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.н.

__________ А. В. Новиков

2020

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

56 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* и *NumPy*.

Оглавление

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, IvanovAA_код_1B5.jpg.
- Заполнить поля в файле найденными ответами и сдать файл с ответами, IvanovAA_код_1B5.xlsx.

Параметр "код" в имени файла означает двузначный код задания, **.

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

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

Здесь вектор $\vec{a}$ является строкой, состоящей из k информационных символов. Матрица $\mathbf{G}$ называется **порождающей матрицей** линейного блочного кода[1]. 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,** $0<R\le 1$. 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 $\vec{s}$ кодера[3]. 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)

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

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 01**1** 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 векторов[4], $d_{ab}=w(a+b)$, 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 $d_{ab}=w(a+b)=w(c)$.

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, **кодовое расстояние.** Очевидно, что для линейного кода кодовое расстояние $d_к$ равно весу самого "легкого" кодового вектора, отличного от нуля

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

Здесь C обозначает множество[5] векторов линейного кода. Для рассматриваемого кода, табл. 1, кодовое расстояние равно трем. Набор весов кода удобно сжать и представить в виде словаря {ключ: значение}

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

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

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

Здесь спектр состоит из трех элементов. Весовой спектр, в частности, позволяет определить кодовое 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 $d_к$ определяются кратности ошибок, которые код может гарантированно обнаружить и исправить. Кратность ошибки — это количество ошибочных элементов в принятом кодовом 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

$$q_о \leq d_к - 1 \quad .$$

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

Кратность гарантированного исправления[6]

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

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.

b) Задание, код 01

Случайным образом задается порождающая матрица $\mathbf{G}$.

Требуется определить:

- параметры линейного кода: n, k, r;
- кодовую таблицу кода C;
- весовой спектр кода $W_{\text{sp}}(C)$;
- 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

a) Сведения из теории

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

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

Для этого в порождающей матрице найдем три 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".

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

Будем такие столбцы называть опорными. Если в явном виде опорные столбцы обнаружить не удается, 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]

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

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). Проделанной группировке столбцов соответствует следующая перестановка

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

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 $k\times k$

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

в которой делается перестановка столбцов, обратная (6)

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

что в итоге дает

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

Тестом правильности нахождения проверочной матрицы является выполнение любого из двух равенств

$$\begin{gathered}\mathbf{G}\,\mathbf{H}^{T}=\mathbf{0}\quad,\\\mathbf{H}\,\mathbf{G}^{T}=\mathbf{0}\quad.\end{gathered}\tag{9}$$

Для корректно заданного линейного кода единственной матрице $\mathbf{G}$ соответствует единственная матрица $\mathbf{H}$, и наоборот. Это важно! Из правила кодирования (1) и равенств (9) следует, что[8]

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

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

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

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 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 $\vec{v}$ на сумму переданного вектора и вектора ошибки

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

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

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

но

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

поэтому если вектор ошибки $\vec{e}$, который сам по себе случаен, совпадет с любым из кодовых слов $\vec{s}$, то соответствующий синдром $\vec{c}$ будет равен нулю — такая ошибка при декодировании будет пропущена. Таким образом, множество кодовых слов C некоторого кода определяет также множество ошибок, которые не могут быть обнаружены данным кодом, а спектр $W_{\mathrm{sp}}(C)$ 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

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

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 кодовый вектор

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

и в соответствии с ним 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 взятого кодового слова, т. е. его весом, а минимальный ненулевой вес, как мы выяснили выше, 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 (из n по q) столбцов проверочной матрицы, начиная со столбцов по одиночке, $q=1$, 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 одинаковых столбцов проверочной матрицы гарантированно даст нулевые суммы пар таких столбцов[9], поэтому кодовое расстояние такого кода не может быть выше двух (если при этом все столбцы ненулевые, то кодовое расстояние будет равно двум).

Если код задан корректно, то в проверочной матрице всегда можно найти r опорных столбцов, которые образуют базис. Базисные столбцы — это такие столбцы, через которые могут быть выражены все остальные. Базисные столбцы по определению линейно независимы. Значит всегда можно найти систему из $r+1$ столбца, которая будет линейно зависимой, и поэтому сумма этих столбцов даст нулевой столбец. Это равнозначно наличию кодового вектора с весом $r+1$, откуда следует, что кодовое расстояние линейного кода ограничено сверху

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

$$d_{к} \leq r+1 \quad . \tag{13}$$

Это неравенство известно как **граница Синглтона[10],** которую также можно вывести из порождающей матрицы кода, записанной в систематической форме (а это всегда можно сделать)

$$\mathbf{G}_{\mathbf{s}} = \left[\mathbf{I}^{(k)}, \mathbf{Q} \right] \quad .$$

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 слова, а значит и на кодовое расстояние.

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

a) 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 $\vec{a}$.

