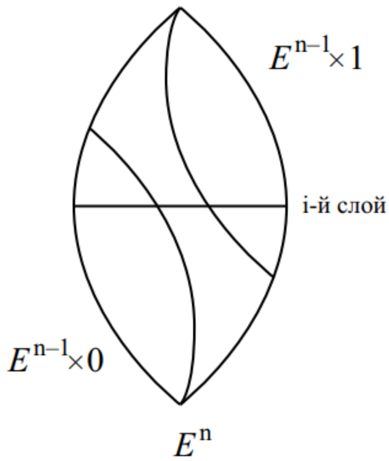

Рис.9. Двумерная проекция E^n

Следует отметить, что для данного линейного кода представление порождающей (проверочной) матрицей не единственно.

Опишем подробнее задание линейного кода посредством проверочной матрицы, имеющей канонический вид. Пусть от отправителя в кодер поступило сообщение $u = (u_1, u_2, \ldots, u_k)$. Сформируем кодовое слово $x=(x_1, x_2, \ldots, x_n)$. Положим первую часть кодового слова состоящей из символов самого сообщения (называемых информационными символами): $x_1 = u_1$, $x_2 = u_2, \ldots, x_k = u_k$. Далее следуют n − k символов, называемых проверочными $x_{k+1}, \ldots, x_n$. Они выбираются таким образом, чтобы все кодовые слова удовлетворяли уравнению

$$Hx^T = 0^{n-k}$$

4.6 Основные теоремы кодирования

Теорема 1. **О связи проверочной и порождающей матриц**. Если проверочная матрица линейного кода задана в каноническом виде $H = [A_{n-k,k}|E_{n-k}]$, то порождающая матрица этого кода имеет вид $G = [E_k| - A^T_{n-k,k}]$. Верно обратное.

Теорема 2. Граница Хэмминга. Для любого двоичного кода C длины n (не обязательно линейного) с кодовым расстоянием d выполняется неравенство $|C| \leq \frac{2^n}{\sum_{i=0}^{[\frac{d-1}{2}]} C_n^i}$.

Теорема 3. О столбцах проверочной матрицы. Если H – проверочная матрица кода длины n, то код имеет кодовое расстояние d тогда и только тогда, когда любые $d - 1$ столбцов матрицы H линейно независимы и найдутся d линейно зависимых столбцов.

Теорема 4. Граница Синглтона. Для любого линейного [n, k, d]-кода выполняется $n - k \geq d - 1$.

Код, достигающий границу Синглтона, называется *MDS-кодом*. Код, полученный из данного кода удалением одной или более координат во всех кодовых словах, называется *выколотым кодом*.

Теорема 5. Граница Синглтона для нелинейных q-значных кодов. Для любого $(n, M, d)_q$-кода выполняется $log_q M \leq n - d + 1$.

Теорема 6. Граница Плоткина. При $n < 2d$ для любого двоичного (n, M, d)-кода C справедливо неравенство $M \leq 2[d/(2d - n)]$,

где M – мощность кода C.

4.7 *Код Хэмминга*

Код Хемминга – это блочный код, позволяющий исправлять одиночные и фиксировать двойные ошибки, разработанный Ричардом Хеммингом в сороковых годах прошлого столетия [13, 14].

Для построения линейного кода Хэмминга с m проверками на четность, исправляющего одну ошибку, воспользуемся теоремой 3: определим код посредством проверочной матрицы, столбцами которой являются все ненулевые векторы длины m. Очевидно, что любые два столбца этой матрицы линейно независимы и найдутся три линейно зависимых столбца, следовательно по теореме 3 кодовое расстояние равно 3 и значит код исправляет одну ошибку. Этот код называется кодом Хэмминга, далее будем его обозначать H^n.

Параметры кода Хэмминга:

$$n = 2^m - 1, \qquad k = n - log(n+1), d = 3],$$

$m = log(n + 1)$ (здесь и всюду далее $log(\cdot)$ является двоичным логарифмом, если не оговорено особо).

Утверждение. Код Хэмминга H^n является совершенным кодом, исправляющим одну ошибку.

Доказательство. Код H^n исправляет одну ошибку (по определению кода). По построению его мощность равна

$$|H^n| = 2^{n-m} = \frac{2^n}{n + 1},$$

где $m = log(n + 1)$. Следовательно, он достигает границы Хэмминга (см. теорему 2) и потому является совершенным.

Идея кодов Хемминга заключается в разбиении данных на блоки фиксированной длины и вводе в эти блоки контрольных бит, дополняющих до четности несколько пересекающихся групп, охватывающих все биты блока.

Ричард Хемминг рассчитал минимальное количество проверочных бит, позволяющих однозначно исправлять однократные ошибки.

Если длина информационного блока, который требуется закодировать m бит. Количество контрольных бит, используемых для его кодирования k, то закодированный блок будет иметь длину: $n = m + k$ бит. Для каждого блока такой длины возможны n различных комбинаций, содержащих ошибку.

Таким образом, для каждого передаваемого информационного блока может существовать n–блоков, содержащих однократную ошибку, и один блок - без ошибок.

Следовательно, максимальное количество различных закодированных блоков, содержащих не больше одной ошибки, будет: $2^m(n+1)$, где $n = m + k$.

