

Министерство образования и науки РФ

**Федеральное государственное бюджетное образовательное
учреждение высшего образования**

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Радиотехнический факультет (РТФ)

Кафедра радиотехнических систем (РТС)

СБОРНИК ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ ПО ОБЩЕЙ ТЕОРИИ СВЯЗИ

**Учебно-методическое пособие для самостоятельной работы
студентов, обучающихся по направлениям подготовки**

"Радиотехника" и

"Инфокоммуникационные технологии и системы связи"

РАЗРАБОТЧИК

доц. каф. РТС, к.т.н.,

_____ А. В. Новиков

Новиков А.В. Сборник индивидуальных заданий по дисциплине "Общая теория связи", ТУСУР, 2020.

60 с.

Сборник является значительно переработанной версией кафедрального сборника контрольных работ, опубликованного в виде приложения к учебному пособию [1]. Сборник содержит 7 заданий, которые охватывают 4 темы в порядке нарастания сложности. К каждой теме прилагается теория с необходимыми и достаточными для выполнения заданий формулами. Сборник включает [генератор заданий](#) и проверочные модули для проверки ответов. Программные компоненты написаны на языке *Python*. Использовались модули *OpenPyXL*, *SciPy*, *NumPy* и *Pytils*.

Оглавление

1. Общие требования к выполнению заданий.....	4
2. Линейные блочные коды.....	5
2.1. Определение характеристик кода по порождающей матрице.....	5
а) Сведения из теории.....	5
б) Задание, код 01.....	9
2.2. Определение характеристик кода по проверочной матрице.....	9
а) Сведения из теории.....	9
б) Задание, код 02.....	16
2.3. Декодирование линейного кода.....	16
а) Сведения из теории.....	16
Задача декодирования.....	16
Определение информационного вектора по кодовому.....	16
Коррекция ошибок в принятом векторе.....	18
б) Задание, код 03.....	22
2.4. Неравенство Хемминга.....	23
а) Сведения из теории.....	23
б) Задание, код 04.....	26
3. Вероятность ошибки при оптимальном приеме цифрового сигнала.....	26
а) Сведения из теории.....	26
б) Задание, код 05.....	38
4. Регенерация цифрового сигнала при передаче на большие расстояния.....	39
а) Сведения из теории.....	39
б) Задание, код 06.....	43
5. Кодирование источника.....	44
5.1. Коды Хаффмана и Шеннона-Фано.....	44
а) Сведения из теории.....	44
Коды Хаффмана.....	48
Коды Шеннона-Фано.....	50
Кодирование объединенных символов.....	51
б) Задание, код 07.....	53
6. Литература.....	53
7. Приложение А. К расчету вероятности ошибки при некогерентном приеме АМн-сигнала.....	54
8. Приложение Б. Вероятность символьной ошибки при когерентном приеме ФМн-8 в канале с АБГПШ.....	56

1. Общие требования к выполнению заданий

Каждому студенту выдается вариант задания со случайными параметрами в виде файла Excel, в котором заведены поля для ввода ответов. Требуется:

- Оформить ход решения и отсканировать его в формат jpeg. Имя файла скопировать с имени файла-задания Excel;
- Заполнить поля в файле найденными ответами и сдать файл с ответами.

Ход решения будет проверен при условии, что в файле с ответами все ответы будут правильными. При наличии верных ответов и файла с ходом решения будет проставлен 1 балл за ответы и баллы от 0 до 9 за ход решения, что в итоге даст диапазон от 1 до 10 баллов за одно выполненное задание. При отсутствии верных ответов ставится 0 баллов и студент должен перепроверить свое решение и отправить на проверку еще раз. Максимальное количество таких итераций определяется индивидуально преподавателем. Пересчет баллов в традиционную оценку делается по следующей шкале

0 → 2,

1...5 → 3,

6...8 → 4,

9...10 → 5.

При необходимости студент может самостоятельно генерировать задания и проверять ответы, https://github.com/nawww83/generator_idz.

2. Линейные блочные коды

2.1. Определение характеристик кода по порождающей матрице

а) Сведения из теории

Линейные блочные коды являются помехоустойчивыми кодами, предназначенными для обнаружения и исправления символьных ошибок, возникающих в канале передачи информации с некоторой вероятностью p . Обнаружение и исправление ошибок происходит за счет добавления к информационным символам проверочных символов. Проверочные символы при линейном блочном кодировании добавляются по правилам линейной алгебры

$$\vec{s} = \vec{a} \mathbf{G} . \quad (1)$$

Здесь вектор \vec{a} является строкой, состоящей из k информационных символов. Матрица \mathbf{G} называется **порождающей матрицей** линейного блочного кода¹. Матрица \mathbf{G} состоит из k строк и n столбцов, поэтому ее размер $k \times n$. Результатом процедуры кодирования (1) является вектор \vec{s} , состоящий из n символов. Параметр n называется **длиной кода**. Всегда $n > k$, поэтому разность $n - k > 0$ определяет число проверочных символов кода, которое обозначается как $r = n - k$. Отношение $R = k/n$ называется **скоростью кодирования**, $0 < R \leq 1$. Скорость кодирования играет ключевую роль при оценке корректирующей способности кода. Если $R = 1$, то кодирование отсутствует.

Будем рассматривать двоичные коды, и вместо слова "символ" говорить "бит"².

Приведем пример задания линейного (n, k) кода порождающей матрицей \mathbf{G} , а также пример определения кодовой таблицы, которая показывает взаимно-однозначное соответствие между входом \vec{a} и выходом \vec{s} кодера³. Пусть порождающая матрица кода имеет вид

1 Далее слово "блочного" будем опускать и говорить "линейного кода"

2 Бит — двоичный символ

3 Кодер — кодирующее устройство, кодирующий блок (алгоритм, программа)

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \quad (2)$$

Определяем ее размер как (3×6) — три строки и шесть столбцов. Результат кодирования (1) всех комбинаций информационного вектора \vec{a} показан в табл. 1. Примечательно, что информационным векторам с одной единицей соответствуют кодовые векторы, являющиеся строками порождающей матрицы (в таблице фон этих строк затемнен). Остальные кодовые векторы — сумма тех строк порождающей матрицы, которые соответствуют единицам в информационном векторе. Например, вектор 011 кодируется суммой двух нижних строк порождающей матрицы

$$101010 + 001101 = 100111.$$

Здесь работает правило умножения матриц, которое в сжатом виде гласит как "строка на столбец". Строка 011 поворачивается на 90 градусов так, что 0 становится напротив первой строки матрицы, а 1 — напротив последней. Говоря языком математики, кодирование (1) представляет собой линейную комбинацию строк порождающей матрицы, причем коэффициентами линейной комбинации являются информационные символы. Векторы складываются поэлементно, "по модулю два" (операция xor), переносы разрядов отсутствуют в принципе.

Таблица 1: Кодовая таблица линейного кода (6, 3)

Информационный вектор	Кодовый вектор	Вес w кодового вектора
000	000000	0
001	001101	3
010	101010	3
011	100111	4
100	111100	4
101	110001	3
110	010110	3
111	011011	4

Нулевой вектор является особенным. Он присутствует в любом линейном коде и является опорным, с которым сравниваются все остальные. Векторы сравниваются с помощью такой метрики как расстояние. Так как опорный вектор нулевой, то расстояние от некоторого кодового вектора до опорного совпадает с

количеством единиц в этом векторе — весом. В табл. 1 вычислен вес w всех кодовых векторов. Вес, определенный как "количество ненулевых элементов в векторе", называют **хемминговым весом** (весом по Хеммингу), в честь Ричарда Хемминга — американского исследователя помехоустойчивых кодов. Оказывается, что именно вес кодовых векторов линейного кода играет ключевую роль в плане определения корректирующей способности кода. Это так, потому что корректирующая способность любого кода зависит от того, насколько далеко друг от друга расположены кодовые векторы — это естественный принцип построения помехоустойчивых кодов: чем дальше друг от друга расположены кодовые векторы, тем выше вероятность исправить некоторую ошибку. В свою очередь расстояние d_{ab} между двоичными векторами a и b равно весу w от суммы (xor) этих векторов⁴, $d_{ab} = w(a+b)$, а для линейного кода сумма двух кодовых векторов дает кодовый вектор, поэтому если векторы принадлежат коду C , т. е. $a, b \in C$, то и суммарный вектор c также принадлежит коду, т. е. $a+b=c \in C$ и $d_{ab} = w(c)$.

Вообразим пространство, образуемое некоторой решеткой, в узлах которой закреплены "электроны" и "дырки". Электронам соответствуют концы разрешенных векторов, а дыркам — концы запрещенных. Расстояния между узлами в таком пространстве измеряются минимальным количеством ребер, которые требуется пройти, чтобы попасть из одного узла в другой. Если два вектора отличаются единственным элементом, то расстояние между ними равно единице, если двумя элементами, то — двум, и т. д. Такое расстояние называют **хемминговым расстоянием**. Чтобы сравнить два двоичных кодовых вектора, их следует сложить "по модулю два" и подсчитать вес w , например

$$w(110001+010110)=w(100111)=4.$$

Здесь мы просуммировали кодовые векторы под номерами 6 и 7, в результате чего получили кодовый вектор под номером 4 (те же операции выполняются и над информационными векторами). Таким образом, если код линейный, то нет необходимости в попарном сравнении кодовых векторов — достаточно подсчитать их вес и все множество расстояний будет найдено.

4 Сумма xor эквивалентна разности по модулю два

Слабым местом любого корректирующего кода являются два самых близких кодовых вектора. Расстояние между такими векторами определяет **минимальное расстояние кода** или, проще говоря, **кодвое расстояние**. Очевидно, что для линейного кода кодвое расстояние d_k равно весу самого "легкого" кодового вектора, отличного от нуля

$$d_k = \min_{\vec{s} \in C, w(\vec{s}) \neq 0} w(\vec{s}) . \quad (3)$$

Здесь C обозначает множество⁵ векторов линейного кода. Для рассматриваемого кода, табл. 1, кодвое расстояние равно трем. Набор весов кода удобно сжать и представить в виде словаря {ключ: значение}

$$W_{sp} = \{w : N_w\} , \quad (4)$$

в котором указаны вес w кодового вектора и количество N_w векторов с таким весом. Формула (4) задает **весовой спектр кода**. Словарь должен быть отсортирован по ключу — в данном случае по весу — в порядке возрастания.

Для кода C в табл. 1 весовой спектр имеет вид

$$W_{sp}(C) = \{0: 1, 3: 4, 4: 3\} .$$

Здесь спектр состоит из трех элементов. Весовой спектр, в частности, позволяет определить кодвое расстояние линейного кода как значение ключа второго элемента. Отметим, что сумма всех значений N_w в спектре равна количеству кодовых векторов кода 2^k . Также весовой спектр кода показывает кратности ошибок, которые не может обнаружить код и их количество. Однако, это будет понятно после изучения проверочной матрицы кода (смотри далее).

На основании кодового расстояния d_k определяются кратности ошибок, которые код может гарантированно обнаружить и исправить. Кратность ошибки — это количество ошибочных элементов в принятом векторе \vec{v} . Ошибку удобно моделировать вектором ошибки $\vec{e} = \vec{v} - \vec{s}$, тогда кратность ошибки q определяется хемминговым весом вектора ошибки, $q = w(\vec{e})$. Кратность гарантированного обнаружения

$$q_0 \leq d_k - 1 .$$

⁵ Множество — это набор уникальных элементов, т. е. совпадающих элементов в множестве нет по определению.

Кратность гарантированного исправления⁶

$$q_{\text{и}} \leq \left\lfloor \frac{d_{\text{к}} - 1}{2} \right\rfloor .$$

Для кода C в табл. 1 соответствующие кратности равны 2 и 1, соответственно. Таким образом, с помощью рассмотренного кода можно обнаруживать все ошибки кратности не выше второй и при этом исправлять все ошибки кратности не выше первой.

б) Задание, код 01

Случайным образом задается порождающая матрица G . Требуется определить:

- параметры линейного кода: n , k , r ;
- кодовую таблицу кода C ;
- весовой спектр кода $W_{\text{сп}}(C)$;
- кодовое расстояние кода, $d_{\text{к}}$;
- кратности гарантированного обнаружения и исправления, $q_{\text{о}}$ и $q_{\text{и}}$.