10 На границе Синглтона лежат коды Рида-Соломона — недвоичные коды

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 $\mathbf{G}_{\boldsymbol{k}}$. 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

$$\mathbf{G}_{\boldsymbol{k}}=\begin{pmatrix}1 & 1 & 1\\ 1 & 0 & 1\\ 0 & 0 & 1\end{pmatrix}\sim\begin{pmatrix}1 & 1 & 1\\ 0 & 1 & 1\\ 0 & 0 & 1\end{pmatrix}\sim\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 1\\ 0 & 0 & 1\end{pmatrix}\sim\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} .$$

Сформируем из вектора $\vec{s}$ укороченный вектор

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

Запишем укороченную систему линейных уравнений

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

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 компонуется 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

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

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

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

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)

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

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

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

в карте соответствий M выбирается один и 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

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

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

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

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 ошибок[11], 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

$$\vec{e}\mathbf{H}^{T}=\vec{c} . \tag{14}$$

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 ошибок[12]

11 Все они являются решением недоопределенной системы линейных уравнений
12 В общем случае их количество равно r

$$\begin{cases} e_1 = c_1 + e_2 + e_5 \\ e_3 = c_2 + e_2 + e_5 + e_6 \\ e_4 = c_3 + e_2 + \quad\quad e_6 \end{cases} . \tag{15}$$

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 (15) следует, что для рассматриваемого кода векторы ошибок имеют следующую структуру

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

Это общее решение системы (14). 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)

$e_2 e_5 e_6$	$e_1 e_3 e_4$	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.

b) 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 вектор[13].

2.4. Неравенство Хемминга

a) 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

$$N_{\text{синдр.}}=2^r-1 \quad ,$$

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

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

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

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

$$N_{\text{ош.}} \le N_{\text{синдр.}} \quad \Rightarrow \quad \sum_{q=1}^{q_и} C_n^q \le 2^r - 1 \quad .$$

Для удобства упростим последнее неравенство

$$\sum_{q=0}^{q_и} C_n^q \le 2^r \quad . \tag{16}$$

Здесь $q_и$ является искомой наибольшей величиной, при которой неравенство еще выполняется. Кратность исправления можно искать последовательным перебором, начиная с нуля. Известно[14], 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

$$q_и \le \frac{r}{2} \quad ,$$

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

$$q_и \approx \frac{r}{4} \quad , \tag{17}$$

т. е. нечто среднее между крайними значениями 0 и $r/2$. Так что, можно начинать перебор со значения (17), 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 $q_и$ снизу вверх последовательно дает

14 Смотри предыдущие задания

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

Значит $q_{и}=4$, и кодовое расстояние двоичного кода $(63,43)$ не может быть выше девяти, т. е. $d_{к} \leq 2q_{и}+1=2\cdot 4+1=9$. Видим, что приближенная оценка по (17)

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

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 (16). Вероятность ошибки кратности q в рассматриваемом канале определяется формулой Бернулли

$$P(q)=C_n^q p^q (1-p)^{n-q} \quad , \quad 0 \leq q \leq n \quad , \qquad (18)$$

Значит вероятность кодовой ошибки на выходе декодера[15] равна

$$P_{\text{ош. дек.}} = \sum_{q=q_и+1}^{n} P(q) \quad . \qquad (19)$$

При $q_{и} \ll n$ вероятность (19) лучше вычислять от обратного

$$P_{\text{ош. дек.}} = 1 - \sum_{q=0}^{q_и} P(q) \quad . \qquad (20)$$

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

- $p=0.002$. Вычисления по (20) дают следующий результат

15 Здесь декодер настроен на исправление ошибок минимальной кратности

$$P_{\text{ош. дек.}}=1-\sum_{q=0}^{4}P(q)\approx 2.04\cdot 10^{-7} \quad .$$

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 формуле[16]

$$P_{\text{бит. дек.}}\approx P_{\text{ош. дек.}}\frac{d_{\text{к}}}{n} \quad . \tag{21}$$

Эта формула основана на выбранной ранее (стр. 19) 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 (21) дают

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

- $p=0.03$. Последовательные вычисления по (20) и (21) дают

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

- $p=0.05$. Вычисления по (20) и (21) дают

$$P_{\text{ош. дек.}}\approx 0.2067 \quad , \quad P_{\text{бит. дек.}}\approx 0.03 \quad .$$

- $p=0.079$. Вычисления по (20) и (21) дают

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

Таким образом, при приближении произведения $M[q]=np$, которое равно средней кратности ошибки в канале, 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.