Если для информационных данных длиной m подобрать такое количество контрольных бит k, что максимально возможное количество различных последовательностей длиной $m+k$ будет больше или равно максимальному количеству различных закодированных информационных блоков, содержащих не больше одной ошибки, то точно можно утверждать, что существует такой метод кодирования информационных данных с помощью k контрольных бит, который гарантирует исправление однократной ошибки.

Следовательно, минимальное количество контрольных бит, необходимых для исправления однократной ошибки, определяется из равенства:

$2^m (n+1)=2^n$

Учитывая, что $n = m + k$, получаем:

$k=2^k – m – 1$

Так как количество бит должно быть целым числом, то k, вычисленное с помощью этого уравнения, необходимо округлить до ближайшего большего целого числа.

Например, для информационных данных длиной 7 необходимо 4 контрольных бита, чтобы обеспечить исправление однократных ошибок, а для данных длинной 128 бит необходимо 8 контрольных бит.

Мало определить минимальное количество контрольных бит, необходимых для исправления ошибки. Необходимо разработать алгоритм проверки данных с помощью этих контрольных разрядов. Ричард Хемминг предложил следующий алгоритм.

Все биты, порядковые номера которых являются степенью двойки, – это контрольные разряды. То есть если порядковый номер бита обозначить символом ‘n’, то для контрольных бит должно быть справедливо равенство: $n=2^k$, где $к$ – любое положительное целое число.

Например, для закодированной последовательности длиной 13 бит проверочными будут: 1, 2, 4 и 8 биты, так как $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$.

Каждый выбранный, таким образом, контрольный бит будет проверять определенную группу бит, т.е. в контрольный бит будет записана сумма по модулю два всех битов группы (дополнение до четного количества единиц), которую он проверяет.

Для того, чтобы определить какими контрольными битами контролируют бит, необходимо разложить его порядковый номер по степени 2. Таким образом, девятый бит будет контролироваться битами 1 и 8, так как $9 = 2^0 + 2^3 = 1 + 8$.

Рассмотрим пример кодирования бинарной последовательности данных, состоящей из семи элементов: 1001101.

1. Определим необходимое количество контрольных разрядов. Расчет будем вести по формуле: $k = 2^k - m - 1$, где k – количество контрольных разрядов, m – количество информационных разрядов. Так как количество бит должно быть целым числом, то k, вычисленное с помощью этого уравнения, необходимо округлить до ближайшего большего целого числа. Таким образом, число контрольных бит - 4.

Определим расположение проверочных бит в результирующей закодированной последовательности. Обозначим информационные биты символом И, а контрольные биты символом П. Индекс около этих символов будет означать их порядковый номер в закодированной последовательности.

Контрольные биты будут занимать четыре позиции с порядковыми номерами, равными степени двойки: 2^0, 2^1, 2^2, 2^3 => 1,2,4,8.

Определим, какие группы контролируют проверочные биты. Для этого разложим порядковые номера информационных бит по степени двойки:

$И_3$: $3 = 2^0 + 2^1 = 1 + 2$ => Информационный бит $И_3$ проверяется контрольными битами $П_1$ и $П_2$.

$И_5$: $5 = 2^0 + 2^2 = 1 + 4$ => Информационный бит $И_5$ проверяется контрольными битами $П_1$ и $П_4$.

$И_6$: $6 = 2^1 + 2^2 = 2 + 4$ => Информационный бит $И_6$ проверяется контрольными битами $П_2$ и $П_4$.

$И_7$: $7 = 2^0 + 2^1 + 2^2 = 1 + 2 + 4$ => Информационный бит $И_7$ проверяется контрольными битами $П_1$, $П_2$ и $П_4$.

$И_9$: $9 = 2^0 + 2^3 = 1 + 8$ => Информационный бит $И_9$ проверяется контрольными битами $П_1$ и $П_8$.

$И_{10}$: $10 = 2^1 + 2^3 = 2 + 8$ => Информационный бит $И_{10}$ проверяется контрольными битами $П_2$ и $П_8$.

$И_{11}$: $11 = 2^0 + 2^1 + 2^3 = 1 + 2 + 8$ => Информационный бит $И_{11}$ проверяется контрольными битами $П_1$, $П_2$ и $П_8$.

Рассчитаем значения контрольных бит. Для этого определим группы для всех контрольных бит, просуммируем их по модулю два, а результат запишем в соответствующие контрольные биты (дополним группы до четности единиц):

$П_1 = И_3 + И_5 + И_7 + И_9 + И_{11} = 1 + 0 + 1 + 1 + 1 = 0$

$П_2 = И_3 + И_6 + И_7 + И_{10} + И_{11} = 1 + 0 + 1 + 0 + 1 = 1$

$П_4 = И_5 + И_6 + И_7 = 0 + 0 + 1 = 1$

$П_8 = И_9 + И_{10} + И_{11} = 1 + 0 + 1 = 0$

Проверим на четность единиц все группы, контролируемые проверочными разрядами:

$П_1$: $П_1 + И_3 + И_5 + И_7 + И_9 + И_{11} = 0 + 1 + 0 + 1 + 1 + 1 = 0 = 0$

$П_2$: $П_2 + И_3 + И_6 + И_7 + И_{10} + И_{11} = 1 + 1 + 1 + 1 + 0 + 1 = 1 \neq 0$

$П_4$: $П_4 + И_5 + И_6 + И_7 = 1 + 0 + 1 + 1 = 1 \neq 0$

$П_8$: $П_8 + И_9 + И_{10} + И_{11} = 0 + 1 + 0 + 1 = 0 \neq 0$

Так как проверка четности единиц не прошла для групп контрольных битов $П_2$ и $П_4$, то ошибка содержится в битах, принадлежащих этим группам. Алгоритм Хемминга позволяет исправлять только одну ошибку, поэтому будем искать неисправность, исходя из предположения, что могла произойти только одна ошибка. Если это предположение - неверно, то все попытки исправить данные только привнесут дополнительные искажения информации.

