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", 2020.

55 с.

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.

Включает генератор заданий и проверочный модуль для проверки ответов. Программные компоненты написаны на языке Python. Использовались модули: scipy, numpy, python-docx.

Сборник является значительно переработанной версией кафедрального сборника контрольных работ, опубликованного в виде приложения к учебному пособию "Теория и техника передачи информации", https://edu.tusur.ru/publications/1750.

Оглавление

1. Общие требования к выполнению заданий

Каждому студенту выдается вариант задания со случайной параметрами в виде файла формата Excel. В этом же файле заведены поля для ввода ответов. Требуется:

- Заполнить поля в файле найденными ответами и сдать файл с ответами, ИвановИИ_код_1В5.xlsx.
- Сформировать файл с ходом решения: оформить "от руки" и перевести в формат jpeg, дав ему такое же имя, как и файлу Excel.

Параметр "код" в имени файла означает код задания.

В ответах числа с плавающей точкой округлять и давать с шестью знаками после запятой. Например 0.0042014578 переводить в $4.201458 \cdot 10^{-3}$, 478.231807 переводить в $4.782318 \cdot 10^{2}$ и т. п.

Ход решения будет проверен при условии, что в файле с ответами все ответы будут правильными. В итоге, при наличии верных ответов и хода решения будет проставлен один балл за ответы и балл от 1 до 4 за ход решения, итого: от 2 до 5 баллов за задание.

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

$$\vec{s} = \vec{a}\,\mathbf{G} \quad . \tag{1}$$

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.** Она принадлежит полуинтервалу $0<R\le 1$ и 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 определения кодовой таблицы, которая показывает взаимно-однозначное соответствие между входом $\vec{a}$ и выходом $\vec{s}$ кодера[3].

Пусть порождающая матрица кода имеет вид

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

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, совпадающие с соответствующими строками порождающей матрицы — в таблице фон этих строк затемнен. Кодирование остальных векторов осуществляется просто: складываются те строки порождающей матрицы, которые соответствуют единицам в информационном векторе. Например, вектор 011 кодируется суммой двух нижних строк порождающей матрицы

---

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

2 Бит — двоичный символ

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

$$101010+001101=100111 \quad .$$

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".

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

Информационный вектор	**Кодовый вектор**	**Вес 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. Вектора 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 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 вес кодовых векторов линейного кода играет ключевую роль в плане определения корректирующей способности кода. Это так, потому что для линейного кода сумма двух кодовых векторов дает кодовый вектор. 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.

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 одним 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.

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

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

Здесь C обозначает множество[4] векторов линейного кода. Для рассматриваемого кода кодовое расстояние равно трем.

Набор весов кода удобно сжать и представить в виде словаря {ключ: значение}

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

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

в котором указаны вес w кодового вектора и количество векторов N_w 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

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

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 $d_{\text{к}}$ 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 весом вектора ошибки, $q=w(\vec{e})$.

Кратность гарантированного обнаружения

$$q_{\text{о}}\le d_{\text{к}}-1\ .$$

Кратность гарантированного исправления

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

Для кода C в табл. 1 соответствующие кратности равны 2 и 3, соответственно. Таким образом, рассмотренный код может обнаруживать все ошибки кратности не выше второй, и исправлять все ошибки кратности не выше первой. При вычислении кратности исправления результат деления на два следует округлить вниз (округление типа "floor" – от англ. "пол", "низ").

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). Выпишем порождающую матрицу кода

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

Далее в порождающей матрице следует найти три столбца с одной единицей, т. е. три столбца с единичным весом. В данном случае это столбцы под номерами 2, 5 и 6

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

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 слов $\vec{s}\in C$ не

меняется! Информационные слова $\vec{a}$ здесь игнорируются. Опорные столбцы обязаны найтись, ибо в противном случае код будет задан некорректно. Опорные столбцы — это своего рода базис.

Далее группируем опорные столбцы в левое положение так, чтобы они образовали единичную матрицу $\mathbf{I}^{(3)}$ размером 3×3 [5]

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

Индексом $\mathbf{s}$ обозначено то, что в такой форме задается **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 $\mathbf{G}_s$, куда будут вставлены соответствующие столбцы матрицы $\mathbf{G}$. Например, второй столбец $\mathbf{G}$ стал первым столбцом $\mathbf{G}_s$.