16 Это граница сверху

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, $q_и$;
- 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 передача и прием M двоичных кодовых векторов фиксированного размера посредством набора из M попарно различных сигнальных импульсов, $s_i(t)$, $i=1,\dots,M$. Импульсам ставится в соответствие символ[17], который может принимать M значений. Часто вместо "значение символа" говорят "символ". Символы следуют через интервал времени T, при этом говорят, что "частота следования символов равна $1/T$". Символьная частота определяет **скорость телеграфирования,** которая измеряется в бодах. Одному боду соответствует скорость один символ в секунду. Скорость телеграфирования называют также "бóдовой скоростью передачи информации". Так как символ модуляции может принимать M разных значений, то в одном символе модуляции содержится $\log_2 M$ битов[18]. Множество из M значений образует **алфавит.** В качестве сигнальных импульсов будем рассматривать радиоимпульсы[19]

$$s_i(t)=\sqrt{\frac{2E_i}{T}}\cos(2\pi f_i t+\varphi_i)\ , \quad i=1,\dots,M\ , \tag{22}$$

17 Его называют символом модуляции, однако, мы часто будем говорить "символ"
18 Число *M* берут так, что двоичный логарифм дает натуральное число
19 Это используется в **радио**связи

заданные на интервале времени $0<t<T$ и имеющие энергию E_i, которая определена через интеграл от мгновенной мощности[20] $p_i(t)=s_i^2(t)$

$$E_i \stackrel{\text{def}}{=} \int_0^T p_i(t)dt = \int_0^T s_i^2(t)dt \ . \tag{23}$$

Величина E_i в (22) тем точнее соответствует энергии (23), чем лучше выполняется неравенство $f_i T \gg 1$, которое говорит о том, что количество периодов колебания, приходящихся на интервал времени T, должно быть много больше единицы. Однако, уже при $f_i T > 12$ относительная ошибка не превышает 1%. Каждое значение символа имеет свою вероятность p_i, $1 \le i \le M$, поэтому можно определить средние характеристики импульса (22)

- Среднюю энергию на один символ, $E_s = \sum_i E_i p_i$.
- Среднюю частоту, $f_0 = \sum_i f_i p_i$.

Средняя фаза, как правило, равна нулю и не несет особенного смысла. Определив среднюю частоту f_0, сигнал (22) можно записать в виде

$$s_i(t) = \sqrt{\frac{2E_i}{T}} \cos(2\pi f_0 t + \theta_i) \ , \tag{24}$$

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

$$\Delta f_i \ll f_0$$

отвечает за узкополосность формируемого сигнала. Сигнал (24) принято записывать в виде квадратурных компонентов[21]

$$s_i(t) = I_i(t)\cos(2\pi f_0 t) - Q_i(t)\sin(2\pi f_0 t) \ , \tag{25}$$

где

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

20 На один ом сопротивления

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

Квадратурные компоненты удобно отображать на плоскости в декартовых координатах (I,Q), где время t является параметром. Частота f_0 информации не содержит, и называется **несущей частотой.** Формула (25) определяет способ формирования радиосигнала с произвольным видом модуляции, который задается зависимостями $I_i(t)$ и $Q_i(t)$. Устройство, осуществляющее способ (25), называется **универсальным квадратурным модулятором.** Энергию на один символ E_s можно пересчитать в энергию на один бит

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

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

$$E_b = \frac{E_s}{R \log_2 M}. \qquad (26)$$

Здесь $R=k/n$ — скорость кодирования[22]. Применение корректирующих кодов приводит к уменьшению энергии на один символ. Это связано с тем, что 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 $R=1$. Если информация содержится в амплитуде[23] $\sqrt{E_i}$, то такой вид модуляции называют **амплитудной манипуляцией,** АМн. Дискретные уровни, как правило, задаются симметрично относительно нуля. Тем самым экономится средняя мощность, затрачиваемая на передачу одного бита. Если информация содержится в фазе φ_i, то такой вид модуляции называют **фазовой манипуляцией,** ФМн. Дискретные уровни, как правило, задают с равным шагом по окружности, т. к. это обеспечивает минимум вероятности ошибки при передаче и приеме равновероятных символов. Если информация содержится в частоте f_i, то такой вид модуляции называют **частотной манипуляцией,** ЧМн. Дискретные уровни, как правило, задают с равным шагом по оси частот. Шаг по частоте по возможности выбирается так,

---

22 Смотри стр. 5

23 Амплитуда пропорциональна корню из энергии (или мощности)

чтобы импульсы были взаимно ортогональными, т. е. чтобы их скалярное произведение было равно нулю

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

Если информация одновременно содержится и в амплитуде $\sqrt{E_i}$, и в фазе, то такой вид модуляции называют **квадратурной амплитудной модуляцией,** КАМ. При этом среднее значение комплексной амплитуды $\sqrt{E_i} \exp\left(j \varphi_i\right)$ равно нулю.

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 . Дисперсия шума пропорциональна его средней мощности. Всякий шум, который измеряется[24], ограничен по полосе частот. Тепловой шум приемника устроен так, что его средняя мощность распределена по всей рабочей полосе частот Δf равномерно, какой бы большой эта полоса ни была[25]. По этой причине считают, что на входе приемника имеется шум, мощность которого распределена равномерно по всей бесконечно большой полосе частот — такой шум называют **белым шумом.** После прохождения белым шумом некоторого блока с единичным коэффициентом передачи в полосе пропускания Δf и нулевым в полосе заграждения, шум становится белым с ограниченной полосой. Мощность такого шума конечна и равна