2.2. Определение характеристик кода по проверочной матрице

а) Сведения из теории

Найдем проверочную матрицу линейного кода, заданного порождающей матрицей (2)

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} .$$

Для этого в порождающей матрице найдем три разных столбца с единственной единицей, т. е. три столбца с единичным весом. В данном случае это столбцы под номерами 2, 5 и 6

⁶ При вычислении кратности исправления результат деления на два следует округлить вниз. Это округление типа "floor" – от англ. "пол", "низ".

$$\mathbf{G} = \begin{pmatrix} 1 & \mathbf{1} & 1 & 1 & 0 & 0 \\ 1 & \mathbf{0} & 1 & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{0} & 1 & 1 & \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

Будем такие столбцы называть опорными. Если в явном виде опорные столбцы обнаружить не удастся, то следует прибегнуть к суммированию строк. Например, если бы мы сложили первые две строки матрицы и записали результат на место первой строки, то опорными столбцами стали бы столбцы под номерами 1, 2 и 6. Какие строки суммировать — дело случая, т. к. правильных ответов несколько, и выбирать можно любой. Такие преобразования порождающей матрицы будут эквивалентными: после них множество кодовых векторов $\vec{s} \in C$ не меняется! Информационные векторы \vec{a} здесь игнорируются. Опорные столбцы обязаны найтись, ибо в противном случае код будет задан некорректно. Опорные столбцы — это своего рода базис. Далее группируем опорные столбцы слева так, чтобы они образовали единичную матрицу $\mathbf{I}^{(3)}$ размером 3×3 ⁷

$$\mathbf{G}_{\text{сист.}} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & 1 & 1 & 1 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & 1 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & 0 & 1 & 1 \end{pmatrix} = [\mathbf{I}^{(3)}, \mathbf{Q}] = [\mathbf{I}^{(k)}, \mathbf{Q}]. \quad (5)$$

Здесь индексом **сист.** обозначено то, что такой структуре матрицы соответствует **систематический** код, в котором информационные символы отделены от проверочных (в данном случае информационные сгруппированы слева). Прделанной группировке столбцов соответствует следующая перестановка

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 5 & 6 & 2 & 3 \end{pmatrix}. \quad (6)$$

Здесь в первой строке указаны номера столбцов матрицы \mathbf{G} , а во второй — номера столбцов в матрице $\mathbf{G}_{\text{сист.}}$, куда будут вставлены соответствующие столбцы матрицы \mathbf{G} . Например, второй столбец \mathbf{G} стал первым столбцом $\mathbf{G}_{\text{сист.}}$. Далее на основании (5) формируем проверочную матрицу по следующему правилу

⁷ В общем случае размером $k \times k$

$$\mathbf{H}_{\text{сист.}} = [-\mathbf{Q}^T, \mathbf{I}^{(r)}] = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad (7)$$

в которой делается перестановка столбцов, обратная (6)

$$\pi^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 5 & 6 & 1 & 3 & 4 \end{pmatrix},$$

что в итоге дает

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \quad (8)$$

Тестом правильности нахождения проверочной матрицы является выполнение любого из двух равенств

$$\begin{aligned} \mathbf{G}\mathbf{H}^T &= \mathbf{0}, \\ \mathbf{H}\mathbf{G}^T &= \mathbf{0}. \end{aligned} \quad (9)$$

Для корректно заданного линейного кода матрице \mathbf{G} соответствует единственная⁸ матрица \mathbf{H} , и наоборот. Из правила кодирования (1) и равенств (9) следует, что⁹

$$\begin{aligned} \vec{s}\mathbf{H}^T &= (\vec{a}\mathbf{G})\mathbf{H}^T = \vec{a}(\mathbf{G}\mathbf{H}^T) = \vec{0}, \\ \mathbf{H}\vec{s}^T &= \mathbf{H}(\vec{a}\mathbf{G})^T = \mathbf{H}(\mathbf{G}^T\vec{a}^T) = (\mathbf{H}\mathbf{G}^T)\vec{a}^T = \vec{0}^T, \quad \forall \vec{a}. \end{aligned} \quad (10)$$

Таким образом, любой кодовый вектор при умножении его на проверочную матрицу дает нулевой вектор $\vec{0}$, а любой вектор, не принадлежащий коду, $\vec{v} \notin C$, даст ненулевой вектор-произведение. Данный признак позволяет определять принадлежность произвольного вектора заданному линейному коду. Произведение

$$\vec{c} = \vec{v}\mathbf{H}^T, \quad \dim(\vec{c}) = 1 \times r, \quad (11)$$

называется **синдромом** — вектором, состоящим из результатов r проверок на четность. Размер проверочной матрицы, $\dim(\mathbf{H}) = r \times n$. Число строк r определяет количество проверок на четность.

⁸ С точностью до элементарных преобразований над строками

⁹ Известно общее правило $(\mathbf{u}\mathbf{v})^T = \mathbf{v}^T\mathbf{u}^T$

Рассмотрим важную особенность линейных кодов. Для этого в синдроме (11) заменим принятый вектор \vec{v} суммой переданного вектора и вектора ошибки

$$\vec{c} = (\vec{s} + \vec{e}) \mathbf{H}^T = \vec{s} \mathbf{H}^T + \vec{e} \mathbf{H}^T = \vec{e} \mathbf{H}^T . \quad (12)$$

Видим, что синдром зависит лишь от вектора ошибки, что позволяет по синдрому определять множество возможных ошибок и с некоторой вероятностью их исправлять. Обнаружение ошибки происходит тогда, когда синдром отличен от нуля. Если же синдром \vec{c} равен нулю, то либо ошибка отсутствует, либо она не может быть обнаружена данным кодом. Как мы выяснили ранее

$$\vec{s} \mathbf{H}^T = \vec{0} ,$$

но

$$\vec{c} = \vec{e} \mathbf{H}^T ,$$

поэтому если вектор ошибки \vec{e} , который сам по себе случаен, совпадет с любым из кодовых векторов \vec{s} , то соответствующий синдром \vec{c} будет равен нулю — такая ошибка при декодировании будет пропущена. Таким образом, множество кодовых векторов C некоторого кода определяет также множество ошибок, которые не могут быть обнаружены данным кодом, а спектр $W_{sp}(C)$ кода показывает кратности пропускаемых ошибок, w , и их количество, N_w . Заметим, что нулевой кратности соответствует нулевой вектор ошибки (ошибка отсутствует), и здесь синдром тривиально нулевой, т. е. если $\vec{e} = 0$, то $\vec{c} = 0$.

Итак, рассмотренный код, табл. 1, не может обнаружить семь разных векторов (шаблонов) ошибок. Все они перечислены в кодовой таблице.

Давайте взглянем на равенство

$$\mathbf{H} \vec{s}^T = \vec{0}^T ,$$

с другой стороны. Оно означает то, что сумма некоторых столбцов проверочной матрицы дает нулевой столбец, а именно тех столбцов, номера которых соответствуют номерам единиц в кодовом векторе \vec{s} . Например, возьмем из табл. 1 следующий кодовый вектор

$$\vec{s} = 110001 ,$$

и согласно ему просуммируем (по модулю два) первый, второй и последний столбцы проверочной матрицы

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Если некоторая сумма столбцов дала ноль, то значит суммируемые столбцы являются линейно зависимыми. Действительно, сумма первых двух дает третий столбец. Количество складываемых столбцов определяется количеством ненулевых элементов рассматриваемого кодового вектора, т.е. его весом, а минимальный ненулевой вес, как мы выяснили ранее, показывает кодовое расстояние линейного кода. Значит кодовое расстояние можно определить как наименьшее количество линейно зависимых столбцов проверочной матрицы. Для рассматриваемого кода кодовое расстояние равно трем, значит какие бы два столбца проверочной матрицы мы бы ни взяли, они должны быть линейно независимыми. Таким образом, последовательно перебирая все сочетания столбцов проверочной матрицы, начиная со столбцов по одиночке, и проверяя текущую комбинацию столбцов на линейную зависимость, можно отыскать первую попавшуюся комбинацию линейно зависимых столбцов, количество столбцов которой укажет на кодовое расстояние кода. Заметим, что столбец, состоящий из нулей, автоматически будет линейно зависимым, и кодовое расстояние такого кода будет равно единице. Наличие же одинаковых столбцов проверочной матрицы гарантированно даст нулевые суммы пар таких столбцов¹⁰, поэтому кодовое расстояние такого кода не может быть выше двух (если при этом все столбцы ненулевые, то кодовое расстояние будет равно двум).

Рассмотрим пример. Найдем кодовое расстояние двоичного кода, заданного проверочной матрицей

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \quad (13)$$

¹⁰ Для двоичных кодов верно равенство: $a+a=0$, $\forall a=(0,1)$

Размер проверочной матрицы 5×9 . Так как проверочная матрица была найдена по алгоритму (5)–(8), то она обязана содержать r единичных столбцов, в нашем случае — 5. Если это не так, то эквивалентными преобразованиями матрицы добиваемся этого. Номера таких столбцов: 1, 2, 4, 5 и 6. Назовем их базисными и сгруппируем отдельно, тогда оставшиеся небазисные столбцы фактически образуют матрицу $N = -Q^T$ и будут использоваться для определения кодового расстояния

$$N = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (14)$$

Подготовка сделана. Далее следуем относительно простому алгоритму:

1. Ищем в проверочной матрице (13) нулевые столбцы. Если они есть, то завершаем алгоритм с кодовым расстоянием $d_k = 1$. Иначе — идем дальше.
2. Ищем в проверочной матрице (13) одинаковые столбцы¹¹. Если они есть, то завершаем алгоритм с кодовым расстоянием $d_k = 2$. Иначе — идем дальше.
3. Ищем все возможные суммы¹² пар столбцов вспомогательной матрицы (14). В рассматриваемом случае их $C_4^2 = 6$. Выпишем их в виде строк

01010, 01100, 11001, 00110, 10011, 10101.

Среди найденных сумм ищем вектор с единичным хемминговым весом. Если есть хотя бы один такой, то завершаем алгоритм с кодовым расстоянием $d_k = 3$. Иначе — идем дальше.

4. Последовательно увеличивая $q > 2$, ищем все возможные суммы¹³ q столбцов вспомогательной матрицы (14). Таких сумм¹⁴ C_k^q . Среди найденных сумм ищем вектор с единичным хемминговым весом. Если есть

11 Для двоичного кода следует искать два коллинеарных вектора-столбца

12 Для двоичного кода следует искать все линейные комбинации пар столбцов

13 Для двоичного кода следует искать все линейные комбинации q столбцов

14 Для двоичного кода таких сумм $(m-1)^q C_k^q$, где m — основание кода

хотя бы один такой, то завершаем алгоритм с кодовым расстоянием $d_k = q+1$. Иначе — идем дальше.

5. Если $q=k$, то завершаем алгоритм с кодовым расстоянием $d_k = q+w$, где w — вес единственного вектора¹⁵, равного сумме всех k столбцов матрицы \mathbf{N} .

В рассматриваемом случае при $q=3$ получаются следующие суммы:

$$10000, 01001, 00101, 00011,$$

среди которых есть вектор с единичным весом, поэтому кодовое расстояние $d_k = q+1=4$. Пример закончен.

Если код задан корректно, то в проверочной матрице всегда найдутся r столбцов, которые образуют базис — набор линейно независимых столбцов. Значит некоторый столбец матрицы, который не входит в базис, может быть выражен в виде линейной комбинации базисных столбцов, в лучшем случае в виде суммы всех r столбцов. Именно этому лучшему случаю соответствует кодовый вектор \vec{s} с весом $r+1$, который и ограничит кодовое расстояние линейного кода сверху величиной

$$d_k \leq r+1. \quad (15)$$

Это неравенство известно как **граница Синглтона**¹⁶, которую также можно вывести из порождающей матрицы кода, записанной в систематической форме (а это всегда можно сделать)

$$\mathbf{G}_s = [\mathbf{I}^{(k)}, \mathbf{Q}] .$$

Здесь число столбцов матрицы \mathbf{Q} равно r , поэтому в лучшем случае в строке этой матрицы r ненулевых элементов, а в любой строке единичной матрицы $\mathbf{I}^{(k)}$ всегда один ненулевой элемент. Именно наличие единичной матрицы позволяет проанализировать ограничение сверху на вес самого "легкого" кодового вектора, а значит и на кодовое расстояние кода.

¹⁵ Для не двоичного кода следует вычислить все $(m-1)^k$ сумм и среди них выбрать единственный вектор с наименьшим весом. Здесь m — основание кода

¹⁶ На границе Синглтона лежат коды Рида-Соломона — не двоичные циклические коды

b) Задание, код 02

Случайным образом задается порождающая матрица \mathbf{G} . Требуется:

- По заданной порождающей матрице определить проверочную матрицу кода;
- По проверочной матрице определить кодовое расстояние кода.

2.3. Декодирование линейного кода**а) Сведения из теории****Задача декодирования**

Декодирование линейного кода основано на проверочной матрице кода, а точнее на равенствах (10)–(12). Задача декодирования делится на две части:

- по принятому из канала вектору \vec{v} оценивается переданный кодовый вектор \vec{s} .
- по оцененному кодовому вектору \vec{s} определяется информационный вектор \vec{a} .

Основная соль декодирования заключена в первой части. Оценка кодового вектора \vec{s} по принятому вектору \vec{v} , по существу, означает отыскание такого вектора ошибок \vec{e} , который корректирует вектор \vec{v} в кодовый вектор $\vec{s} = \vec{v} - \vec{e}$, дающий нулевой синдром $\vec{s} \mathbf{H}^T = \vec{0}$. Ошибка может быть исправлена лишь после ее обнаружения. Факт обнаружения ошибки равнозначен отличию от нуля синдрома $\vec{c} = \vec{v} \mathbf{H}^T$, и после коррекции ошибки синдром обязан стать равным нулю, при этом верно ли в итоге была исправлена ошибка — не известно (вероятность ошибки существует всегда). Несмотря на важность первой части, начнем рассмотрение со второй как с более простой.

Определение информационного вектора по кодовому

Определение информационного вектора \vec{a} по кодовому вектору \vec{s} делается на основании формулы кодирования

$$\vec{s} = \vec{a} \mathbf{G}.$$

Для этого в матрице \mathbf{G} следует найти k линейно независимых столбцов и выписать их номера. Далее из вектора \vec{s} следует выбрать элементы с найденными номерами и сформировать из них укороченный вектор \vec{s}_k . Аналогично следует укоротить матрицу \mathbf{G} до матрицы \mathbf{G}_k . Наконец, следует решить получившуюся систему из k линейных уравнений относительно элементов вектора \vec{a} . Рассмотрим пример. Зададим кодовый вектор $\vec{s} = 010110$ кода из табл. 1. Порождающая матрица кода равна

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Возьмем первые три столбца, $(1, 2, 3)$. Очевидно, что они будут линейно независимыми, потому что эквивалентными преобразованиями набор этих столбцов можно свести к набору, образующему единичную матрицу. Для этого следует последовательно выполнить три операции: 1) поменять местами первый столбец со вторым, 2) ко второму прибавить первый и 3) к третьему прибавить сумму первых двух

$$\mathbf{G}_k = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Сформируем из вектора \vec{s} укороченный вектор

$$\vec{s}_k = s_1 s_2 s_3 = 010.$$

Запишем укороченную систему линейных уравнений

$$(0 \ 1 \ 0) = (a_1 \ a_2 \ a_3) \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

откуда однозначно находим информационные символы

$$\begin{cases} a_1 + a_2 = 0 \\ a_1 = 1 \\ a_1 + a_2 + a_3 = 0 \end{cases} \Rightarrow \begin{cases} a_2 = 1 \\ a_1 = 1 \\ a_3 = 0 \end{cases},$$

из которых komponуется информационный вектор

$$\vec{a} = a_1 a_2 a_3 = 110 ,$$

совпадающий с соответствующим вектором, табл. 1, в предпоследней строке кодовой таблицы. Если взять, допустим, столбцы с номерами (1, 2, 5), то они будут линейно зависимыми, т. к. сумма первого и второго дает пятый. В этом случае укороченная система линейных уравнений будет выглядеть следующим образом

$$(0 \ 1 \ 1) = (a_1 \ a_2 \ a_3) \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} .$$

Попытка ее решения приведет к неопределенной ситуации относительно бита a_3

$$\begin{cases} a_1 + a_2 = 0 \\ a_1 = 1 \\ a_2 = 1 \end{cases} .$$

Конечно, никто не мешает бит a_3 найти отдельно: он в данном случае будет равен биту s_6 .