Итак, если ошибка была только одна, то она должна быть в одном из битов, принадлежащих обеим группам. Это биты $И_6$ и $И_7$.

Для того, чтобы уточнить, в каком именно бите произошла ошибка, обратимся к группам, в которых проверка на четность прошла успешно, а, следовательно, в этих группах все биты - корректны. Как видите, к одной из исправных групп принадлежит бит $И_7$, а, следовательно, он верен.

Таким образом, очевидно, что ошибка произошла в бите $И_6$, и, инвертировав его значение, мы восстановим принятые данные.

Можно доработать описанный выше алгоритм, и сделать так, чтобы он не только исправлял однократные ошибки, но и фиксировал двойные. Для этого, после того, как информационные данные закодированы, к полученному коду приписывается еще один разряд, дополняющий его до четности единиц.

Если предположить, что количество ошибок - не более двух, то:

- Данные верны, если во всех контрольных группах количество единиц - четное, и общее количество единиц - тоже четное.

- Произошла однократная ошибка, если в некоторых контрольных группах количество единиц - нечетное, и общее количество единиц - нечетное.

- Ошибка в дополнительном контрольном разряде, если во всех контрольных группах количество единиц - четное, а общее количество единиц - нечетное.

- Двойная ошибка, если в некоторых контрольных группах количество единиц - нечетное, а общее количество единиц - четное.

Как видите, алгоритм коррекции ошибок Хемминга - достаточно прост и надежен. При этом эффективность кода растет при увеличении информационных блоков. Так, для кодирования 7 бит данных избыточность составляет чуть больше 57%, для кодирования 256 бит избыточность будет 3.5%, а для 1024 – 1%.

Алгоритм кодирования Хэмминга – очень популярен и позволяет значительно повысить надежность передачи и хранения информации. Особенно, он выгоден при кодировании больших блоков данных. Существует большое количество различных способов реализации этого алгоритма.

Помехоустойчивое кодирование не является обязательной операцией при передаче информации. Эта процедура может отсутствовать. Однако это может привести к очень существенным потерям в помехоустойчивости системы, значительному уменьшению скорости передачи и снижению качества передачи информации. Поэтому практически все современные системы (за исключением, быть может, самых простых) должны включать и обязательно включают помехоустойчивое кодирование данных.

4.8 Коды Рида-Маллера

Коды Рида-Маллера относятся к линейным двоичным кодам, имеющим большие кодовые расстояния и исправляющим благодаря этому много ошибок. Они пригодны для каналов с малым отношением сигнал/помеха. Этот класс кодов интересен и потому, что с ним связаны многие другие сигналы, применяемые в радиотехнических системах: ортогональные и биортогональные сигналы, симплексные коды, последовательности и коды Хэмминга.

Рассматрим простейшие коды Рида-Маллера, слова которых являются линейными комбинациями некоторых двоичных функций, обладающих полезными для практики свойствами. Функции выбраны такими, что их отображение в поле действительных чисел дает систему ортогональных функций. Это свойство используется при декодировании. Данное ограничение означает, что в базис кода не входят произведения двоичных функций, т.е. рассматривается код Рида-Маллера 1-го порядка.

Кодовое слово длины n обычно рассматривается как булева функция (или ее инверсия), заданная в 2^m точках, т.е. на наборах из m двоичных элементов. Можно многими способами нумеровать позиции кодового слова -разрядными двоичными векторами. Ясно, что, как в случае кодов Хэмминга, такая перестановка не влияет на помехоустойчивость получаемых кодов. Будем нумеровать позиции кодового слова числами в двоичной системе счисления $(v_1, v_2 \dots v_m)$, где $v_i = 0{,}1$ для $i = 1{,}2, \dots m$. Ввиду линейности кодов Рида-Маллера каждый символ кодового слова S_i представим линейной комбинацией

$$S_i = a_1v_1 + a_2v_2 + \ldots + a_mv_m$$

или ее инверсией

$$1 + S_i = a_0 1 + a_1v_1 + a_2v_2 + \ldots + a_mv_m$$

где $a_0a_1a_2 \ldots a_m -$ известные информационные символы.

В соответствии с определением порождающей матрицы и правилом покомпонентного сложения векторов элементы $(1, v_1, v_2 \ldots v_m)$ являются столбцами матрицы G. Для $m = 3$ порождающая матрица размера $m + 1 = 4$ на $n = 8$ имеет вид:

$$G = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{vmatrix}$$

Столбцы матрицы G без верхней строки представляют собой последовательность чисел, записанных в двоичной системе счисления (младшие разряды внизу). Таким образом, столбцы матрицы можно рассматривать как последовательность состояний двоичного суммирующего счетчика:

$$G = \begin{vmatrix} 1 \\ S_{\delta 1} \\ \cdots \\ S_{\delta m} \end{vmatrix},$$

где 1 – последовательность из единиц; $S_{\delta 1}$ – последовательность состояний последнего (старшего) разряда счетчика; $S_{\delta m}$ – последовательность состояний первого (младшего) разряда. Отметим, что перестановка столбцов и строк порождающей матрицы приводит к эквивалентным кодам.