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

Здесь N_0 — спектральная плотность мощности, Вт/Гц или В^2/Гц. Она показывает сколько в среднем ватт приходится на полосу один герц. Чем выше эта плотность, тем выше уровень шума. Если обработка происходит в цифровом виде, то полоса обработки Δf и интервал дискретизации Δt связаны равенством $\Delta f = \dfrac{1}{2\Delta t}$. Выбранный шаг дискретизации фактически определяет полосу

24 Допустим, анализатором спектра
25 Ограничения наступают для частот в терагерцовой области и выше; смотри закон Планка

обработки, и если эта полоса не шире полосы пропускания аналоговой части приемника, то оцифрованный тепловой шум будет белым. Тепловой шум приемника $n(t)$ является одновременно и белым, и гауссовским. Это значительно упрощает нахождение алгоритма оптимального приема сигнала на фоне такого шума; однако, как это ни странно, белый гауссовский шум является самым "плохим" аддитивным шумом в плане искажения информации. Таким образом, принимаемый сигнал имеет вид

$$v(t)=s_i(t)+n(t) \quad . \tag{27}$$

Задача приемника заключается в угадывании номера $i=1,\ldots,M$ переданного символа по критерию минимальной вероятности ошибки. Приемник наблюдает $v(t)$ на интервале времени $0<t<T$, заранее зная множество импульсов $\{s_i(t)\}$. Доказано, что оптимальный прием сигнала на фоне **аддитивного белого гауссовского шума** (АБГШ) основан на вычислении всех скалярных произведений и выборе наибольшего

$$\max_{1\le i\le M}\left[\left(v(t),s_i(t)\right)\right]\to i_{\text{опт.}} \quad . \tag{28}$$

Здесь $i_{\text{опт.}}$ — номер опорного сигнала, дающего максимальное скалярное произведение, которое по смыслу является степенью похожести двух сигналов. Однако, принятый сигнал перед обработкой должен быть приведен к **единому** уровню, определяемому средней энергией опорных сигналов $\{s_i(t)\}$. Это особенно важно для модуляций типа КАМ-16 и выше. Алгоритмы автоматического масштабирования сигнала не входят в наш круг и заслуживают отдельного внимания. Оптимальность приема здесь понимается в смысле минимума вероятности ошибки. Стратегия приема (28) получена в предположении равной вероятности всех M значений символа. Выравнивания вероятностей добиваются скремблирующим кодированием. Приемник, работающий по правилу (28), называют **корреляционным приемником,** а блок, вычисляющий корреляционный интеграл

$$v_i=\left(s_i(t),v(t)\right)=\int_0^T s_i(t)v(t)\,dt \quad , \tag{29}$$

— **коррелятором.** Заметим, что здесь под словом "прием" понимают обработку принятого сигнала с целью извлечения информации, и поэтому каким процедурам подвергался сигал $v(t)$ до коррелятора — нас не интересует. Если сигнальных импульсов всего два, т. е. $M=2$, то процедура (28) упрощается до сравнения двух величин v_1 и v_2 между собой.

Корреляционный интеграл (29), если подставить в него (27), можно расписать на сигнальную и шумовую составляющие

$$v=\int_0^T s_i(t)\left(s_j(t)+n(t)\right)dt=v_{\text{с}}+v_{\text{ш}}\ , \qquad (30)$$

причем шумовая составляющая будет распределена по гауссовскому закону, как и шум $n(t)$, ее образующий. Сигнальная составляющая будет равна скалярному произведению двух сигнальных импульсов

$$v_{\text{с}}=\int_0^T s_i(t)s_j(t)dt=\sqrt{E_i E_j}r_{ij}\ , \qquad (31)$$

которое можно выразить через среднее геометрическое их энергий, умноженное на коэффициент корреляции r_{ij} этих импульсов[26]. При $i=j$ коэффициент корреляции равен единице и $v_{\text{с}}=E_i=E_j$. Так как шумовая компонента имеет гауссовский закон распределения с нулевым средним, то определим оставшийся параметр: дисперсию (среднюю мощность) шумовой компоненты[27]

$$\begin{gathered}\sigma_{\text{ш}}^2\stackrel{\text{def}}{=}\overline{v_{\text{ш}}^{\,2}}=\overline{\left(\int_0^T n(t)s_i(t)dt\right)^2}=\int_0^T\int_0^T\overline{n(t_1)n(t_2)}s_i(t_1)s_i(t_2)dt_1dt_2=\\ =\frac{N_0}{2}\int_0^T s_i^2(t)dt=\frac{E_iN_0}{2}\end{gathered}\ . \qquad (32)$$

Видим, что мощность шумовой компоненты пропорциональна спектральной плотности мощности шума, а также энергии опорного импульса, участвующего в обработке[28]. Говорят так: "мощность шума после обработки", или: "мощность шума на выходе коррелятора". Мощность сигнальной компоненты после обработки

26 Для модуляции ФМн-2 коэффициент взаимной корреляции достигает минимума и равен −1. Это своего рода исключение из правил, поэтому этот вид модуляции самый помехоустойчивый относительно АБГШ

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

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

$$v_c^2 = E_i^2 \quad ,$$

поэтому разумно определить отношение сигнал-шум по мощности после обработки

$$q^2 = \frac{2E_i}{N_0} \quad . \qquad (33)$$

Введенное отношение сигнал-шум соответствует обнаружению импульса $s_i(t)$. В системах передачи информации решается задача различения M импульсов, поэтому для удобства дополнительно вводят отношение сигнал-шум по так называемому **разностному сигналу [1, 2].** Видим, что мощность сигнальной компоненты растет как квадрат от энергии, в то время как мощность шумовой линейно зависит от энергии. Линейный рост вызван тем, что шум является белым. В среднем выходит так, что реализации $n(t)$ ортогональны опорному сигналу $s_i(t)$, и поэтому шумовой интеграл

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

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

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

под знаком которого стоит произведение одинаковых сигналов.