Коррекция ошибок в принятом векторе

Коррекция ошибок в принятом векторе \vec{v} заключается в такой модификации этого вектора, которая дает нулевой синдром

$$\mathbf{H}(\vec{v} - \vec{e})^T = \vec{0}^T , \quad \vec{e} = ? .$$

Векторов-решений \vec{e} поставленной задачи множество, потому что число строк проверочной матрицы меньше числа столбцов (\mathbf{H} не является квадратной). Это также можно обосновать тем, что каков бы ни был принятый вектор \vec{v} , содержащий обнаруживаемую ошибку, всегда можно найти 2^k разных векторов ошибок, переводящих этот вектор в кодовый — кодовых векторов как мы помним 2^k , — а значит дающих в итоге нулевой синдром. Естественно, задачу коррекции возможно решать и методом Монте-Карло, т. е. случайным

перебором векторов \vec{e} . Однако эта задача решается и более детерминировано: путем предварительного составления карты соответствий (map)

$$M = \{\vec{c}, \{\vec{e}\}\}.$$

Заметим, что одному синдрому соответствует несколько векторов ошибок, а именно 2^k . Это следует из рассуждений, сделанных парой абзацев выше. Нулевому синдрому соответствуют кодовые векторы кода и одновременно такие векторы ошибок, которые кодом пропускаются. Ненулевым синдромам соответствуют векторы, которые коду не принадлежат и одновременно такие векторы ошибок, которые кодом обнаруживаются. Вычислив по принятому вектору синдром

$$\vec{c} = \vec{v} \mathbf{H}^T,$$

в карте соответствий M выбирается один из векторов ошибок \vec{e} . По какому правилу выбирать, вопрос интересный... Выбор можно делать случайно, или на основании кратности ошибки, допустим, выбирая вектор ошибки с наименьшей кратностью. Способ выбора зависит от канала, в котором происходят ошибки. Естественным является выбор наиболее вероятного вектора ошибки. Рассмотрим пример коррекции ошибки в принятом векторе. Зададим код проверочной матрицей

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Так как все столбцы матрицы различные и ненулевые, то сумма по модулю два любых двух столбцов отлична от нуля, значит системы векторов, образованные одиночными столбцами и любыми парами столбцов будут линейно независимыми. Из этого следует, что кодовое расстояние кода больше двух. Но найдутся такие три столбца, например, под номерами 1, 3 и 5, сумма которых даст нулевой столбец — это три линейно зависимых столбца, поэтому кодовое расстояние кода равно трем. В итоге, данный код может гарантированно исправить все ошибки вплоть до первой кратности включительно. Вычислим синдромы для всех однократных ошибок

$$\vec{c} = \vec{e} \mathbf{H}^T.$$

Если код двоичный, то синдромы однократных ошибок совпадают со столбцами проверочной матрицы, только в транспонированном виде (табл. 2).

Таблица 2 Синдромы однократных ошибок для линейного кода (6, 3)

Вектор ошибки	Синдром
100000	100
010000	111
001000	010
000100	001
000010	110
000001	011

Видим, что все синдромы разные и ненулевые — это и говорит о том, что все однократные ошибки код гарантированно исправит. Так как код линейный, то синдромы для двукратных ошибок можно найти, вычисляя соответствующие линейные комбинации пар строк табл. 2. Допустим, складывая первые две строки, получаем вектор ошибки 110000 с синдромом $100+111=011$. Найденный синдром совпадает с синдромом для однократной ошибки 000001, значит дополнительно к однократным все двукратные ошибки рассматриваемый код исправить не сможет — синдромы попросту заняты векторами однократных ошибок. Единственный свободный синдром равен 101, который можно отдать, например, под исправление двукратной ошибки 1000100. Пусть принят вектор $\vec{v}=111010$ кода, заданного табл. 1. Вычисляем синдром, который в транспонированном виде будет равен сумме столбцов под номерами 1, 2, 3 и 5

$$\vec{c}^T = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Из табл. 2 следует, что вектор ошибки равен

$$\vec{e} = 010000.$$

Корректируем принятый вектор

$$\vec{s} = \vec{v} - \vec{e} = 111010 - 010000 = 101010.$$

Ошибка исправлена. В действительности, найденному синдрому соответствует $2^k = 2^3 = 8$ разных векторов ошибок¹⁷, но согласно заданной логике декодера — корректировать ошибки векторами минимальной кратности — мы выбрали единственный вектор, соответствующий однократной ошибке, при этом если в канале чаще всего происходят ошибки малой кратности, то с большой вероятностью декодер верно исправит имеющуюся ошибку, и лишь изредка может ошибиться, например, если произойдет двукратная ошибка.

Выпишем все векторы ошибок для найденного выше синдрома 111. Для этого решим относительно \vec{e} систему уравнений

$$\vec{e} \mathbf{H}^T = \vec{c} . \quad (16)$$

Эта система недоопределенная, потому что число уравнений меньше числа неизвестных. Ранее в (8) мы нашли проверочную матрицу с опорными столбцами под номерами 1, 3 и 4. Эти опорные столбцы позволят нам явно выразить три зависимых элемента искомого вектора ошибок¹⁸

$$\begin{cases} e_1 = c_1 + e_2 + e_5 \\ e_3 = c_2 + e_2 + e_5 + e_6 \\ e_4 = c_3 + e_2 + e_6 \end{cases} . \quad (17)$$

Структура данной системы уравнений определяется оставшимися столбцами проверочной матрицы под номерами 2, 5 и 6

$$-\mathbf{Q}^T = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} .$$

Элементы e_2 , e_5 и e_6 являются свободными, и т. к. они являются битами, то число разных вариантов равно $2^k = 2^3 = 8$, что и говорит о восьми разных векторах ошибок, приходящихся на один синдром. Из (17) следует, что для рассматриваемого кода векторы ошибок имеют следующую структуру

$$\vec{e} = f(\vec{c}) = (c_1 + e_2 + e_5, e_2, c_2 + e_2 + e_5 + e_6, c_3 + e_2 + e_6, e_5, e_6) .$$

¹⁷ Все они являются решением недоопределенной системы линейных уравнений

¹⁸ В общем случае их количество равно r

Это общее решение системы (16). Перебор свободных элементов и подстановка синдрома $\vec{c}=111$ дают список векторов ошибок, табл. 3. Найденные векторы ошибок указывают на такие столбцы проверочной матрицы, сумма которых дает заданный синдром. Из табл. 3 следует, что синдрому $\vec{c}=111$ соответствует единственная ошибка первой кратности. Карта соответствий $M=[\vec{c},\{\vec{e}\}]$ называется **классами смежности**. Каждый класс характеризуется своим синдромом. Классу нулевого синдрома соответствуют пропускаемые ошибки или кодовые векторы линейного кода, в зависимости от того как посмотреть.

Таблица 3 Возможные векторы ошибок для синдрома 111 кода (6, 3)

$e_2e_5e_6$	$e_1e_3e_4$	e	$q=w(e)$
000	111	101100	3
001	100	100001	2
010	001	000110	2
011	010	001011	3
100	000	010000	1
101	011	011101	4
110	110	111010	4
111	101	110111	5

Таким образом, каждому синдрому линейного (n,k) кода соответствует 2^k векторов ошибок \vec{e} . Эти векторы являются решением недоопределенной системы линейных уравнений $\vec{e}\mathbf{H}^T=\vec{c}$. Решение существует в том числе и для нулевого синдрома — этим решением будет множество кодовых векторов кода.

б) Задание, код 03

Случайным образом задается порождающая матрица \mathbf{G} и принятый кодовый вектор. Требуется:

- По порождающей матрице определить проверочную матрицу кода;
- По проверочной матрице определить кодовое расстояние кода;
- Декодировать принятый кодовый вектор¹⁹.

¹⁹ При исправлении **всегда** ориентироваться на ошибки минимальной кратности; при неоднозначности выбора выбор делать случайным образом, не нарушая упомянутого принципа минимума

2.4. Неравенство Хемминга

а) Сведения из теории

Неравенство Хемминга, записанное для некоторого линейного двоичного кода, заданного параметрами n и $r=n-k$, позволяет определить **верхнюю границу** кратности исправления $q_{\text{и}}$, т. е. потенциальные корректирующие способности (n, k) кода. Неравенство основано на том, что синдромы — это инструменты для исправления ошибок, а количество различных ненулевых синдромов равно

$$N_{\text{синдр.}} = 2^r - 1,$$

а различных векторов ошибок кратности не выше $q_{\text{и}}$

$$N_{\text{ош.}} = \sum_{q=1}^{q_{\text{и}}} C_n^q,$$

где C_n^q — количество сочетаний из n по q . Естественно, что для гарантированного исправления всех ошибок вплоть до кратности $q_{\text{и}}$ включительно, количество синдромов должно превышать количество ошибок, т. е. должно выполняться неравенство

$$N_{\text{ош.}} \leq N_{\text{синдр.}} \Rightarrow \sum_{q=1}^{q_{\text{и}}} C_n^q \leq 2^r - 1.$$

Для удобства упростим последнее неравенство

$$\sum_{q=0}^{q_{\text{и}}} C_n^q \leq 2^r. \quad (18)$$

Здесь $q_{\text{и}}$ является искомой наибольшей величиной, при которой неравенство еще выполняется. Кратность исправления можно искать последовательным перебором, начиная с нуля. Известно²⁰, что кодовое расстояние кода не может быть выше $r+1$, и в то же время кратность исправления определяется как половина от кодового расстояния, поэтому справедливо неравенство

²⁰ Смотри предыдущие задания

$$q_{\text{и}} \leq \frac{r}{2} ,$$

что говорит о возможности последовательного перебора сверху вниз. Для больших случайных двоичных кодов чаще всего должно быть

$$q_{\text{и}} \approx \frac{r}{4} , \quad (19)$$

т. е. нечто среднее между крайними значениями 0 и $r/2$. Так что, можно начинать перебор со значения (19), после чего в зависимости от знака неравенства перебирать либо вверх, либо вниз. Скорее всего, такая стратегия быстрее приведет к решению неравенства. Рассмотрим пример. Зададим $n=63$, $r=20$. Здесь $2^{20}=1048576$. Перебор $q_{\text{и}}$ снизу вверх последовательно дает

$$\begin{array}{ll} 1 & < 2^{20} \\ 64 & < 2^{20} \\ 2017 & < 2^{20} \\ 41728 & < 2^{20} \\ 637393 & < 2^{20} \\ 7666240 & > 2^{20} \end{array} .$$

Значит $q_{\text{и}}=4$, и кодовое расстояние двоичного кода (63, 43) не может быть выше девяти, т. е. $d_{\text{к}} \leq 2q_{\text{и}} + 1 = 2 \cdot 4 + 1 = 9$. Видим, что приближенная оценка по (19)

$$q_{\text{и}} \approx \frac{r}{4} = 20/4 \approx 5$$

дает довольно-таки точное значение кратности исправления. Задавая далее вероятность битовой ошибки как p , и рассматривая **канал с независимыми ошибками**, можно определить вероятность появления ошибочного кодового вектора на выходе декодера линейного кода. Кодовая ошибка на выходе декодера будет тогда, когда в канале произойдет ошибка, кратность которой выше кратности, определенной из неравенства Хемминга (18). Вероятность ошибки кратности q в рассматриваемом канале определяется формулой Бернулли

$$P(q) = C_n^q p^q (1-p)^{n-q}, \quad 0 \leq q \leq n, \quad (20)$$

Значит вероятность кодовой ошибки на выходе декодера²¹ равна

$$P_{\text{ош. дек.}} = \sum_{q=q_{\text{и}}+1}^n P(q). \quad (21)$$

При $q_{\text{и}} \ll np$ вероятность (21) точнее вычислять от обратного

$$P_{\text{ош. дек.}} = 1 - \sum_{q=0}^{q_{\text{и}}} P(q). \quad (22)$$

Произведение np равно средней кратности ошибки в канале.

Продолжим рассматривать код (63,43) и зададим несколько вероятностей битовой ошибки в канале p .