Кодовое слово есть линейная комбинация базисных векторов (строк матрицы G):

$$S = (s_1s_2 \ldots s_n) = a_1 1 + a_2S_{\delta 1} + a_2S_{\delta 2} + \ldots + a_mS_{\delta m}$$

Вид матрицы указывает простой способ формирования базисных векторов и получения кодового слова. Схема кодирующего устройства для $m = 3$ (рис. 9) содержит трехразрядный двоичный счетчик, вырабатывающий функции $S_{\delta 1}, S_{\delta 2}, S_{\delta 3}$ и комбинационную схему, реализующую булеву функцию. Естественно, длительность информационных символов, подаваемых в этот кодер, предполагается равной длительности кодового слова, т.е. в данной случае 8 длительностям символов канала. Двоичный вектор $S = (s_1s_2 \ldots s_n)$ с компонентами $S_{\delta m}$ может быть отображен в вектор $W = (w_1, w_2, \ldots, w_n)$ с действительными компонентами $w_i = \pm 1$. Для этого надо «0» в двоичном векторе заменить на (+1), а «1» – на (–1). Такое отображение можно определить формулами:

$$W = (-1)^{s}, w_i = (-1)^{s_i} \tag{7}$$

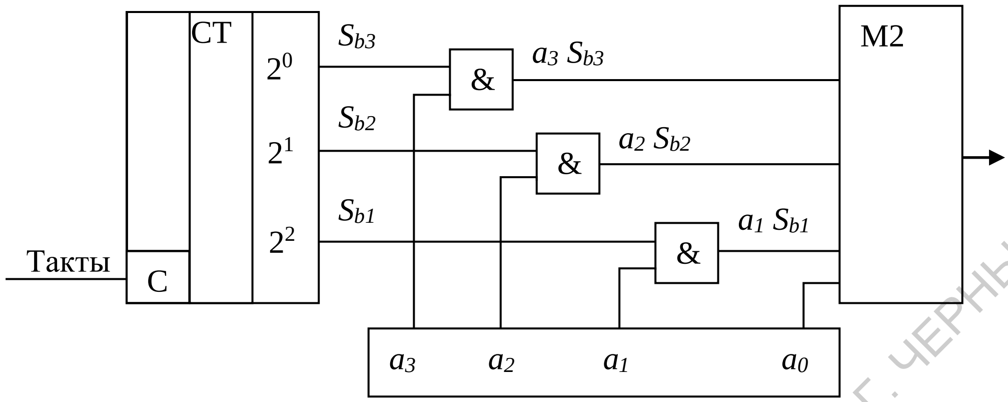

Рис.9. Кодер кода Рида-Миллера длина $n = 8$

В табл. 2 приведены все 16 кодируемых информационных последовательностей и соответствующие им кодовые слова. Обратим внимание, что кодовые слова правой половины таблицы являются инверсией слов левой половины. Тогда операции суммирования двоичных последовательностей будет соответствовать покомпонентное умножение последовательностей с элементами ±1. Можно говорить о том, что аддитивная группа двоичных векторов отображается в мультипликативную группу действительных векторов. Конечно, отображение (7) не требует для технического воплощения дополнительного оборудования, так как делается разработчиком аппаратуры мысленно и связано с разной трактовкой одних и тех же уровней напряжений (высоких и низких) в технических устройствах.

Таблица 2. Кодовые слова Рида-Маллера

Информационные символы	Кодовое слово	Информационные символы	Кодовое слово
0000	00000000	1000	11111111
0001	01010101	1001	10101010
0010	00110011	1010	11001100
0011	01100110	1011	10011001
0100	00001111	1100	11110000
0101	01011010	1101	10100101
0110	00111100	1110	11000011
0111	01101001	1111	10010110

Если применить отображение (7) к строкам матрицы, получим функции Радемахера:

$$R_1 = (-1)^{s_{\delta_1}}, R_2 = (-1)^{s_{\delta_2}}, \dots, R_m = (-1)^{s_{\delta_1}},$$

Известно, что всевозможные произведения функций Радемахера образуют полную ортогональную систему функций, которые называются функциями Уолша, и им соответствуют слова кода Рида-Маллера. Ортогональность функций означает, что для двух произвольных функций Уолша W_i и W_j, $i \neq j$ скалярное произведение

$$\left(W_i, W_j\right) = \sum_{k=1}^{n} w_{ik}\, w_{jk} = 0$$

т.е. число позиций, на которых символы последовательностей совпадают, равно числу позиций, на которых последовательности отличаются, и равно поэтому $\frac{n}{2}$. Из определения расстояния Хэмминга заключаем, что кодовое расстояние кода Рида-Маллера равно $\frac{n}{2}$. Отметим, что для пар слов, являющихся инверсией друг друга, расстояние равно n.