Обработка (28) подразумевает, что параметры опорных сигналов $s_i(t)$ в приемнике известны. Амплитуда и частота всегда известны, пусть и с небольшой ошибкой, ибо в противном случае прием будет невозможен. Начальная фаза φ_i в (24) — вот тот параметр, который может оставаться неизвестным в блоке обработки (28), т. е. фактически случайным, и прием при этом будет возможен. 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[29].** Приемники, в которых фаза

29 На практике их называют когерентными

известна априори, являются **когерентными**[30]**.** Оценка начальной фазы делается контуром фазовой автоподстройки частоты, ФАПЧ, что является отдельной важной темой. Для некогерентных систем количество корреляторов удваивается, потому что вводится квадратурный канал для компенсации фактора случайности фазы по принципу основного тригонометрического тождества

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

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

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 (33). Приведем формулы для модуляций с числом символов $M=\{2,4,8\}$. Используем дополнительную функцию ошибок

$$\operatorname{erfc}(z)=\frac{2}{\sqrt{\pi}}\int_z^{\infty}\exp\left(-t^2\right)dt \quad , \tag{34}$$

а также функцию Маркума[31]

$$\mathrm{Q}_1(a,b)=\int_b^{\infty} t\exp\left(\frac{t^2+a^2}{2}\right)\mathrm{I}_0(a\,t)\,dt \quad . \tag{35}$$

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

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

        - Некогерентная[32]

30 Их в природе не существует, это эталоны. Существует только математическая модель.
31 Смотри Приложение А
32 Подробности смотри в Приложении А

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

- Частотная, ЧМн.
  - Когерентная

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

  - Некогерентная

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

- Фазовая, ФМн.
  - Когерентная

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

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

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

- Число символов $M=4$. Здесь $E_s=2E_b$.
  - КАМ-4, ФМн-4
    - Когерентная

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

    - Частично-когерентная — ФАПЧ с дифференциальным кодированием дибитов[33]

33 Дибит — символ из двух битов, четырехзначный символ

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

- Число символов $M=8$. Здесь $E_s=3\,E_b$.
  - ФМн-8
    - Когерентная[34]

$$P_{\text{ош. дем.}}=1-\frac{1}{2\pi}\int\limits_{-\frac{\pi}{8}}^{\frac{\pi}{8}} e^{-\frac{q^2}{2}\sin^2\varphi}\int\limits_{0}^{\infty} r\,e^{-\frac{1}{2}(r-q\cos\varphi)^2}\,dr\,d\varphi\approx\operatorname{erfc}\left(\frac{q}{\sqrt{2}}\sin\frac{\pi}{8}\right)\quad,\quad q^2=\frac{2\,E_s}{N_0}\quad.$$

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

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 ошибки[35] 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 $P_{\text{бит. дем.}}=2\,p(1-p)$. При $M=8$ 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

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

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 которой[36] дана в п. 2.4.

34 Вывод смотри в Приложении Б
35 Смотри код Грея
36 Для жесткого декодера, hard-decision decoder

b) 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 требуемого отношения сигнал-шум $\dfrac{E_b}{N_0}$.

4. Регенерация цифрового сигнала при передаче на большие расстояния

a) Сведения из теории

Данная тема актуальна при рассмотрении проблемы передачи цифровой информации[37] 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 — биты[38], 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

---

37 То есть информации, квантованной как по времени, так и по уровню
38 В подавляющем большинстве случаев исходная информация представлена в виде последовательности битов

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.

Будем рассматривать такие НРП, регенераторы в которых делают лишь простое усиление импульсов без коррекции их формы[39]. 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 оконечным 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 оконечным приемником Rx.

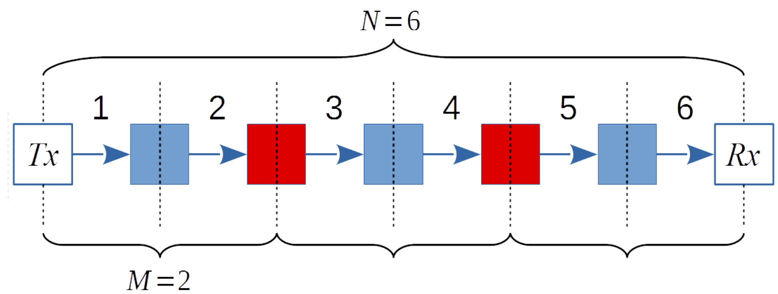

Рис 1. Структурная схема системы передачи информации с пятью регенераторами, два из которых — обслуживаемые.