- $p=0.002$. Вычисления по (22) дают следующий результат

$$P_{\text{ош. дек.}} = \sum_{q=5}^{63} P(q) \approx 2.04 \cdot 10^{-7}.$$

Значит декодер, работающий в режиме исправления, с вероятностью примерно $2 \cdot 10^{-7}$ может выдать ошибочный кодовый вектор, при этом сколько в нем будет ошибочных бит — неизвестно. Вероятность битовой ошибки можно примерно оценить по следующей формуле²²

$$P_{\text{бит. дек.}} \approx P_{\text{ош. дек.}} \frac{d_{\text{к}}}{n}. \quad (23)$$

Эта формула основана на выбранной ранее (стр. 21) стратегии исправления ошибок векторами малой кратности, в результате чего при превышении кратностью ошибки величины $q_{\text{и}}$ скорректированный неверно вектор будет кодовым вектором, отличающимся от истинного кодового вектора, вероятнее всего, минимальным числом элементов, т. е. как раз на величину кодового расстояния. Вычисления по (23) дают

$$P_{\text{бит. дек.}} \approx 2.04 \cdot 10^{-7} \frac{9}{63} \approx 2.91 \cdot 10^{-8}.$$

²¹ Здесь декодер настроен на исправление ошибок минимальной кратности

²² Это граница сверху

- $p=0.03$. Последовательные вычисления по (22) и (23) дают

$$P_{\text{ош. дек.}} \approx 0.0407 \quad , \quad P_{\text{бит. дек.}} \approx 5.81 \cdot 10^{-3} \quad .$$

- $p=0.079$. Вычисления по (22) и (23) дают

$$P_{\text{ош. дек.}} \approx 0.56 \quad , \quad P_{\text{бит. дек.}} \approx 0.08 \quad .$$

Таким образом, при приближении произведения np к кратности исправления $q_{\text{и}}$, вероятность битовой ошибки на выходе декодера практически совпадает с вероятностью битовой ошибки в канале, и толк от кодирования исчезает — рассматриваемый код с таким количеством ошибок не справляется.

b) Задание, код 04

Случайным образом задаются параметры линейного кода n и k , а также вероятность битовой ошибки в канале p . Требуется определить:

- Кратность исправления, $q_{\text{и}}$;
- Вероятность появления ошибочного кодового вектора на выходе декодера, $P_{\text{ош. дек.}}$;
- Вероятность битовой ошибки на выходе декодера, $P_{\text{бит. дек.}}$.

3. Вероятность ошибки при оптимальном приеме цифрового сигнала

a) Сведения из теории

Рассматривается процесс передачи и приема двоичных кодовых векторов фиксированной размерности n с помощью набора из M попарно различных импульсов, $s_i(t)$, $i=1, \dots, M$. Каждому импульсу соответствует определенный символ²³. Это как каждой воображаемой букве — символу — человек может поставить в соответствие ее начертание — сигнал. Так как символ модуляции может принимать M разных значений, то в одном символе содержится $n = \log_2 M$ битов²⁴. Множество из M значений образует **алфавит**.

²³ Говорят "символа модуляции", однако, мы чаще будем говорить коротко — "символа"

²⁴ Число M берут так, чтобы двоичный логарифм был натуральным числом

Символы следуют с интервалом времени T , при этом говорят, что "частота следования символов равна $1/T$ ". Символьная частота определяет **скорость телеграфирования**, которая измеряется в бодах. Один бод — один символ в секунду. Скорость телеграфирования называют также "бóдовой скоростью передачи информации".

В качестве сигнальных импульсов рассмотрим радиоимпульсы²⁵

$$s_i(t) = \sqrt{\frac{2E_i}{T}} \cos(2\pi f_i t + \varphi_i), \quad i=1, \dots, M, \quad (24)$$

заданные на интервале времени $0 < t < T$. Интеграл от мгновенной мощности²⁶ $p_i(t) = s_i^2(t)$ по определению дает энергию импульса

$$E_i \stackrel{\text{def}}{=} \int_0^T p_i(t) dt = \int_0^T s_i^2(t) dt. \quad (25)$$

Величина E_i в (24) тем точнее соответствует энергии (25), чем лучше выполняется неравенство $f_i T \gg 1$, говорящее о том, что количество периодов колебания на отрезке времени T много больше единицы. Однако, уже при $f_i T > 12$ относительная ошибка не превышает 1%, и на практике это неравенство выполнено с достаточной точностью.

Каждый символ из алфавита имеет свою вероятность p_i , $1 \leq i \leq M$, поэтому определим средние характеристики радиоимпульса (24)

- Средняя энергия на один символ, $E_s = \sum_i E_i p_i$.
- Средняя частота, $f_0 = \sum_i f_i p_i$.

Средняя фаза обычно равна нулю и не несет особенного смысла. Определив среднюю частоту f_0 , радиоимпульс (24) можно записать в виде

$$s_i(t) = \sqrt{\frac{2E_i}{T}} \cos(2\pi f_0 t + \theta_i), \quad (26)$$

²⁵ Они используются для **радиосвязи**

²⁶ На один ом сопротивления

где $\theta_i = 2\pi \Delta f_i t + \varphi_i$ — фаза, несущая в себе информацию, $\Delta f_i = f_i - f_0$ — величина отклонения частоты относительно среднего значения. Условие

$$\Delta f_i \ll f_0$$

необходимо для обеспечения узкополосности формируемого сигнала. В этом случае такой сигнал полезно записать через квадратурные компоненты²⁷

$$s_i(t) = I_i(t) \cos(2\pi f_0 t) - Q_i(t) \sin(2\pi f_0 t), \quad (27)$$

где

$$I_i(t) = \sqrt{\frac{2E_i}{T}} \cos \theta_i, \quad Q_i(t) = \sqrt{\frac{2E_i}{T}} \sin \theta_i.$$

Квадратурные компоненты наглядно отображать на плоскости в декартовых координатах (I, Q) , где время t является параметром. Частота f_0 информации не содержит, и называется **несущей частотой**. Формула (27) определяет способ формирования радиосигнала с произвольным видом модуляции, который задается зависимостями $I_i(t)$ и $Q_i(t)$. Устройство, осуществляющее способ (27), называется **универсальным квадратурным модулятором**. В этом и состоит полезность квадратурного представления узкополосных сигналов — радиосигналов: несущая формируется отдельно, квадратуры — отдельно²⁸.

Энергию на один символ E_s полезно пересчитать в энергию на один бит информации

$$E_b = \frac{E_s}{\log_2 M},$$

однако, если в системе передачи информации применяется помехоустойчивое кодирование, то следует использовать более общую формулу

$$E_b = \frac{E_s}{R \log_2 M}. \quad (28)$$

Здесь $R = k/n$ — скорость кодирования²⁹. Применение корректирующих кодов приводит к уменьшению энергии на один символ. Это связано с тем, что

²⁷ Используется формула $\cos(A+B) = \cos A \cos B - \sin A \sin B$

²⁸ Говорят при этом, что задача факторизовалась на две подзадачи

²⁹ Смотри стр. 5

скорость передачи информационных битов фиксирована, и k битов должны быть переданы с помощью n кодовых битов, что ведет к их укорочению в R раз. При отсутствии кодирования $R=1$. С другой стороны, если не укорачивать биты, влияние избыточного кода на энергию на один бит можно понимать через уменьшение количества информационных битов в R раз и, значит, через увеличение затрат энергии в пересчете на один бит, который в среднем передается с помощью $1/R \geq 1$ кодовых битов.

Если информация содержится в амплитуде³⁰ $\sqrt{E_i}$, то такой вид модуляции называют **амплитудной манипуляцией**, Амн, при этом дискретные уровни, как правило, задаются симметрично относительно нуля. Этим минимизируется средняя мощность формируемого сигнала.

Если информация содержится в фазе φ_i , то такой вид модуляции называют **фазовой манипуляцией**, ФМн. Дискретные уровни обычно задают с равномерно по всей окружности, т. к. это обеспечивает минимум вероятности ошибки при приеме равновероятных символов.

Если информация содержится в частоте f_i , то такой вид модуляции называют **частотной манипуляцией**, ЧМн, при этом дискретные уровни, как правило, задают с равным шагом по оси частот. Шаг по частоте выбирается из условия взаимной ортогональности пар импульсов

$$(s_i(t), s_j(t)) = \int_0^T s_i(t) s_j(t) dt = 0, \quad i \neq j.$$

Этим минимизируется вероятность ошибки при приеме ЧМн-сигналов, но при этом в спектре формируемого сигнала могут возникнуть особенности в виде игл — практического проявления дельта-функций.

Если информация одновременно содержится и в амплитуде $\sqrt{E_i}$, и в фазе, и при этом сигнальные точки (с координатами I и Q) расположены в узлах прямоугольной сетки, то такой вид модуляции называют **квадратурной амплитудной модуляцией**, КАМ. Среднее значение комплексной амплитуды $\sqrt{E_i} \exp(j\varphi_i)$ при этом равно нулю, т. к. все символы равновероятные и расположены симметрично относительно начала координат.

³⁰ Амплитуда пропорциональна корню из энергии (или мощности)

Существует также широко распространенная **амплитудно-фазовая манипуляция**, АФМ, при которой символы расположены на нескольких заранее выделенных концентрических окружностях — орбитах. Фактически, такой вид модуляции можно рассматривать как множество фазовых модуляций ФМн с разными энергиями. Достоинством АФМ по сравнению с КАМ-модуляцией является меньший пик-фактор³¹ сигнала и меньший требуемый динамический диапазон³² передатчика, что достигается ценой некоторого снижения минимального расстояния между символами³³. Модуляция АФМ позволяет использовать усилители мощности с большим КПД (либо более дешевые), что актуально, например, для спутниковых ретрансляторов телевизионного вещания и трансляции сети Интернет.

Будем считать, что сигнал в процессе передачи и приема не искажается, а единственным мешающим фактором является тепловой шум приемника $n(t)$. Экспериментально показано, что амплитуда n теплового шума распределена по гауссовскому закону с нулевым средним и некоторой дисперсией σ^2 , которая пропорциональна средней мощности шума.

Всякий шум, который измерен³⁴, ограничен некоторой полосой частот. Тепловой шум приемника устроен так, что его средняя мощность распределена равномерно Δf по всей рабочей полосе частот, какой бы большой эта полоса ни была³⁵. По этой причине полагают, что на входе приемника действует шум, мощность которого распределена равномерно по всей бесконечно большой полосе частот — такой шум называют **белым**.

Важный момент! Условимся, что радиосигнал в приемнике с помощью гетеродина и смесителя перенесен на нулевую частоту, т. е. преобразован в видеосигнал. Говорят при этом, что вся дальнейшая обработка идет на **baseband** (в основной полосе частот).

31 Отношение максимальной мощности радиосигнала к средней

32 Отношение максимальной мощности к минимальной. Для передатчиков минимальной мощностью является мощность точки сигнального созвездия, наиболее близкой к началу координат (0, 0)

33 Например, сравнивая КАМ-16 и АФМ-16 при одинаковой средней мощности, пик-фактор удастся снизить на 0,8 дБ с наименьшим снижением минимального расстояния между символами (на 1,5 дБ), при этом внешняя окружность с восемью символами получается на 5 дБ мощнее внутренней. У КАМ-16 максимальная и минимальная мощности отличаются на 9,5 дБ.

34 Допустим, анализатором спектра

35 Ограничения наступают для частот в терагерцовой области и выше; смотри закон Планка

Рассматривая обработку на **baseband**, можно утверждать, что после прохождения белым шумом идеального фильтра нижних частот с единичным коэффициентом передачи и частотой среза Δf , шум $n(t)$ становится белым с ограниченной полосой, мощность которого конечна и определяется дисперсией

$$\sigma^2 = N_0 \Delta f$$

дискретного шума, полученного из аналогового шума $n(t)$ выборкой отсчетов с шагом³⁶ $\Delta t = 1/(2 \Delta f)$. Здесь N_0 — односторонняя спектральная плотность мощности, Вт/Гц или В²/Гц. Она показывает сколько в среднем ватт приходится на полосу один герц. Чем больше плотность, тем выше уровень шума и выше вероятность ошибки в приемнике.

Принимаемый сигнал имеет вид (напомним, что это обработка на **baseband**)

$$v(t) = s_i(t) + n(t) . \quad (29)$$

Задача приемника заключается в наблюдении импульса $v(t)$ на интервале времени $0 < t < T$ и в дальнейшем угадывании номера $i = 1, \dots, M$ переданного символа. Естественно, что множество эталонных импульсов $\{s_i(t)\}$ в приемнике известно заранее.

Чтобы решить задачу поиска структуры оптимального приемника, следует определиться с критерием оптимальности. Для систем передачи информации критерием является **минимум вероятности ошибки**. Тепловой шум приемника $n(t)$ является одновременно и белым, и гауссовским, что значительно упрощает поиск структуры оптимального приемника. Доказано, что оптимальный приемник-различитель импульсов на фоне **аддитивного белого гауссовского шума**, АБГШ, должен работать по принципу наименьшего евклидова расстояния

$$d^2(v, s_i) = \int_0^T [v(t) - s_i(t)]^2 dt . \quad (30)$$

Оптимальный приемник вычисляет все расстояния для $i = 1, \dots, M$ и выбирает тот индекс i , которому соответствует наименьшее.

³⁶ Данный шаг определен как координата первого нуля автокорреляционной функции белого шума, ограниченного по полосе частот. В этом случае получаемые отсчеты шума будут некоррелированными, и, если шум — гауссовский, то и независимыми

Важный момент! В (29) перед сигналом $s_i(t)$ стоит единица, а значит мощность поступающего на вход оптимального приемника сигнала $v(t)$ заранее выровнена относительно средней энергии эталонных импульсов³⁷.

На основании анализа (30) можно показать, что эквивалентной, но более удобной, структурой является скалярное произведение

$$(v(t), s_i(t)) = \int_0^T v(t) s_i(t) dt . \quad (31)$$

Интервал от 0 до T называют **интервалом наблюдения**. В результате такого наблюдения принятый импульс $v(t)$ сворачивается в число, равное скалярному произведению (31). Данное число является мерой, которая косвенно показывает сколько информации содержится в импульсе $v(t)$ относительно импульса $s_i(t)$. Чем больше это число, тем больше информации, следовательно тем больше уверенности в том, что принятый импульс $v(t)$ соответствует импульсу $s_i(t)$. Значит, вычисляя скалярные произведения для всех возможных $i=1, \dots, M$ и сравнивая их между собой, путем выбора наибольшего можно вынести вердикт относительно переданного импульса и, значит, ему соответствующего символа модуляции, состоящего из $n = \log_2 M$ битов. Если сигнальных импульсов всего два, т. е. $M=2$, то процедура (31) упрощается до сравнения двух величин v_1 и v_2 между собой.