Таким образом, коды Рида-Маллера имеют длину $n = 2^m$, m+1 информационный символ, кодовое расстояние $d_0 = \frac{n}{2} = 2^{m-1}$. Отображение двоичных кодовых слов в область действительных чисел ± 1 дает множество функций Уолша, включающее 2^m функций Уолша W_i, $i = 1,2, \dots, n$, и 2^m противоположных функций $(-W_i)$, $i = 1,2, \dots, n$. Множество сигналов, составленных из функций Уолша и противоположных им, называется системой биортогональных сигналов. Если в систему не включать противоположные сигналы, то получим систему ортогональных сигналов, которые используются в качестве адресов абонентов в системах множественного доступа с кодовым разделением каналов. Применение ортогональных сигналов в качестве канальных позволяет разделять их в таких системах без взаимных помех.

Для кодов Рида-Маллера разработаны достаточно эффективные алгоритмы порогового (мажоритарного) декодирования. Здесь рассмотрим декодирование кодов Рида-Маллера по принципу максимума правдоподобия. Для симметричного канала это совпадает с декодированием по минимуму расстояния между векторами, при котором в качестве оценки переданного вектора S^c берется слово, ближайшее к принятому вектору Y. Имея в виду преобразование (7), рассмотрим коэффициент корреляции F_j между принятым вектором Y и функцией Уолша W_j. При $a_0 = 0$

$$F_{j0} = \left(Y, W_j\right) = \sum_{i=1}^{n} y_i w_{ij}$$

где y_i и w_{ij} принимают значения ± 1.

Поскольку при совпадении знаков y_i и w_{ij} их произведение равно 1, а при несовпадении – (–1), то

$$F_{j0} = n_C - n_{BC} = n_C - 2n_{BC} = n - 2d(Y, W_j)$$

где n_C и n_{BC} – соответственно числа совпадающих и несовпадающих символов в Y и W_j, а $n = n_C + n_{BC}$. При $a_0 = 1$, очевидно, получим

$$F_{j1} = 2d\left(Y, W_j\right) - n$$

Таким образом, оптимальный алгоритм декодирования предполагает следующие этапы:

1.Вычисление коэффициентов корреляции F_j между Y и функциями Уолша W_j, где $j = 1,2, \dots 2^m$.

2. Поиск максимального по абсолютной величине коэффициента $\left|F_j\right|_{max}$.

3. Принятие решения по правилу: $a_0^c = 0$, если $F_j > 0$, и $a_0^c = 1$, если $F_j < 0$.

Данная процедура представляет собой многоканальный корреляционный прием. Ее сложность пропорциональна числу n^2 операций сложения и вычитания. Разработаны быстрые алгоритмы декодирования кодов Рида-Маллера, по своей сути аналогичные алгоритмам быстрого преобразования Фурье.

4.9 Код Финка-Хагельберга.

Свёрточные коды (СК) относятся к непрерывным кодам [13, 14]. Выходные элементы, в данном случае, зависят от ряда предшествующих информационных элементов. Свёрточный код описывается тремя целыми числами (*n, k,K*). В данном случае *k* элементов, поступающих в кодер, порождают n элементов на его выходе. *K* – длина кодового ограничения, оно определяется числом разрядов (ячеек памяти) в кодирующем регистре сдвига. Кодовое ограничение определяет мощность и сложность кода. Выходные элементы СК зависят не только от текущего входного элемента, но и от (*K-1*) предыдущих, то есть СК имеет память. Основными элементами сверточного кода являются: регистр сдвига, сумматор по модулю 2, коммутатор.

Регистр сдвига - это динамическое запоминающее устройство, хранящее двоичные символы 0 и 1. Память кода определяет число триггерных ячеек m в регистре сдвига. Когда на вход регистра сдвига поступает новый информационный символ, то символ, хранящийся в крайнем правом разряде, выводится из регистра и сбрасывается. Остальные символы перемещаются на один разряд вправо и, таким образом, освобождается крайний левый разряд куда будет поступать новый информационный символ.

Сумматор по модулю 2 осуществляет сложение поступающих на него символов 1 и 0. Правило сложения по модулю 2 таково: сумма двоичных символов равна 0, если число единиц среди поступающих на входы символов четно, и равно 1, если это число нечетно.

Коммутатор последовательно считывает поступающие на его входы символы и устанавливает на выходе очередность кодовых символов в канал связи. По аналогии с блоковыми кодами, сверточные коды можно классифицировать на систематические и несистематические.

Значения выходного вектора для текущего элемента и предшествующих, можно найти поразрядно суммируя (по модулю два) коэффициенты ячеек, содержащих единицы.

$$V = x_i g_0 \oplus x_{i+1} g_1 \oplus x_{i+2} g_2$$

Матрицы коэффициентов и полиномов являются взаимно транспонированными. В этом случае выходной вектор можно получить путем умножения входного вектора на транспонированную матрицу полиномов.

При сверточном кодировании преобразование информационных последовательностей в выходные и кодовые происходит непрерывно.

Число информационных символов, поступающих за один такт на вход кодера - k. Число символов на выходе кодера - n, соответствующих k, поступившим на вход символам в течение такта. Скорость кода определяется отношением $R=k/n$ и характеризует избыточность, вводимую при кодировании.

Память кода (входная длинна кодового ограничения или информационная длина кодового слова), определяется максимально степенью порождающего многочлена в составе порождающей матрицы. Вес w двоичных кодовых последовательностей определяется числом "единиц", входящих в эту последовательность или кодовые слова.