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

39 Но, возможно, с преобразованием частоты

трансформатора. При этом шум на выходе преобразователя образуется как результат возможной междуканальной и индустриальной помехи — такие помехи за счет скремблирующих кодов в каждом канале в полосе частот приемника можно считать белым шумом.

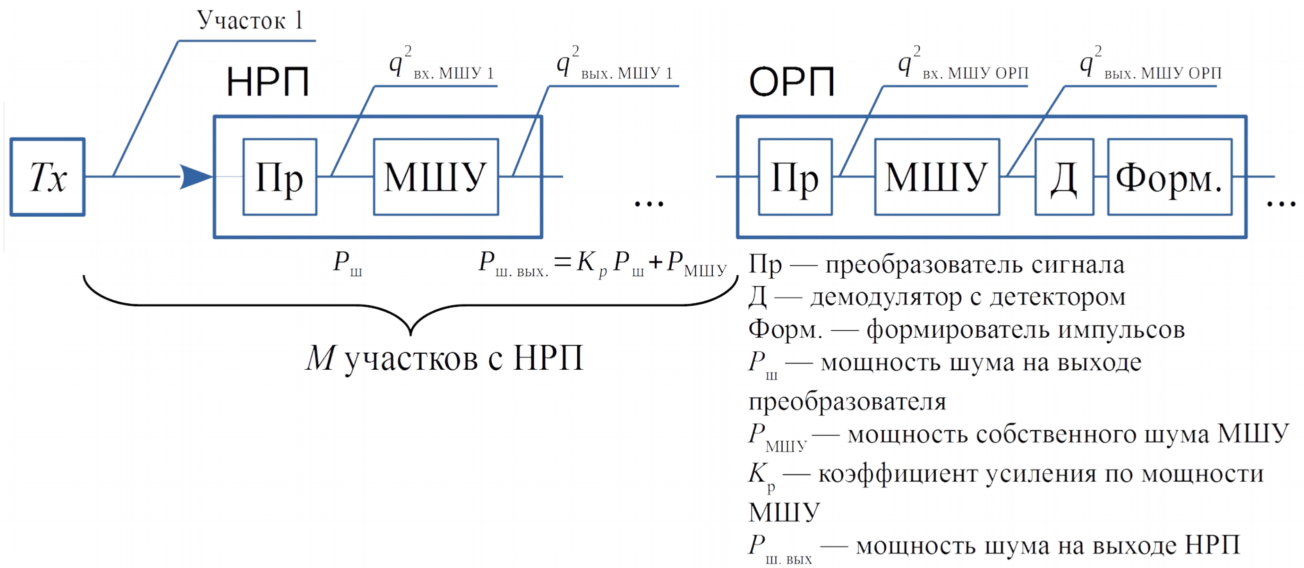

Рис 2. Разделение приемной части на преобразователь и МШУ. Отличие НРП от ОРП

- 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 $P_{\text{ср}}$ к мощности шума $P_{\text{ш}}$

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

Мощность шума определяется через шумовую полосу[40] приемника $B_{\text{ш}}$ и спектральную плотность мощности шума $n(t)$ N_0 как $P_{\text{ш}} = N_0 B_{\text{ш}}$. 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

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

Проходя через МШУ, как сигнальная часть, так и шумовая усиливаются по мощности в K_p раз, однако за счет усилительных элементов[41] добавляется внутренний шум мощностью $P_{\text{МШУ}}$, что определяет сигнал-шум на выходе МШУ

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

Здесь $F_{\text{ш}} = 1 + \frac{P_{\text{МШУ}}}{K_p P_{\text{ш}}}$ — шум-фактор[42] 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

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

На выходе МШУ ОРП формируется целевое отношение сигнал-шум

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

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.

41 И других нелинейных элементов, в основном, смесителей при преобразовании частоты
42 Коэффициент шума

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

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

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 оконечный приемный, поэтому вероятность ошибки p

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

Здесь вероятность битовой ошибки на выходе ОРП $p_{\text{ош. ОРП}}$ вычисляется исходя из отношения сигнал-шум $q^2_{\text{вых. МШУ ОРП}}$ 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 Tx. 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

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

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

b) Задание, код 06

Случайным образом задаются: общее количество участков N, количество участков с НРП M, отношение сигнал-шум на входе первого МШУ $q^2_{\text{вх. МШУ 1}}$, 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 и по точной,
- Требуемое отношение сигнал-шум (в дБ) на входе первого МШУ $q^2_{\text{вх. МШУ 1}}$ для обеспечения битовой вероятности ошибки $p_{\text{ош. итоговая}}$ при условии, что:
  - Все блоки — НРП, т. е. $M=N$,
  - Все блоки — ОРП, т. е. $M=1$.