Приемник, работающий по правилу (31), называют **корреляционным приемником**, а блок, вычисляющий корреляционный интеграл — **коррелятором**. Заметим, что здесь под словом "прием" понимают обработку принятого сигнала с целью извлечения информации, поэтому каким "аналоговым" процедурам подвергался сигнал $v(t)$ до коррелятора — усилению, преобразованию частоты, фильтрации, автоматической регулировке по мощности — нас не интересует.

Корреляционный интеграл, если подставить в него (29), полезно расписать на сигнальную и шумовую составляющие

$$v = \int_0^T s_i(t) [s_j(t) + n(t)] dt = v_c + v_{ш} , \quad (32)$$

³⁷ Тема автоматического выравнивания мощности здесь не рассматривается

причем шумовая составляющая будет распределена по гауссовскому закону, как и шум $n(t)$, ее образующий, а сигнальная составляющая будет неслучайной и равной скалярному произведению двух сигнальных импульсов

$$v_c = \int_0^T s_i(t) s_j(t) dt = \sqrt{E_i E_j} r_{ij}, \quad (33)$$

которое можно выразить через среднее геометрическое их энергий, умноженное на коэффициент корреляции r_{ij} двух импульсов³⁸.

При $i=j$ коэффициент корреляции равен единице и $v_c = E_i = E_j = E$. Так как шумовая компонента распределена по гауссовскому закону с нулевым средним, то определим оставшийся параметр: дисперсию (среднюю мощность) шумовой компоненты³⁹

$$\begin{aligned} \sigma_{\text{ш}}^2 \stackrel{\text{def}}{=} \overline{v_{\text{ш}}^2} &= \overline{\left(\int_0^T n(t) s_i(t) dt \right)^2} = \int_0^T \int_0^T \overline{n(t_1) n(t_2)} s_i(t_1) s_i(t_2) dt_1 dt_2 = \\ &= \frac{N_0}{2} \int_0^T s_i^2(t) dt = \frac{E N_0}{2} \end{aligned} \quad (34)$$

Видим, что мощность шумовой компоненты пропорциональна спектральной плотности мощности шума, а также энергии опорного импульса, участвующего в обработке⁴⁰. Величину (34) называют "мощностью шума после обработки" или "мощностью шума на выходе коррелятора". Мощность неслучайной сигнальной компоненты после обработки

$$v_c^2 = E^2,$$

поэтому разумно определить отношение сигнал-шум (по мощности) после обработки

$$q^2 = \frac{2E}{N_0}. \quad (35)$$

38 Для модуляции ФМн-2 коэффициент взаимной корреляции достигает минимума и равен -1 . Это своего рода исключение из правил, поэтому этот вид модуляции самый помехоустойчивый в канале с АБГШ

39 Вывод формулы основан на фильтрующем свойстве дельта-функции, чем является функция автокорреляции белого шума

40 Заметим, что величина $\sigma^2 = N_0 \Delta f$ обозначает мощность шума до обработки

Мощность сигнальной компоненты растет пропорционально квадрату энергии, в то время как мощность шумовой линейно зависит от энергии. Линейный рост вызван белостью шума, когда в среднем выходит так, что реализации $n(t)$ ортогональны опорному сигналу $s_i(t)$, и поэтому шумовой интеграл

$$\left(\int_0^T n(t) s_i(t) dt \right)^2 ,$$

в среднем, с ростом энергии E_i растет медленнее, чем сигнальный интеграл

$$\left(\int_0^T s_i(t) s_i(t) dt \right)^2 ,$$

под знаком которого стоит произведение одинаковых сигналов.

Введенное отношение сигнал-шум (35) соответствует отношению сигнал-шум в задаче обнаружения (например, радиолокационного) импульса $s_i(t)$. В системах передачи информации решается задача различения M импульсов, поэтому для удобства вводят отношение сигнал-шум по так называемому **разностному сигналу [1, 2]**. Важность и существование разностного сигнала очевидны, если рассмотреть интеграл (30) при $M=2$ и выразить его через энергию на один бит E_b

$$\int_0^T [s_1(t) - s_2(t)]^2 dt = 2 E_b (1 - r_{12}) . \quad (36)$$

Здесь r_{12} — коэффициент взаимной корреляции двух импульсов

$$r_{12} \stackrel{\text{def}}{=} \frac{1}{\sqrt{E_1 E_2}} \int_0^T s_1(t) s_2(t) dt .$$

Шум мы положили равным нулю, тем самым найдя сигнальную компоненту⁴¹. Можно показать [1], что мощность шумовой компоненты при этом будет равна величине

41 На самом деле разность двух сигнальных компонент. Первой соответствует прием импульса $s_1(t)$, второй — $s_2(t)$, при этом опорный сигнал коррелятора — разностный сигнал $w(t) = s_1(t) - s_2(t)$

$$E_b N_0 (1 - r_{12}) , \quad (37)$$

откуда вытекает и становится определением отношение сигнал-шум по разностному сигналу

$$q_c^2 \stackrel{\text{def}}{=} \frac{4 E_b (1 - r_{12})}{N_0} . \quad (38)$$

Корреляционная обработка принимаемых импульсов подразумевает, что параметры эталонных сигналов $s_i(t)$ в приемнике в той или иной мере известны. Амплитуда и частота известны с некоторой погрешностью, которая определяется каналом распространения сигнала⁴² и классом приемопередающей аппаратуры, а начальная фаза φ_i в (26) — вот тот параметр, который может оставаться полностью неизвестным, т. е. фактически случайным, и полноценный прием при правильном выборе алгоритма обработки будет возможен. По тому, насколько известна начальная фаза принимаемых импульсов, определяют **степень когерентности приемника**.

Если в процессе приема начальная фаза φ_i опорного сигнала никак не оценивается, то такой приемник является **некогерентным**. Если же начальная фаза оценивается, то — **частично-когерентным**⁴³. Приемники, в которых фаза известна априори, являются **когерентными**⁴⁴.

Оценка начальной фазы делается контуром фазовой автоподстройки частоты, ФАПЧ, что является отдельной важной темой и здесь не рассматривается. Для некогерентных систем количество корреляторов удваивается, потому что вводится квадратурный канал для компенсации случайности фазы, исходя из основного тригонометрического тождества⁴⁵

$$\cos^2 \varphi + \sin^2 \varphi = 1 .$$

Некогерентные приемники возможны при амплитудной манипуляции, частотной и даже при фазовой, при этом в последнем случае должно быть введено дифференциальное (разностное) кодирование битов. Для модуляций типа КАМ-М приемник всегда будет когерентным (в реальности — частично-когерентным).

42 Разного рода замирания в канале, ослабление сигнала и эффект Допплера

43 На практике их называют когерентными

44 Их в природе не существует, это эталоны. Существует только математическая модель

45 Не левым, так правым вытянем информацию, по аналогии с полным приводом у автомобиля

Вероятность символьной ошибки $P_{\text{ош. дем.}}$ на выходе демодулятора зависит от вида модуляции, от вида обработки — когерентная или некогерентная, — а также от соотношения сигнал-шум (35). Приведем формулы для модуляций с числом символов $M = \{2, 4, 8\}$. Используем дополнительную функцию ошибок

$$\operatorname{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} \exp(-t^2) dt, \quad (39)$$

а также функцию Маркума⁴⁶

$$Q_1(a, b) = \int_b^{\infty} t \exp\left(-\frac{t^2 + a^2}{2}\right) I_0(at) dt. \quad (40)$$

Подробный вывод формул для вероятности ошибки при $M=2$ смотри в [2], кроме некогерентной ФМн с дифференциальным кодированием битов.

- Число символов $M=2$. Здесь $E_s = E_b$.

- Амплитудная, АМн.

- Когерентная

$$P_{\text{ош. дем.}} = \frac{1}{2} \operatorname{erfc}\left(\frac{q}{2\sqrt{2}}\right), \quad q^2 = \frac{4E_s}{N_0}.$$

- Некогерентная⁴⁷

$$P_{\text{ош. дем.}} = \frac{1}{2} \exp\left(-\frac{v_n^2}{2}\right) + \frac{1}{2} [1 - Q_1(q, v_n)], \quad q^2 = \frac{4E_s}{N_0}, \quad I_0(q v_n) = \exp(q^2/2).$$

- Частотная, ЧМн.

- Когерентная

$$P_{\text{ош. дем.}} = \frac{1}{2} \operatorname{erfc}\left(\frac{q}{2\sqrt{2}}\right), \quad q^2 = \frac{4E_s}{N_0}.$$

- Некогерентная

$$P_{\text{ош. дем.}} = \frac{1}{2} \exp\left(-\frac{E_s}{2N_0}\right).$$

⁴⁶ Смотри Приложение А

⁴⁷ Подробности смотри в Приложении А

- Фазовая, ФМн.

- Когерентная

$$P_{\text{ош. дем.}} = \frac{1}{2} \operatorname{erfc} \left(\frac{q}{2\sqrt{2}} \right) , \quad q^2 = \frac{8 E_s}{N_0} .$$

- Частично-когерентная — ФАПЧ с дифференциальным кодированием битов [3]

$$P_{\text{ош. дем.}} = 2 p (1 - p) , \quad p = \frac{1}{2} \operatorname{erfc} \left(\frac{q}{2\sqrt{2}} \right) , \quad q^2 = \frac{8 E_s}{N_0} .$$

- Число символов $M = 4$. Здесь $E_s = 2 E_b$.

- КАМ-4, ФМн-4

- Когерентная

$$P_{\text{ош. дем.}} = p (2 - p) , \quad p = \frac{1}{2} \operatorname{erfc} \left(\frac{q}{2\sqrt{2}} \right) , \quad q^2 = \frac{8 E_s}{N_0} .$$

- Частично-когерентная — ФАПЧ с дифференциальным кодированием дибитов⁴⁸

$$P_{\text{ош. дем.}} = 2 p_c (1 - p_c) , \quad p_c = p (2 - p) , \quad p = \frac{1}{2} \operatorname{erfc} \left(\frac{q}{2\sqrt{2}} \right) , \quad q^2 = \frac{8 E_s}{N_0} .$$

- Число символов $M = 8$. Здесь $E_s = 3 E_b$.

- ФМн-8

- Когерентная⁴⁹

$$P_{\text{ош. дем.}} = 1 - \frac{1}{2\pi} \int_{-\frac{\pi}{8}}^{\frac{\pi}{8}} e^{-\frac{q^2}{2} \sin^2 \varphi} \int_0^\infty r e^{-\frac{1}{2}(r - q \cos \varphi)^2} dr d\varphi \approx \operatorname{erfc} \left(\frac{q}{\sqrt{2}} \sin \frac{\pi}{8} \right) , \quad q^2 = \frac{2 E_s}{N_0} .$$

Приближение сделано для $q \geq 2$, при этом ошибка не превышает 1%.

⁴⁸ Дибит — символ из двух битов, четырехзначный символ

⁴⁹ Вывод смотри в Приложении Б

Частота битовых ошибок в случае $M=2$ совпадает с вероятностью символьной ошибки $P_{\text{ош. дем.}}$. При $M=4$ она зависит от способа отображения дибитов на символ; оптимальным в плане минимизации вероятности ошибки⁵⁰ является следующее отображение: максимально удалены друг от друга дибиты 00 и 11, а также 01 и 10, соответственно. В этом случае для когерентных систем $P_{\text{бит. дем.}} = p$, а для частично-когерентных

$$P_{\text{бит. дем.}} = 2p(1-p). \text{ При } M=8 \text{ для ФМн-8 с кодом Грея } P_{\text{бит. дем.}} \approx \frac{1}{3} P_{\text{ош. дем.}}.$$

Все виды модуляции сравниваются по вероятности битовой ошибки при определенном значении сигнал-шум на один бит

$$\frac{E_b}{N_0}.$$

Или, наоборот, задается вероятность битовой ошибки, например, 10^{-5} , и вычисляется требуемое отношение сигнал-шум на один бит для нескольких видов модуляции или нескольких прототипов модемов, включающих некоторый помехоустойчивый код. При наличии кода производительность модема характеризуется вероятностью ошибки на выходе декодера $P_{\text{бит. дек.}}$, методика оценки которой⁵¹ дана в п. 2.4.

б) Задание, код 05

Случайным образом задаются: вид модуляции m , требуемая вероятность битовой ошибки на выходе жесткого декодера $P_{\text{бит. дек.}}$ (hard-decision decoder), работающего в режиме исправления ошибок (Forward Error Correction, FEC), а также длина кода n и кратность исправления $q_{\text{и}}$. Требуется:

- Подобрать скорость кодирования R так, чтобы обеспечить минимум

$$\text{требуемого отношения сигнал-шум } \frac{E_b}{N_0}.$$

⁵⁰ Смотри код Грея

⁵¹ Для жесткого декодера, hard-decision decoder

4. Регенерация цифрового сигнала при передаче на большие расстояния

а) Сведения из теории

Данная тема актуальна при рассмотрении проблемы передачи цифровой информации⁵² на большие расстояния — десятки, сотни и тысячи километров, как правило, по медному или оптическому кабелю, или же по воздуху (например, радиоинтерфейсу радиорелейных линий).