Сверточный кодер принадлежит к классу устройств называемых конечными автоматами. Конечные автоматы это системы обладающие памятью о прошлых событиях (сигналах). При этом число состояний, в которых может находиться система - конечно. Состояние отражает информацию о прошлых событиях и определяет возможное поведение системы в будущем.

Состояние должно содержать минимум информации о прошлом на основание которой, совместно с текущими входными данными можно определить данные на выходе. Из каждого текущего состояния возможны переходы лишь в некоторые из состояний и не возможны в другие. Для СК со скоростью 1/n, состояний могут быть представлены содержимым (К-1) младших ячеек регистра. Поступление следующего элемента будет определять, как переход в следующее состояние, так и элементы на выходе кодера.

Для блочных кодов исправляющая способность есть номинальное количество ошибок в блоке длиной n, которое может быть гарантированно исправлено. Данная способность однозначно определяется кодовым расстоянием:

$$t_e = \frac{d_0 - 1}{2}$$

Для сверточного кода (СК) исправляющая способность не может быть задана так однозначно. Данная способность определяется через просвет или свободное расстояние кода по аналогичной формуле:

$$t_{u_{\text{СК}}} = \frac{d_f - 1}{2} \text{ или } \frac{d_f}{2} - 1 \text{ для чётных}$$

где d_f – просвет или свободное расстояние кода. При декодировании по принципу максимального правдоподобия СК способен исправить заданное количество ошибок в пределах нескольких длин кодовых ограничений («несколько» - это от 3 до 5 К). Точное значение зависит от распределения ошибок. Просветом или свободным расстоянием называется минимальный вес пути, начинающегося и заканчивающегося в нулев.

Систематически свёрточный код – это код, содержащий в своей выходной последовательности кодовых символов породившую ее последовательность информационных символов. Иначе код называют несистематическим. У систематических кодов при той же длине кодового ограничения и скорости кода свободное расстояние меньше. А значит, меньше и исправляющая способностью.

4.10 Рекуррентное кодирование Финка – Хагельбаргера:

Идея рекуррентного кодирования принадлежит Л. М. Финку (1955 г). Существенный вклад в развитие такого рода кодирования внесен Хагельбаргером, Возенкрафтом, Килмером, Элспасом и рядом других авторов. Эти коды в основном предназначены для коррекции пачек ошибок, хотя и могут с успехом применяться для коррекции независимых ошибок.

Отличительная черта рекуррентного кодирования от всех ранее рассмотренных методов заключается в том, что здесь операции кодирования и декодирования осуществляются над непрерывной последовательностью символов.

В рекуррентных кодах, как и ранее, проверочные символы получаются из информационных с помощью линейных операций. Однако проверочные символы располагаются вперемежку с информационными так, что на каждые *n* переданных символов приходится *m* информационных. Такие коды принято характеризовать отношением *m/n*, что непосредственно выражает скорость передачи.

Например, запись (1/2) означает, что на каждый информационный символ приходится один проверочный. В этом случае рекуррентная последовательность (по Финку) имеет вид: x_1y_1 x_2y_2 x_3y_3 x_4y_4 … x_jy_j, где x_j– информационный символ, y_j –проверочный символ, $y_j = x_{j-\gamma} \otimes x_{j+\gamma+1}$. Число γ называют шагом рекуррентного кода. Оно играет важную роль и определяет практически все основные свойства рекуррентной последовательности.

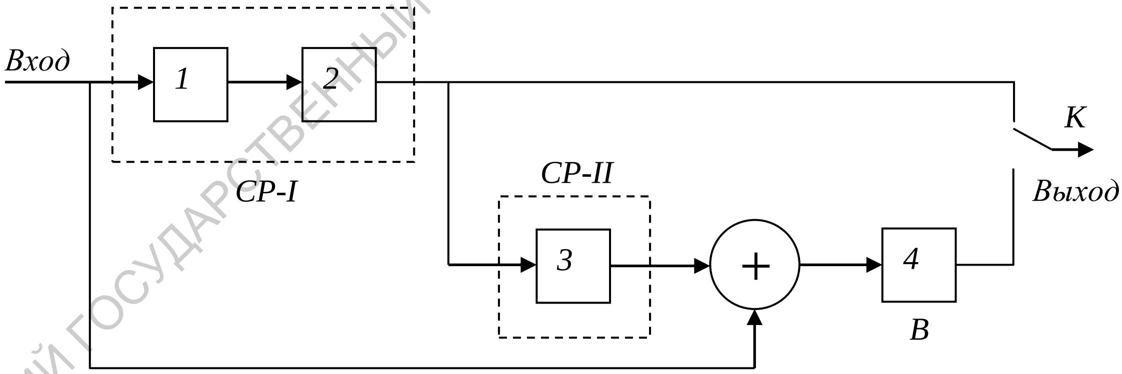

Рис. 10. Устройство для получения рекуррентной последовательности Финка $\frac{m}{n} = \frac{1}{2}$.

Кодирующее устройство для случая $\gamma = 1$ представлено на рис. 10. Оно содержит два сдвигающих регистра *СР-I* и *СР-II*. Первый из них имеет две $(\gamma + 1)$, а второй — одну γ ячейку памяти. Ячейка памяти, помеченная на рис. 4 буквой “*B*”, играет вспомогательную роль. Она подключается к *СР-II* через сумматор по модулю два.