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

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

5. Кодирование источника

5.1. Коды Хаффмана и Шеннона-Фано

a) 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 $p(x_i)$, $1 \le i \le m$, 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

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

Своеобразной аксиомой здесь является выражение

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

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 $i \to x_i$, 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 нуля[43] 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 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

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

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**

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

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

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

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

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

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 пятеричного 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.

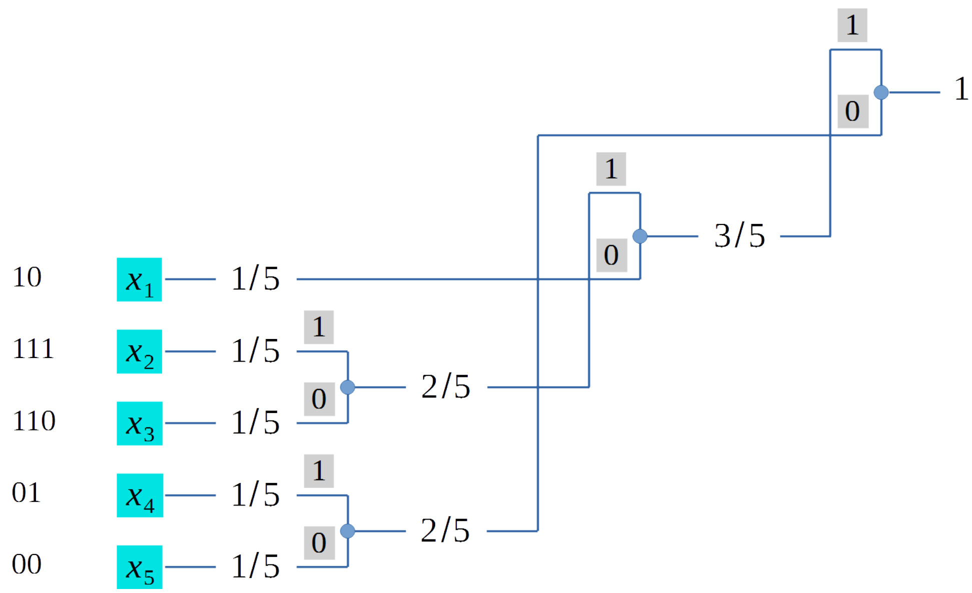

Рис. 3 Код и дерево Хаффмана при кодировании пятеричного символа, принимающего равновероятные значения

Видим, что свойство префикса выполняется, средняя длина кодового слова равна $L=2{,}4$ бита, что несколько больше энтропии источника $H(X)=\log_2 5\approx 2{,}32$ бит/символ. Дефект вызван тем, что вероятности не кратны отрицательным степеням двойки. При кодировании может возникнуть две одинаковые вероятности, например, $2/5$ на рис. 3, и здесь мы вольны любую из $2/5$ объединить с $1/5$. Это приводит к неоднозначности кода в смысле конкретных значений кодовых слов; средняя же длина кода и свойство префикса сохраняются. Таким образом, даже для символа с равновероятными значениями может существовать неравномерный код, чуть более экономный чем, равномерный. Например, для кодирования пятеричного символа равномерным кодом потребовалось бы как минимум $\lceil\log_2 5\rceil=3$ три бита[44], а код Хаффмана позволил обойтись $2{,}4$ битами[45]. Оказывается, что некогда аксиоматически введенная величина

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

показывает длину кодового слова для кодирования значения x_i некоторым идеальным кодом с минимально возможной средней длиной. Так как

44 Здесь использовалось округление вверх, в "потолок"

45 Это справедливо и актуально при передаче большого количества символов и при условии, что вероятности сохраняют свое значение

$\log_2 5 \approx 2{,}32$ 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.

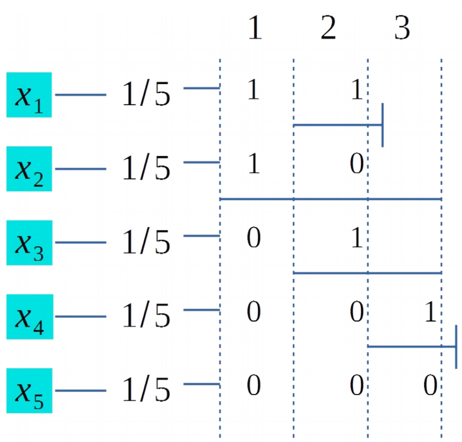

Рис. 4 Кодирование кодом Шеннона-Фано пятеричного символа с равновероятными значениями

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 оптимальными[46]. 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** одинаковой[47]
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

46 Субоптимальными
47 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 $c^{\langle i\rangle}$, 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

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

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 $p(x_i/x_j)$

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

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

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

Проиллюстрируем влияние зависимости на перекос 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

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

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

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

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

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

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

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

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

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

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

$$0\le H_n(X)\le n\log m=nH_{1\max} \quad ,$$

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

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

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 $(x_i x_j \ldots x_s)$, что и определяет главенствующую роль распределения вероятностей в оценке возможностей экономного кодирования или, что эквивалентно, информационной энтропии .

b) 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 $\operatorname{erfc}(1)$

```
0.15729920705028516
```

Функция Маркума может быть вычислена в программах *Matlab* или *Octave*[48]

$$\mathrm{Q}_1(a,b)=\operatorname{marcumq}(a,b) \quad ,$$

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

$$1-\mathrm{Q}_1(a,b)=\mathrm{stats.ncx2.cdf}\left(b^2,2,a^2\right) \quad .$$

Также можно использовать любые программы, в которые встроена функция распределения cdf()[49] закона распределения Райса[50].