Регенерация означает восстановление чего-либо. В данном случае восстанавливаются двоичные символы — биты⁵³, и восстанавливаются они из принимаемого сигнала, кодирующего эти биты согласно заданному закону модуляции. По существу, регенерация и есть прием сигнала в смысле извлечения исходной информации. Однако, различают регенерацию без извлечения информации. Это, фактически, усиление с возможной коррекцией формы сигнальных импульсов, но без их демодуляции и детектирования. И различают регенерацию с извлечением информации (полноценный прием и последующая передача). Соответственно, регенерационные пункты — пункты, в которых производится регенерация импульсов — делят на **необслуживаемые** (НРП) и **обслуживаемые** (ОРП). Естественно, обслуживаемые регенераторы сложнее и требуют больше ресурсов, начиная с потребляемой мощности и заканчивая требуемым временем на обработку — вносимой задержкой. Однако, за счет введения разумного, как правило небольшого, количества обслуживаемых регенераторов возможно значительно снизить итоговую вероятность битовой ошибки, или снизить требуемое отношение сигнал-шум на входе приемника, что равнозначно увеличению максимально допустимой длины участка или снижению требуемой мощности формируемых передатчиком импульсов.

⁵² То есть информации, квантованной как по времени, так и по уровню

⁵³ В подавляющем большинстве случаев исходная информация представлена в виде последовательности битов

Будем рассматривать такие НРП, регенераторы в которых делают лишь простое усиление импульсов без коррекции их формы⁵⁴. Трассу передачи информации разобьем на $N > 0$ участков, причем через каждые $M \leq N$ участков будет идти один ОРП. Если $M = N$, то в системе передачи информации всего лишь один ОРП, который является окончательным приемником Rx . Например, при $N = 6$ и $M = 2$ регенераторная структура системы передачи информации имеет вид как на рис. 1. Здесь $N - 1 = 6 - 1 = 5$ регенерационных пунктов, из них $m = N/M - 1 = 2$ являются ОРП (выделены красным цветом), а $n = N - N/M = 3$ пункта — НРП (показаны синим цветом). Количество участков между соседними ОРП равно M . Величина $m + 1$ показывает количество ОРП вместе с окончательным приемником Rx .

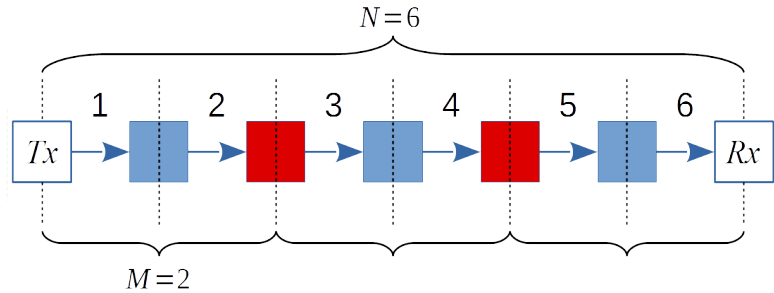


Рис 1. Структурная схема системы передачи информации с пятью регенераторами, два из которых — обслуживаемые.

Определим отношение сигнал-шум q^2 . Для этого выделим участок от исходного передатчика Tx до первого приемника, рис. 2, которым может быть как НРП, так и ОРП. В радио- или оптическом приемнике решающую роль в плане накопления шумов играет малошумящий усилитель (МШУ), поэтому разумно часть, идущую до МШУ, обозначить отдельным блоком "Преобразователь" (Пр). Таким преобразователем может быть антенна приемника плюс полосовой фильтр, если это прием радиоволн, а также фотоприемник, если это прием оптического сигнала. С приемом электрического сигнала интереснее. Здесь вместо МШУ используется широкополосный видеоусилитель (например, технология Ethernet по витой паре), а в качестве преобразователя используется гальваническая развязка типа трансформатора. При этом шум на выходе преобразователя образуется как результат возможной междуканальной и промышленной помехи — такие помехи за счет скремблирующих кодов в каждом канале в полосе частот приемника можно считать белым шумом.

54 Но, возможно, с преобразованием частоты

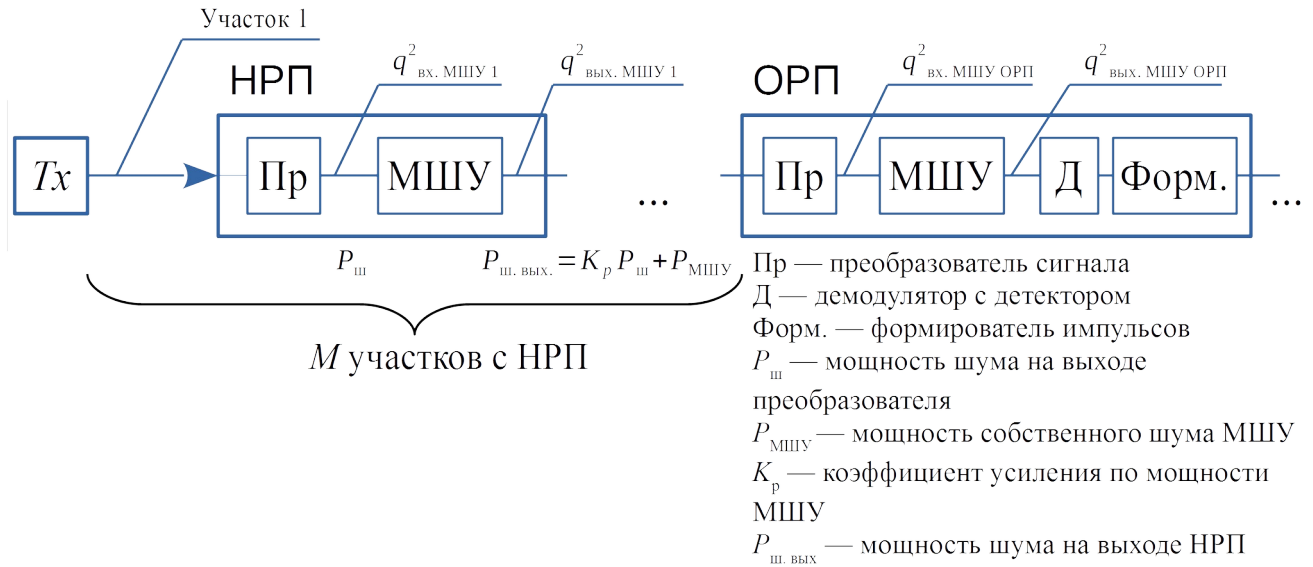


Рис 2. Разделение приемной части на преобразователь и МШУ. Отличие НРП от ОРП

- Вход первого малошумящего усилителя (МШУ) приемника

На входе первого МШУ присутствует аддитивная смесь $v(t)$ полезного сигнала $s(t)$ с шумом $n(t)$, $v(t) = s(t) + n(t)$, поэтому отношение сигнал-шум на входе — это отношение средней мощности сигнала $P_{ср}$ к мощности шума $P_{ш}$

$$q^2_{вх. МШУ 1} = \frac{P_{ср}}{P_{ш}}.$$

Мощность шума определяется через шумовую полосу⁵⁵ приемника $B_{ш}$ и спектральную плотность мощности шума $n(t)$ N_0 как $P_{ш} = N_0 B_{ш}$. Здесь спектральная плотность N_0 определяется возможным шумом антенны, если это радиорелейная линия, или шумом фотоприемника, если это оптоволоконная линия, или суммарными помехами, если это прием электрического сигнала.

- Выход первого МШУ приемника

Проходя через МШУ, как сигнальная часть, так и шумовая усиливаются по мощности в K_p раз, однако за счет усилительных элементов⁵⁶ добавляется внутренний шум мощностью $P_{МШУ}$, что определяет сигнал-шум на выходе МШУ

⁵⁵ Энергетическую полосу. Эту полосу следует отличать от полосы по критерию половинной мощности

⁵⁶ И других нелинейных элементов, в основном, смесителей при преобразовании частоты

$$q_{\text{вых. МШУ 1}}^2 = \frac{P_{\text{ср}} K_p}{P_{\text{ш}} K_p + P_{\text{МШУ}}} \stackrel{\text{def}}{=} \frac{q_{\text{вх. МШУ 1}}^2}{F_{\text{ш}}} .$$

Здесь $F_{\text{ш}} = 1 + \frac{P_{\text{МШУ}}}{K_p P_{\text{ш}}}$ — шум-фактор⁵⁷ МШУ, причем $F_{\text{ш}} > 1$. Для

удобства шум-фактор выражают в децибелах $F_{\text{ш [дБ]}} = 10 \lg F_{\text{ш}}$. По определению шум-фактор показывает во сколько раз отношение сигнал-шум на выходе некоторого звена меньше отношения сигнал-шум на его входе. Другими словами шум-фактор показывает степень деградации отношения сигнал-шум при прохождении сигналом этого звена.

- Учет последовательно включенных одинаковых НРП

После прохождения M участков с **одинаковыми** НРП с **одинаковым** ослаблением по мощности L , которое полностью компенсируется усилением $K_p = 1/L$, отношение сигнал-шум на входе МШУ ОРП уменьшается в M раз

$$q_{\text{вх. МШУ ОРП}}^2 = \frac{q_{\text{вх. МШУ 1}}^2}{M} .$$

На выходе МШУ ОРП формируется целевое отношение сигнал-шум

$$q_{\text{вых. МШУ ОРП}}^2 = \frac{q_{\text{вх. МШУ 1}}^2}{M F_{\text{ш}}} ,$$

которое будет определять вероятность ошибки при детектировании в текущем ОРП. Видим, что при прохождении нескольких НРП происходит накопление шумов. Однако, при прохождении нескольких ОРП происходит накопление не шумов, а битовых ошибок, которые в итоге могут быть обнаружены и исправлены декодерами корректирующих кодов.

Рассмотрим доказательство уменьшения сигнал-шум в M раз после прохождения M участков с НРП при $M=2$. Отношение сигнал-шум на выходе первого МШУ

$$q_1^2 = \frac{P_{\text{ср}} K_p}{P_{\text{ш}} K_p + P_{\text{МШУ}}} .$$

⁵⁷ Коэффициент шума

Отношение сигнал-шум на выходе второго МШУ

$$q_2^2 = \frac{P_{\text{ср}} K_p L K_p}{[L(P_{\text{ш}} K_p + P_{\text{МШУ}}) + P_{\text{ш}}] K_p + P_{\text{МШУ}}} = \frac{P_{\text{ср}} K_p}{K_p P_{\text{ш}} + P_{\text{МШУ}} + K_p P_{\text{ш}} + P_{\text{МШУ}}} = \frac{q_1^2}{2}.$$

Здесь последовательно учитывалось ослабление на трассе L , добавление шума преобразователя $P_{\text{ш}}$, усиление $K_p = 1/L$ и добавление шума $P_{\text{МШУ}}$. Естественно, что подразумевается одинаковость всех НРП.

Накопление битовых ошибок в ОРП можно учесть по точной формуле или по приближенной. Приближенная выводится из предположения, что ошибки на выходе системы передачи информации не будет, если не было ошибок ни в одном ОРП, включая окончательный приемный, поэтому вероятность ошибки p

$$p_{\text{ош. итоговая}} \approx 1 - (1 - p_{\text{ош. ОРП}})^{m+1}. \quad (41)$$

Здесь вероятность битовой ошибки на выходе ОРП $p_{\text{ош. ОРП}}$ вычисляется исходя из отношения сигнал-шум $q_{\text{вых. МШУ ОРП}}^2$ и вида модуляции. Заметим, что на выходе каждого ОРП отношение сигнал-шум одинаковое, т. к. между любыми соседними ОРП находятся одинаковые участки с НРП и формирователи ОРП совпадают с формирователем передатчика T_x . Точная формула для вероятности ошибки следует из условия, что ошибка на выходе системы передачи информации будет лишь при нечетных сочетаниях битовых ошибок в $m+1$ ОРП

$$p_{\text{ош. итоговая}} = \sum_{\substack{j - \text{нечетное} \\ 0 < j \leq m+1}} C_{m+1}^j p_{\text{ош. ОРП}}^j (1 - p_{\text{ош. ОРП}})^{m+1-j}. \quad (42)$$

Здесь C_n^k — биномиальный коэффициент, т. е. число сочетаний из n по k .

б) Задание, код 06

Случайным образом задаются: общее количество участков N , количество участков с НРП M , отношение сигнал-шум на входе первого МШУ $q_{\text{вх. МШУ 1}}^2$, в дБ, а также коэффициент шума $F_{\text{ш}}$, в дБ. Требуется определить:

- Вероятность битовой ошибки на выходе системы передачи информации $p_{\text{ош. итоговая}}$ по приближенной формуле (41) и по точной (42),

- Требуемое отношение сигнал-шум (в дБ) на входе первого МШУ $q_{\text{вх. МШУ 1}}^2$ для обеспечения битовой вероятности ошибки $p_{\text{ош. итоговая}}$ при условии, что:

- Все блоки — НРП, т. е. $M=N$,
- Все блоки — ОРП, т. е. $M=1$, использовать приближенную формулу (41).

Считать, что используется двоичная частотная манипуляция с ортогональными импульсами и некогерентным приемником, т.е.

$$q_{\text{вых. МШУ ОРП}}^2 = \frac{E_s}{N_0} \quad \text{и} \quad p_{\text{ош. ОРП}} = \frac{1}{2} \exp\left(\frac{-q_{\text{вых. МШУ ОРП}}^2}{2}\right) .$$

5. Кодирование источника

5.1. Коды Хаффмана и Шеннона-Фано

а) Сведения из теории

Под кодированием источника подразумевается экономное кодирование в смысле минимизации **среднего** количества бит, затрачиваемого на передачу некоторого символа X , определяемого источником информации. Символ X характеризуется количеством значений m , которые он может принимать или, что равнозначно, объемом m соответствующего алфавита X . Бит по определению является символом, который может принимать лишь $m=2$ значения, поэтому бит — двоичный символ. Источником X может быть некоторый датчик с аналого-цифровым преобразователем, клавиатура компьютера, генерирующая кодовые комбинации при каждом нажатии на клавишу, часть книги, являющаяся последовательностью печатных символов из заранее известного алфавита. Помимо основания кода m , символ X характеризуется набором вероятностей $p(x_i)$, $1 \leq i \leq m$, с которыми появляются значения x_i . Именно в распределении вероятностей и заключается вся соль, определяющая потенциальные возможности экономного кодирования некоторого источника. В "простонародии" экономное кодирование называют сжатием (компрессией)

информации. Также в узких кругах такое кодирование называют эффективным или энтропийным.