Первоначально ключ *К* устанавливается в положение *1* и на вход кодера подается информационный символ x_1. Одновременно осуществляется сдвиг вправо (на один шаг) символов, записанных в ячейках *СР-I* и *СР-II*. В результате в канал поступает символ из второй ячейки *СР-I* (в данный момейт — символ 0, так как предполагается, что в начальный момент все ячейки содержали нули).

Одновременно в ячейку “*В*” записывается сумма по модулю два входного символа x_1 и символа, хранившегося в последней ячейке *СР-II* (в данный момент запишется символ x_1).

После выполнения этих операций ключ *К* переводится в положение 2 и производится считывание символа из ячейки “*В*” (в канал выдается проверочный символ, который в данном случае совпадает с x_1). Затем ключ *К* перебрасывается в положение *1* вход кодера подается информационный символ x_2 и все указанные операции повторяются.

Таблица 3. Рекуррентное кодирование.

Вход	Состояние ячеек памяти СР				Выход
	1	2	3	4	
x_1	x_1	0	0	x_1	0
–	x_1	0	0	0	x_1
x_2	x_2	x_1	0	x_2	0
–	x_2	x_1	0	0	x_2
x_3	x_3	x_2	x_1	x_3	x_1
–	x_3	x_2	x_1	0	x_3
x_4	x_4	x_3	x_2	$x_1+ x_4$	x_2
–	x_4	x_3	x_2	0	$x_1+ x_4$
x_5	x_5	x_4	x_3	$x_2+ x_5$	x_3
–	x_5	x_4	x_3	0	$x_2+ x_5$
x_6	x_6	x_5	x_4	$x_3+ x_6$	x_4
–	x_6	x_5	x_4		$x_3+ x_6$
…					x_5
…					$x_4+ x_7$
…					x_6
…					$x_5+ x_8$
…					x_7

Начиная с седьмого такта работы кодера (табл. 3) образуется рекуррентная последовательность. Вопрос о декодировании рекуррентной последовательности может быть решен многими способами.

На рис. 5 представлена блок-схема декодера, принцип работы которого сводится к следующему. В сдвигающий регистр, содержащий $4 \times (2\gamma + 1)$ ячеек памяти (в нашем случае 12) вводятся символы так, что в крайней справа ячейке располагается информационный символ (в данной ситуации $\widetilde{x_1}$).

При указанном на рис. 11 расположении символов будет осуществляться декодирование символа $\tilde{x}_4$.

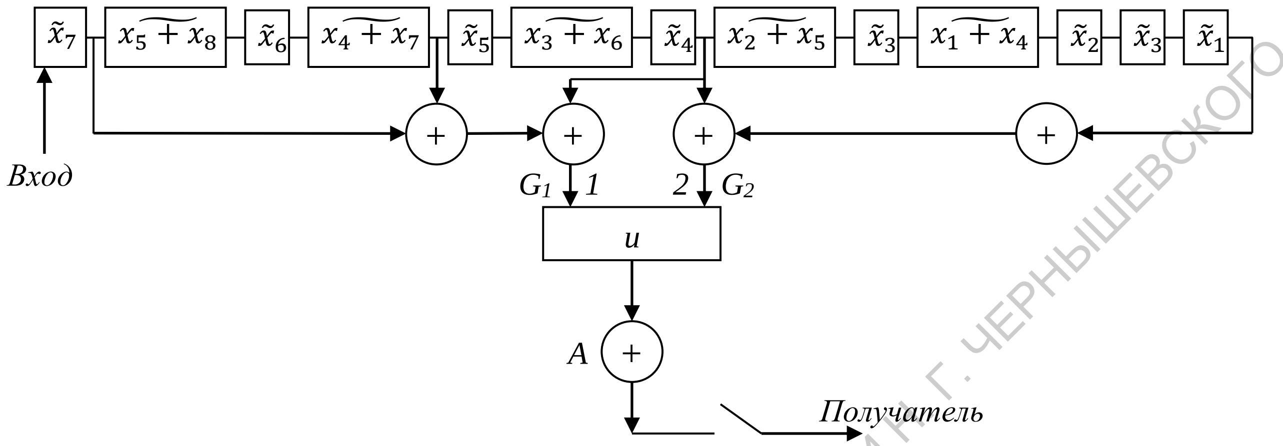

Рис. 11. Блок-схема декодера рекуррентной последовательности Финка, основанного на проверке на четность.

При приеме следующего символа $\tilde{x}_7$ (который также будет информационным) замыкается ключ K и производится сдвиг символов, ранее записанных в регистре, на один шаг вправо. В блоке сумматоров по модулю два осуществляется проверка на четность, т. е. определяются элементы контрольного числа (синдрома) $G_1 = \tilde{x}_7 \otimes \tilde{y}_5 \otimes \tilde{x}_4$ и $G_2 = \tilde{x}_4 \otimes \tilde{y}_2 \otimes \tilde{x}_1$.

Результат проверки G_1 и G_2 на четность подается на каскад совпадений. Если трансформированным оказался символ x_4 то на обоих входах каскада появятся единицы ($G_1 = 1$ и $G_2 = 1$) и на выходе сумматора *“A”* образуется символ, противоположный $\tilde{x}_4$ т. е. x_4. Ошибка исправлена.