Данные для проверки правильности вызова функции:

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

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

48 Требуется Signal Processing Toolbox в Matlab или пакет signal в Octave
49 CDF – Cumulative Distribution Function
50 Нецентрального хи-квадрат распределения с двумя степенями свободы

$$Q(1,2)=0.2690120600359135 \quad ,$$

$$Q(2,1)=0.9181076963694064 \quad .$$

Пороговый уровень $v_п$ находится путем численного решения уравнения[51]

$$\mathrm{I}_0(q\,v_п)=\exp\left(\frac{q^2}{2}\right) \quad .$$

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

$$\mathrm{F}(v)=\left|\mathrm{I}_0(q\,v)-\exp\left(\frac{q^2}{2}\right)\right|$$

функцией $\mathrm{minimize}(\mathrm{F}, v_0)$. 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

$$\mathrm{I}_{0.sc}(x) \stackrel{\mathrm{def}}{=} I_0(x)\exp(-x) \quad , \quad x\geq 0 \quad .$$

Тогда минимизации подлежит следующий модуль разности

$$\mathrm{F}(v)=\left|\mathrm{I}_{0.sc}(q\,v)-\exp\left(\frac{q^2}{2}-q\,v\right)\right| \quad .$$

---

51 $\mathrm{I}_0(x)$ – функция Бесселя первого рода нулевого порядка

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

$$\mathrm{I}_{0.sc}(q\,v)=\exp\left(\frac{q^2}{2}-q\,v\right) \quad ,$$

и сформировать модуль разности заново

$$\mathrm{F}(v)=\left|\ln\left[\mathrm{I}_{0.sc}(q\,v)\right]-q\left(\frac{q}{2}-v\right)\right| \quad .$$

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 `sp.ive(0, x)` означает масштабированную функцию Бесселя $\mathrm{I}_{0.sc}(x)$.

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

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

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

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

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

Уровень на выходе коррелятора состоит из суммы сигнальной части $v_\text{с}$ и шумовой $v_\text{ш}$. 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

$$v_{i\text{с}}=\frac{A^2T}{2}=E_s\ .$$

Здесь считатся, 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

$$m_i=\overline{v_i}=v_{i\text{с}}=E_s\ ,$$

$$\sigma_i^2=\overline{(v_i-\overline{v_i})^2}=\int_0^T\int_0^T \overline{n(t_1)n(t_2)}A\cos(2\pi f t_1)A\cos(2\pi f t_2)dt_1dt_2=$$
$$=\frac{A^2N_0}{2}\int_0^T \cos^2(2\pi f t)dt=\frac{A^2TN_0}{4}=\frac{E_sN_0}{2}$$

$$m_q=\overline{v_q}=0\ ,\quad \sigma_q^2=\sigma_i^2=\sigma^2=\frac{A^2TN_0}{4}=\frac{E_sN_0}{2}\ .$$

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

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

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

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

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

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

и тогда вероятность правильного приема определится элементарно

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

Далее, подставляя плотность в явном виде, получим

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

Переходя к стандартной переменной $q^2 = \dfrac{2E_s}{N_0}$ и группируя слагаемые в показателе экспонент, можно получить

$$P_{\text{прав.}}=\frac{1}{2\pi}\int\limits_{-\pi/8}^{\pi/8} e^{-\frac{q^2}{2}\sin^2\varphi}\int\limits_0^{\infty} e^{-\frac{1}{2}(r-q\cos\varphi)^2} r\,dr\,d\varphi \quad .$$

Вероятность символьной ошибки равна

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

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

$$J(\varphi)=\int\limits_0^{\infty} e^{-\frac{1}{2}(r-q\cos\varphi)^2} r\,dr \quad .$$

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

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

Здесь

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

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

$$J(\varphi)\approx\sqrt{-\frac{2\pi}{q^2\frac{-1}{q^2}}}q\cos\varphi\, e^{\lambda 0}=\sqrt{2\pi}q\cos\varphi \quad .$$

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

$$P_{\text{прав.}} \approx \frac{q}{\sqrt{2\pi}} \int\limits_{-\pi/8}^{\pi/8} e^{-\frac{q^2}{2}\sin^2\varphi} \cos\varphi \, d\varphi = \frac{2}{\sqrt{\pi}} \int\limits_{0}^{\frac{q}{\sqrt{2}}\sin\frac{\pi}{8}} e^{-t^2} \, dt = 1 - \operatorname{erfc}\left(\frac{q}{\sqrt{2}}\sin\frac{\pi}{8}\right) \quad ,$$

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