Понятие *энтропии* тесно связано с понятием *неопределенности*. Сама неопределенность естественным образом связана с некоторой вероятностью p , лежащей на интервале $(0,1)$. Если вероятность чего-либо равна единице или нулю, то все определено и неопределенность отсутствует; в противном случае — присутствует, и то, насколько она присутствует, определяет количественная величина энтропии

$$H(X) = \sum_{i=1}^m p(x_i) \log \frac{1}{p(x_i)}.$$

Своеобразной аксиомой здесь является выражение

$$I(x_i) = \log \frac{1}{p(x_i)} \geq 0,$$

определяющее количество собственной информации, которое содержится в конкретном значении x_i . Ключом для понимания является то, что само значение не играет роли, играет роль лишь вероятность этого значения — чем менее вероятно значение, тем больше информации получает наблюдатель (приемник) при появлении этого значения на выходе источника (входе приемника). Единичной вероятности соответствует нулевое количество информации; в этом случае на выходе источника появляется одно и то же значение, которое заранее известно приемнику. Сами значения не несут информации в том смысле, что играет роль лишь номер i , который определяет адрес в таблице соответствия $i \rightarrow x_i$, по которому приемник восстанавливает конкретное значение. Таблица соответствия известна как в передатчике, так и в приемнике, иначе прием будет просто невозможен. Так вот, при появлении на входе приемника значения x_i наблюдатель получает количество информации $I(x_i)$, т. е. у него снимается неопределенность с $I(x_i)$ до нуля⁵⁸ на величину $\Delta H(x_i) = I(x_i) - 0 = I(x_i)$. При передаче разных значений величина снимаемой неопределенности будет принимать в общем случае разные значения $\Delta H(x_i)$ в зависимости от

⁵⁸ Неопределенность снимается до нуля, потому что рассматриваются каналы без помех. С помехами борются другие коды — помехоустойчивые

выпавшего значения x_i , поэтому естественно желание найти некоторую среднюю величину снимаемой неопределенности, например, ее математическое ожидание, чем и является энтропия $H(X)$. Ее также называют *информационной производительностью* источника. Она показывает сколько в среднем генерируется информации на один символ X или, что равнозначно, среднюю скорость генерации информации. Единицей измерения энтропии чаще является бит/символ, в этом случае основание логарифма берется равным двум

$$H(X) = \sum_{i=1}^m p(x_i) \log_2 \frac{1}{p(x_i)}, \text{ бит/символ.}$$

Эту функцию также называют *энтропийной функцией Шеннона* в честь Клода Шеннона — основателя теории информации.

Доказано, что величина энтропии неотрицательна и ограничена сверху

$$0 \leq H(X) \leq \log m.$$

По этой причине источник X с основанием кода m не может в среднем генерировать больше, чем $H_{\max} = \log m$ информации в единицу времени (на один символ). Причем максимальная производительность $H(X) = H_{\max}$ достигается при равновероятных значениях символа $p(x_i) = 1/m$. В этом случае алфавит используется максимально эффективно. Это образно можно сравнить с конвейером из бочек — физических контейнеров — до краев наполненных вином — информацией. Например, емкость двоичного символа 1 бит, емкость троичного символа $\log_2 3 \approx 1,58$ бит, емкость четверичного — 2 бита, емкость русского алфавита $\log_2 33 \approx 5,04$ бит, емкость байт-символа (256 значений) — 8 бит. Для осознания того, что емкость троичного символа 1,58 бит, требуется составить несколько экономных двоичных кодов (физических контейнеров), вычислить **среднюю арифметическую** длину кодового слова, сравнить ее с величиной 1,58 и убедиться, что никакой код не даст среднюю арифметическую длину кодового слова меньше, чем $\log_2 3 \approx 1,58$ бит. Приведем примеры. Обратите внимание, что символы — равновероятные.

Значение символа X	Вероятность $P(X)$	Двоичный код №1	Длина слова l , бит	Двоичный код №2	Длина слова l , бит
x_1	1/3	00	2	0	1
x_2	1/3	01	2	11	2
x_3	1/3	11	2	10	2
$\sum P(X)=1$			$L=2$		$L=5/3 \approx 1,67$

При произвольных вероятностях теоретическим минимумом на **среднюю длину** (математическое ожидание) L некоторого **двоичного** кода будет величина энтропии — информационной емкости источника

$$L \geq L_{\min} = H(X) .$$

Пример — в таблице ниже.

Значение символа X	Вероятность $P(X)$	Двоичный код №1	Длина слова l , бит	Двоичный код №2	Длина слова l , бит
x_1	4/5	00	2	0	1
x_2	1/10	01	2	11	2
x_3	1/10	11	2	10	2
$\sum P(X)=1$			$L=2$		$L=6/5=1,2$

Здесь энтропия равна

$$H(X) = \frac{4}{5} \log_2 \frac{5}{4} + \frac{1}{10} \log_2 10 + \frac{1}{10} \log_2 10 \approx 0,922 \text{ бит/символ.}$$

Для оценки качества кодирования вводят коэффициент избыточности R

$$R_{\text{до кодир.}} = 1 - \frac{H(X)}{H_{\max}} \quad \text{— до кодирования,}$$

$$R_{\text{после кодир.}} = 1 - \frac{L_{\min}}{L} \quad \text{— после кодирования.}$$

Код №2 оказывается экономнее кода №1 в плане средней длины кодового слова. При кодировании соблюдается принцип — более вероятному значению x_i ставится в соответствие по возможности более короткое кодовое слово, при этом никакое кодовое слово не должно быть началом других слов. Последнее свойство называется **свойством префикса**. Оно позволяет однозначно декодировать

сплошной поток битов без разделительных символов; в противном случае в алфавит следовало бы включить и их, однако в новом образовавшемся алфавите и коде свойство префикса все равно обязано выполняться, как ни крути.

Коды Хаффмана

Существует алгоритм кодирования, позволяющий получить оптимальный код в плане наименьшей средней длины $L_{\text{опт. код}}$, так, что выполняется неравенство

$$L_{\min} = \frac{H(X)}{\log r} \leq L_{\text{опт. код}} < L_{\text{другой код}},$$

где r — основание кода. Такой алгоритм был предложен в 1952 г Дэвидом Хаффманом, а соответствующий код был назван *кодом Хаффмана*. В процессе кодирования строится так называемое дерево, листьями которого являются значения символа x_i . Свойство префикса выполняется автоматически. Рассмотрим пример кодирования пятиричного символа двоичным кодом Хаффмана. Здесь $m=5$, $r=2$. Пусть все значения равновероятные, т. е. $p(x_i)=1/5$. Алгоритм кодирования следующий.

1. Имеющиеся символы сортируются в порядке убывания вероятностей
2. Два символа с наименьшими вероятностями объединяются в один, вероятности складываются, при этом верхнему символу присваивается логическая единица, нижнему — ноль.

Процедура 1-2 повторяется до тех пор, пока не останется один символ, которому ничего не присваивается — это корень дерева. Для каждого исходного символа — листа — определяется путь до корня. Соответствующие биты, записанные в обратном порядке, и будут искомыми кодовыми словами. Получившееся дерево показано на рис. 3.

$\log_2 5 \approx 2,32$ является дробным числом, то найденный код Хаффмана имеет кодовые слова, длина которых аппроксимирует величину $\log_2 5 \approx 2,32$ в натуральных числах, т. е. колеблется от 2 к 3 с перевесом в сторону 2.

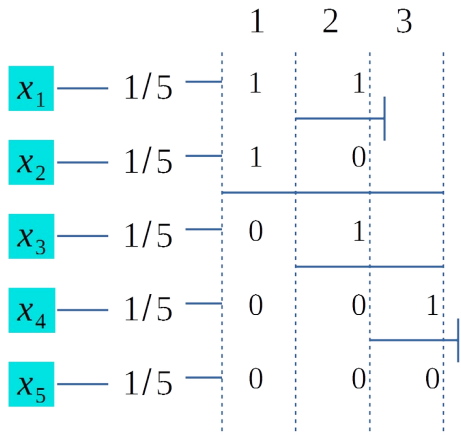


Рис. 4 Кодирование кодом Шеннона-Фано пятеричного символа с равновероятными значениями

Коды Шеннона-Фано

В смысле средней длины кодового слова Коды Шеннона-Фано дают либо такой же результат, как и коды Хаффмана, либо несколько хуже, т. е. они являются почти оптимальными⁶¹. Свойство префикса в любом случае выполняется. Алгоритм кодирования следующий.

1. Символы сортируются в порядке убывания вероятностей
2. Имеющееся множество символов делится на два подмножества так, чтобы сумма вероятностей в каждом из них была бы **максимально** одинаковой⁶²
3. Каждому символу верхней подгруппы присваивается логическая единица, а символу нижней — ноль.

Процедура 2-3 применяется рекурсивно к каждому из подмножеств до тех пор, пока в подмножестве не останется по одному элементу, которому ничего не присваивается. В итоге, напротив каждого символа будет сформировано кодовое слово. Рассмотрим пример кодирования кодом Шеннона-Фано пятеричного символа с равновероятными значениями, рис. 4. Здесь возможна неоднозначность. В частности на первом шаге в верхнее подмножество можно включить либо два символа, либо — три. Выберем первый вариант. На втором шаге верхнее подмножество разделилось однозначно и деление закончилось, а для нижнего был выбран вариант с одним символом в формирующемся верхнем подмножестве. На третьем шаге оставшееся нижнее подмножество из двух нулей однозначно разделилось на два. Деление закончилось. Видим, что в данном случае код

⁶¹ Субоптимальными

⁶² Или, что равнозначно, разность суммарных вероятностей для двух подмножеств была бы минимальной

Шеннона-Фано дал ту же самую среднюю длину $L=2,4$ бита, что и код Хаффмана. Убеждаемся, что свойство префикса выполняется.

Кодирование объединенных символов

В предыдущем параграфе было рассмотрено так называемое посимвольное кодирование, когда одному значению x_i , появляющемуся на выходе некоторого источника X с вероятностью $p(x_i)$ ставится в соответствие кодовое слово $c^{(i)}$, состоящее из l_i кодовых символов. Таким образом, формируется код со средней длиной слова $L = \sum_i p(x_i)l_i$. Естественно, что никто не мешает последовательные значения x_i на выходе источника X группировать по два $x_i x_j$ и обозначать двойной символ как некоторый вектор

$$\vec{Y} = (X^{(1)} X^{(2)}) , \quad n=2 .$$

Здесь n — размер вектора. Объем алфавита \vec{Y} равен $N = m^n = m^2$, где m — объем алфавита X . Вероятности значений y_k определяются вероятностями значений x_i и переходными вероятностями $p(x_i/x_j)$

$$p(y_k) \stackrel{\text{def}}{=} p(x_i x_j) = p(x_i) p(x_j/x_i) = p(x_j) p(x_i/x_j) .$$

Если значения x_i и x_j независимы, то искомая вероятность равна произведению безусловных вероятностей. Однако, при независимости объединяемых значений смысл от объединения не особенно велик, но все же есть: доказано, что после объединения итоговые вероятности $p(y_k)$ становятся ближе к отрицательным степеням двойки и поэтому соответствующий поток значений y_k может быть закодирован несколько более эффективно. Но если объединяемые значения зависимы, то эффективность кодирования может быть повышена в разы. Зависимость x_i и x_j приводит к неравновероятности значений y_k , поэтому в итоге потенциально возможная эффективность кодирования определяется только лишь вероятностями и соответствующей энтропией

$$H(\vec{Y}) = \sum_{k=1}^N p(y_k) \log_2 \frac{1}{p(y_k)} , \quad N = m^n .$$

Проиллюстрируем влияние зависимости на перекося вероятностей группового символа. Пусть имеется двоичный символ X с равновероятными значениями x_1 и x_2 . Считаем, что вероятности текущего значения x_i зависят от предыдущего значения x_j следующим образом

$$p(x_2/x_1) = p(x_1/x_2) = 0,2 \text{ ,}$$

$$p(x_2/x_2) = p(x_1/x_1) = 0,8 \text{ .}$$

Это значит, что за текущим значением с вероятностью 0,8 следует то же самое значение, а с вероятностью 0,2 значение меняется на другое. Таким образом, вероятности пар

$$p(y_1) = p(x_1 x_1) = p(x_1) p(x_1/x_1) = 0,5 \cdot 0,8 = 0,4 \text{ ,}$$

$$p(y_1) = p(x_1 x_1) = p(x_1) p(x_2/x_1) = 0,5 \cdot 0,2 = 0,1 \text{ ,}$$

$$p(y_2) = p(x_1 x_2) = p(x_1) p(x_2/x_1) = 0,5 \cdot 0,2 = 0,1 \text{ ,}$$

$$p(y_1) = p(x_1 x_1) = p(x_1) p(x_1/x_1) = 0,5 \cdot 0,8 = 0,4 \text{ .}$$

Убеждаемся, что сумма вероятностей равна единице и пары далеко не равномерно распределены по вероятностям. Энтропия исходного источника X равна 1 бит/символ — ее еще называют однобуквенной энтропией, — а энтропия источника \vec{Y} равна

$$H(\vec{Y}) = H_2(X) = 2 \cdot 0,4 \log_2 \frac{1}{0,4} + 2 \cdot 0,1 \log_2 \frac{1}{0,1} \approx 1,72 \text{ бит/символ.}$$

Данную энтропию называют двухбуквенной, т. к. она определяется по парам значений источника X . Доказано, что n -буквенная энтропия неотрицательна и ограничена сверху

$$0 \leq H_n(X) \leq n \log m = n H_{1\max} \text{ ,}$$

причем максимума энтропия достигает лишь тогда, когда символы равновероятные и независимые, т. е. при полностью случайном потоке значений. В пересчете на символ X источник Y генерирует

$$\frac{H(\vec{Y})}{n} = \frac{H(\vec{Y})}{2} \approx \frac{1,72}{2} = 0,86 \text{ бит/символ.}$$

Разница между 1 бит/символ и 0,86 бит/символ вызвана зависимостью значений x_1 и x_2 . В данном случае кодирование пар потенциально дает возможность уменьшения средней длины в 0,86 раз относительно однобуквенного кодирования.