Если же на один или одновременно на оба входа каскада совпадений поступают нули (удовлетворяется одна или обе проверки на четность), то на выходе каскада совпадений образуется нуль, и, следовательно, символ $\tilde{x}_4$ будет выдан получателю таким, каким он принят.

После выполнения описанных операций ключ K размыкается, производится сдвиг всех символов вправо на один шаг и в регистр вводится следующий символ рекуррентной последовательности (он будет проверочным - $\tilde{y}_7$)

Затем ключ K снова замыкается, и процедура декодирования осуществляется применительно к следующему символу $\tilde{x}_5$.

Информационный символ $\tilde{x}_j$ (в частности $\tilde{x}_4$) окажется неправильно декодированным либо в случае, когда $\tilde{x}_4 \neq x_4$, а на одном из входов схемы "*И*" образовался символ *0*, либо когда $\tilde{x}_4 = x_4$ но обе проверки на четность оказались невыполненными.

Легко заметить, что символ $\tilde{x}_j$ декодируется правильно, если «длина» пакета не превосходит 3 (в общем случае $2\gamma + 1$), а следующие за ним 9 символов приняты правильно [в общем случае $3 \times (2\gamma + 1)$, а иногда $3 \times (2\gamma + 1) + 1$].

Описанные схемы декодеров, конечно, далеко не исчерпывают всех возможных. Например, можно осуществлять исправление ошибок в принятой последовательности на основе схем, в которых с помощью коммутатора принятые символы сначала рассортировываются на информационные и проверочные, а затем по информационным находятся новые проверочные символы *, осуществляется проверка на четность и в случае необходимости проводится исправление соответствующего информационного символа. В таких схемах, как и ранее, могут быть использованы и метод независимых решений, и метод последовательного исключения независимых переменных.

Список литературы:

1. Белов, В.М. Теория информации. Курс лекций: Учебное пособие для вузов / В.М. Белов, С.Н. Новиков, О.И. Солонская. - М.: ГЛТ, 2012. - 143 с.
1. Малюк, А.А Теория защиты информации / А.А Малюк. - М.: ГЛТ, 2014. - 184 с.
2. Осокин, А.Н. Теория информации: Учебное пособие для прикладного бакалавриата / А.Н. Осокин, А.Н. Мальчуков. - Люберцы: Юрайт, 2016. - 205 с.
3. Н. Винер, А.Н. Колмогоров, В.А. Котельникова, К. Шеннон, Курс теории информации
4. Лидовский В.И. Теория информации. - М., «Высшая школа», 2002г. - 120с.
5. Метрология и радиоизмерения в телекоммуникационных системах. Учебник для ВУЗов. / В.И. Нефедов, В.И. Халкин, Е.В. Федоров и др. - М.: Высшая школа, 2001 г. - 383с.
6. Цапенко М.П. Измерительные информационные системы. - М.: Энергоатом издат, 2005. - 440с.
7. Зюко А.Г., Кловский Д.Д., Назаров М.В., Финк Л.М. Теория передачи сигналов. М: Радио и связь, 2001 г. -368 с.

8. Б. Скляр. Цифровая связь. Теоретические основы и практическое применение. Изд. 2-е, испр.: Пер. с англ. - М.: Издательский дом «Вильямс», 2003 г. - 1104 с.
9. А.Ю. Тропченко, А.А. Тропченко. Методы сжатия изображения, аудиосигналов и видео. СПб: СПбГУ ИТМО, 2009. С. 108.
10. Я. Ричардсон, Видеокодирование H.264 и MPEG-4 – стандарты нового поколения. М.: Техносфера. 2005. С.368.
11. Вологдин Э.И., Гласман К.Ф., Ковалгин Ю.А., Лишин Л.Г. Запись аудио-и видеосигналов: учебник для вузов.- М.: Издательский центр «Академия, 2010.-512 с.
12. Ковалгин Ю.А., Вологдин Э.И., Кацнельсон Л.С. Стереофоническо радиовещание и звукозапись. - М.: Горячая линия - Телеком, 2007.
13. Блох Э.Л., Зяблов В.В. Обобщенные каскадные коды. М.: Связь, 1976.
14. Петерсон У., Уэлдон Э. Коды, исправляющие ошибки. Мир, 1976.
15. Маскаева, А. М. Основы теории информации: учебное пособие для студ. учрежд. СПО. - М.: Форум: ИНФРА-М, 2014..
16. Дворецкий И.М., Дриацкий И.Н. Цифровая передача сигналов звукового вещания. – М.: Радио и связь, 1987.
17. Мак-Вильямс Ф. Дж., Слоэн Н. Дж.А. Теория кодов, исправляющих ошибки. Пер. с англ. — М.: Связь, 1979. — 744 с.
18. Теория электрической связи: учебное пособие / К.К. Васильев, В.А. Глушков, А.В. Дормидонтов, А.Г. Нестеренко; под общ. ред. К.К. Васильева. – Ульяновск: УлГТУ, 2008.
19. Зюко А.Г., Кловский Д.Д., Коржик В.И., Назаров М.В. Теория электрической связи. - М.: Радио и связь, 1999. - 432 с.
20. Новиков Ф.А. Дискретная математика для программистов. – Санкт–Петербург; Изд.дом «Питер», 2008.