Таким образом, к снижению энтропии и возможному увеличению эффективности кодирования приводят

- неравновероятность значений символа X
- статистическая зависимость этих значений,

однако последняя влечет неравновероятность слов $(x_i x_j \dots x_s)$, что и определяет главенствующую роль распределения вероятностей в оценке возможностей экономного кодирования или, что эквивалентно, информационной энтропии.

б) Задание, код 07

Случайным образом задаются вероятности значений троичного символа X , $m=3$. Требуется:

- Построить двоичный код Хаффмана
- Построить двоичный код Шеннона-Фано
- Определить избыточность R до и после кодирования
- Объединить символы по два, $n=2$, и выполнить пункты задания заново.

6. Литература

1. Акулиничев, Ю. П. Теория и техника передачи информации: Учебное пособие [Электронный ресурс] / Ю. П. Акулиничев, А. С. Бернгардт. — Томск: ТУСУР, 2012. — 210 с. — Режим доступа: <https://edu.tusur.ru/publications/1750>.

2. Новиков, А. В. Демодуляция цифровых сигналов. Статистический и сигнальный подходы: Учебное пособие [Электронный ресурс] / Новиков А. В. — Томск: ТУСУР, 2018. — 51 с. — Режим доступа: <https://edu.tusur.ru/publications/7150>.

3. Новиков, А. В. Вероятность битовой ошибки при дифференциальном декодировании: Учебно-методическое пособие для проведения практических занятий и самостоятельной работы [Электронный ресурс] / А. В. Новиков. — Томск: ТУСУР, 2019. — 21 с. — Режим доступа: <https://edu.tusur.ru/publications/9007>

4. Федорюк, М. В. Метод перевала / М. В. Федорюк // Москва: Наука, 1977, 368 с.

7. Приложение А. К расчету вероятности ошибки при некогерентном приеме АМн-сигнала

Дополнительная функция ошибок `erfc()` может быть без особых проблем вычислена в программах, например, *Mathcad*, *Matlab*, *Octave*, *SciPy/Python* путем вызова одноименной функции

```
import scipy.special as sp
```

```
print(sp.erfc(1))
```

Результат вычисления `erfc(1)`

```
0.15729920705028516
```

Функция Маркума может быть вычислена в программах *Matlab* или *Octave*⁶³

$$Q_1(a, b) = \text{marcumq}(a, b) ,$$

а также с помощью библиотеки *SciPy* языка *Python*

$$1 - Q_1(a, b) = \text{stats.ncx2.cdf}(b^2, 2, a^2) .$$

Также можно использовать любые программы, в которые встроена функция распределения `cdf()`⁶⁴ закона распределения Райса⁶⁵.

Данные для проверки правильности вызова функции:

$$Q(1, 1) = 0.7328798037968218 ,$$

$$Q(1, 0) = 1 ,$$

⁶³ Требуется Signal Processing Toolbox в Matlab или пакет signal в Octave

⁶⁴ CDF – Cumulative Distribution Function

⁶⁵ Нецентрального хи-квадрат распределения с двумя степенями свободы

$$Q(1,2)=0.2690120600359135 \text{ ,}$$

$$Q(2,1)=0.9181076963694064 \text{ .}$$

Пороговый уровень v_{π} находится путем численного решения уравнения⁶⁶

$$I_0(q v_{\pi}) = \exp\left(\frac{q^2}{2}\right) \text{ .}$$

Уравнение может быть решено в программах *Mathcad* или *SciPy* языка *Python* путем минимизации модуля разности

$$F(v) = \left| I_0(q v) - \exp\left(\frac{q^2}{2}\right) \right|$$

функцией `minimize(F, v0)` . Также уравнение можно решить графически. Пример решения уравнения на языке *Python*

```
import scipy.special as sp
import scipy.optimize as opt
import math

q = 1.
x0 = 1.
fun = lambda x: math.fabs( sp.iv(0, q * x) - math.exp(q ** 2 / 2) )
print(opt.minimize(fun, x0).x)
```

Результат решения при $q=1$

[1.5020333]

Однако, для больших отношений сигнал-шум описанная процедура приведет к численным ошибкам и неверному результату. Это связано с экспоненциальным ростом функции Бесселя для большого q . Чтобы преодолеть данную проблему, используем масштабированную функцию Бесселя

$$I_{0.sc}(x) \stackrel{\text{def}}{=} I_0(x) \exp(-x) \text{ , } x \geq 0 \text{ .}$$

Тогда минимизации подлежит следующий модуль разности

$$F(v) = \left| I_{0.sc}(q v) - \exp\left(\frac{q^2}{2} - q v\right) \right| \text{ .}$$

66 $I_0(x)$ – функция Бесселя первого рода нулевого порядка

Однако, этого мало, и следует прологарифмировать обе части исходного равенства

$$I_{0.sc}(qv) = \exp\left(\frac{q^2}{2} - qv\right),$$

и сформировать модуль разности заново

$$F(v) = \left| \ln[I_{0.sc}(qv)] - q\left(\frac{q}{2} - v\right) \right|.$$

В качестве начального приближения рекомендуется брать $v_0 = q/2$.

Пример:

```
import numpy as np
import scipy.special as sp
import scipy.optimize as opt

q = 10.
v0 = q / 2.
fun = lambda x: np.fabs( np.log(sp.ive(0., q * x)) - q * (q / 2. - x) )
v = np.fabs(opt.minimize(fun, v0).x[0])
```

Результат:

```
v = 5.290076168076615
```

Здесь $\text{sp.ive}(0, x)$ означает масштабированную функцию Бесселя $I_{0.sc}(x)$.

8. Приложение Б. Вероятность символьной ошибки при когерентном приеме ФМн-8 в канале с АБГШ

Будем рассматривать созвездие ФМн-8 с равновероятными символами. Каждый из восьми символов состоит из трех битов, таким образом скорость рассматриваемого вида модуляции составляет 3 бит/символ. Пусть созвездие ориентировано так, что один из символов, допустим S_0 , лежит на вещественной оси I , тогда ему будет соответствовать сигнал $s_0(t) = A \cos(2\pi f t)$, где A — амплитуда сигнала, которая равна длине сигнального вектора. В канале с АБГШ к полезному сигналу добавляется белый гауссовский шум

$$v(t) = s_0(t) + n(t),$$

с двусторонней спектральной плотностью мощности $N_0/2$, Вт/Гц. Пусть в когерентном приемнике имеются две опорные несущие

$$i(t) = A \cos(2\pi f t) , \quad q(t) = A \sin(2\pi f t) ,$$

тогда при идеальной тактовой синхронизации на выходах корреляторов в каналах I и Q сформируются следующие уровни

$$v_i = \int_0^T v(t) i(t) dt = \int_0^T A \cos(2\pi f t) A \cos(2\pi f t) dt + \int_0^T n(t) A \cos(2\pi f t) dt = v_{ic} + v_{in} ,$$

$$v_q = \int_0^T v(t) q(t) dt = \int_0^T A \cos(2\pi f t) A \sin(2\pi f t) dt + \int_0^T n(t) A \sin(2\pi f t) dt \approx v_{qn} .$$

Уровень на выходе коррелятора состоит из суммы сигнальной части v_c и шумовой $v_{ш}$. При передаче символа S_0 сигнальной частью в канале Q можно пренебречь, т.к. синус и косинус одной частоты на интервале T , значительно превышающем период колебания $1/f$, практически ортогональны. Сигнальная часть в канале I равна энергии принятого импульса

$$v_{ic} = \frac{A^2 T}{2} = E_s .$$

Здесь считается, что опорные сигналы в приемнике выровнены по амплитуде с принимаемыми импульсами. Так как шум в канале гауссовский а коррелятор — линейное устройство, то уровни v_i и v_q также будут иметь гауссовское распределение с параметрами

$$m_i = \overline{v_i} = v_{ic} = E_s ,$$

$$\begin{aligned} \sigma_i^2 &= \overline{(v_i - \overline{v_i})^2} = \int_0^T \int_0^T \overline{n(t_1) n(t_2)} A \cos(2\pi f t_1) A \cos(2\pi f t_2) dt_1 dt_2 = \\ &= \frac{A^2 N_0}{2} \int_0^T \cos^2(2\pi f t) dt = \frac{A^2 T N_0}{4} = \frac{E_s N_0}{2} \end{aligned}$$

$$m_q = \overline{v_q} = 0 , \quad \sigma_q^2 = \sigma_i^2 = \sigma^2 = \frac{A^2 T N_0}{4} = \frac{E_s N_0}{2} .$$

Для рассматриваемой конфигурации модема в канале I сформировано так называемое отношение сигнал-шум после обработки

$$q^2 \stackrel{\text{def}}{=} \frac{m_i^2}{\sigma_i^2} = \frac{A^2 T}{N_0} = \frac{2 E_s}{N_0} .$$

Вероятность символьной ошибки в данном случае удобно вычислить от обратного, найдя сначала вероятность правильного приема. Вспомним, что мы передаем символ S_0 , поэтому прием будет правильным, если квадратуры v_i и v_q попадут в угол, вершина которого совпадает с началом координат, а две стороны образуют с осью I угол $\pm \pi/8$ радиан. Величины v_i и v_q будут некоррелированными, потому что они образованы ортогональными проекциями шума $n(t)$; т. к. они вдобавок гауссовские, то они будут и независимыми. В результате плотность вероятностей пары (v_i, v_q) равна произведению плотностей отдельных величин

$$w(v_i, v_q) = w(v_i) w(v_q) = \frac{1}{2\pi} \frac{1}{\sigma_i \sigma_q} e^{-\frac{1}{2\sigma_i^2}(v_i - m_i)^2} e^{-\frac{1}{2\sigma_q^2}(v_q - m_q)^2} .$$

Здесь удобнее оперировать длинами и углами, поэтому перейдем к полярной системе координат

$$v_i = r \cos \varphi, \quad v_q = r \sin \varphi ,$$

и тогда вероятность правильного приема определится элементарно

$$P_{\text{прав.}} = \int_{-\pi/8}^{\pi/8} \int_0^{\infty} w(r \cos \varphi, r \sin \varphi) r dr d\varphi .$$

Далее, подставляя плотность в явном виде, получим

$$P_{\text{прав.}} = \int_{-\pi/8}^{\pi/8} \int_0^{\infty} \frac{1}{2\pi \sigma^2} e^{-\frac{1}{2\sigma^2}(r \cos \varphi - E_s)^2} e^{-\frac{1}{2\sigma^2}(r \sin \varphi)^2} r dr d\varphi .$$

Переходя к стандартной переменной $q^2 = \frac{2 E_s}{N_0}$ и группируя слагаемые в

показателе экспонент, можно получить

$$P_{\text{прав.}} = \frac{1}{2\pi} \int_{-\pi/8}^{\pi/8} e^{-\frac{q^2}{2} \sin^2 \varphi} \int_0^{\infty} e^{-\frac{1}{2}(r-q \cos \varphi)^2} r dr d\varphi .$$

Вероятность символьной ошибки равна

$$P_{\text{ош. дем.}} = 1 - P_{\text{прав.}} .$$

Оставшиеся символы $S_i, i \neq 0$ рассматривать нет смысла, т. к. все символы равновероятные и при детальном рассмотрении сумма из восьми одинаковых вероятностей разделится на восемь. Получившаяся формула неудобна для численных расчетов, поэтому сделаем аппроксимацию. Для этого отдельно рассмотрим интеграл

$$J(\varphi) = \int_0^{\infty} e^{-\frac{1}{2}(r-q \cos \varphi)^2} r dr .$$

Если $\cos \varphi \neq 0$, то данный интеграл для большого q можно аппроксимировать методом Лапласа [4]

$$J(\lambda) = \int_a^b f(x) e^{\lambda S(x)} dx \approx \sqrt{-\frac{2\pi}{\lambda S''(x_0)}} f(x_0) e^{\lambda S(x_0)} .$$

Здесь

$$f(x) = x, \quad S(x) = -\frac{1}{2} \left(\cos \varphi - \frac{x}{q} \right)^2, \quad \lambda = q^2, \quad x_0 = q \cos \varphi .$$

Величина $x = x_0 \in (a, b)$ обращает показатель экспоненты в ноль. В итоге получаем

$$J(\varphi) \approx \sqrt{-\frac{2\pi}{q^2 - \frac{1}{q^2}}} q \cos \varphi e^{\lambda_0} = \sqrt{2\pi} q \cos \varphi .$$

Точка $x_0 = q \cos \varphi$ при $q \gg 1$ и $\varphi \in (-\pi/8, \pi/8)$ принадлежит интервалу интегрирования $(0, \infty)$ — этим и удобно рассматривать вероятность правильного приема, а не вероятность символьной ошибки. Таким образом, последовательно имеем

$$P_{\text{прав.}} \approx \frac{q}{\sqrt{2\pi}} \int_{-\pi/8}^{\pi/8} e^{-\frac{q^2}{2} \sin^2 \varphi} \cos \varphi d\varphi = \frac{2}{\sqrt{\pi}} \int_0^{\frac{q}{\sqrt{2}} \sin \frac{\pi}{8}} e^{-t^2} dt = 1 - \operatorname{erfc} \left(\frac{q}{\sqrt{2}} \sin \frac{\pi}{8} \right) ,$$

$$P_{\text{ош. дем.}} = 1 - P_{\text{прав.}} \approx \operatorname{erfc} \left(\frac{q}{\sqrt{2}} \sin \frac{\pi}{8} \right) .$$