Далее, на основании (5) формируется проверочная матрица

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

в которой делается перестановка столбцов, обратная (6)

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

Тестом правильности нахождения проверочной матрицы является любое из двух равенств

---

5 В общем случае размером $k\times k$

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

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[6]

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

Таким образом, кодовый вектор при умножении его на проверочную матрицу дает нулевой вектор $\vec{0}$. Так как равенство (10) верно для любого информационного вектора $\vec{a}$, а матрицы $\mathbf{G}$ и $\mathbf{H}$ связаны равенствами (9) взаимно-однозначно, то любой вектор, не принадлежащий коду, $\vec{v}\notin C$, даст ненулевой вектор-произведение. Данный признак позволяет определять принадлежность произвольного вектора заданному линейному коду.

Произведение

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

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.

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

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

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

6 Известно общее правило $(\mathbf{u}\,\mathbf{v})^{T}=\mathbf{v}^{\boldsymbol{T}}\mathbf{u}^{\boldsymbol{T}}$

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

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 ошибки соответствует нулевой вектор ошибки (ошибка отсутствует), и здесь синдром тривиально нулевой, т. е. если $\vec{e}=0$, то $\vec{c}=0$.

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

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. Тех столбцов, номера которых соответствуют номерам единиц в кодовом слове $\vec{s}$. 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

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

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[7], 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 из этих столбцов даст ноль. Это равнозначно наличию кодового вектора с весом $r+1$, откуда следует, что кодовое расстояние линейного кода ограничено сверху

$$d_{\text{к}} \le r+1 \quad . \tag{13}$$

Это неравенство известно под названием **граница Синглтона[8].** Его также можно получить из порождающей матрицы кода, записанной в систематической форме (а это всегда можно сделать)

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

Здесь число столбцов матрицы $\mathbf{Q}$ равно r, поэтому в лучшем случае в строке этой матрицы r ненулевых элементов, а в единичной матрице $\mathbf{I}^{(k)}$

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

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

один ненулевой элемент. Именно наличие единичной матрицы, у которой всего один ненулевой элемент в каждой строке, позволяет проанализировать ограничение сверху на вес самого "легкого" кодового слова.

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 $\vec{a}$.

Основная соль декодирования заключена в первой части. Оценка кодового вектора $\vec{s}$ по принятому вектору $\vec{v}$ по существу означает определение такого вектора ошибок $\vec{e}$, который корректирует вектор $\vec{v}$ в кодовый вектор $\vec{s}=\vec{v}-\vec{e}$, дающее 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 $\vec{c}=\vec{v}\,\mathbf{H}^{T}$, значит после коррекции ошибки синдром обязан стать равным нулю, при этом верно ли в итоге была исправлена ошибка — не важно.

Определение информационного вектора по кодовому

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

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

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

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

Возьмем для разнообразия 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

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

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

$$\vec{s_k}=s_1 s_2 s_3=010 \quad .$$

Запишем укороченную систему линейных уравнений

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

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 компонуется информационный вектор

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

который совпадает с соответствующим вектором в предпоследней строке кодовой таблицы, табл. 1.

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

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

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

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

Векторов-решений $\vec{e}$ поставленной задачи — множество, потому что число строк проверочной матрицы меньше числа столбцов (она не является квадратной). Косвенно 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, а значит дающих в итоге нулевой синдром.

Естественно, задачу коррекции можно решать методом Монте-Карло, т. е. случайным перебором векторов $\vec{e}$. Задачу коррекции также можно решать, заранее составив карту 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 слова кода, и одновременно такие ошибки, которые не могут быть обнаружены кодом. Ненулевым синдромам соответствуют слова, которые коду не принадлежат, и одновременно ошибки, которые обнаруживаются кодом.

Вычислив по принятому вектору синдром

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

в карте соответствий M выбирается вектор ошибки $\vec{e}$. По какому правилу выбирать — вопрос отдельный. 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, 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 по модулю два которых даст нулевой столбец — это три линейно зависимых столбца, поэтому кодовое расстояние кода меньше четырех, что в итоге дает три. В итоге, данный код может гарантированно исправить все ошибки вплоть до первой кратности включительно; попросту говоря, код исправляет все однократные ошибки.

Вычислим синдромы для всех однократных ошибок

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

Для двоичных кодов соответствующие синдромы совпадут 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

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

Из табл. 2 следует, что вектор ошибки равен

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

Корректируем принятый вектор

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

Ошибка исправлена. В действительности, найденному синдрому соответствует $2^6/2^3=8$ разных векторов ошибок[9], 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 вектора ошибок для найденного выше синдрома 111. Для этого решим относительно $\vec{e}$ систему уравнений

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

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

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

Структура последнего уравнения определяется оставшимися столбцами проверочной матрицы под номерами 2, 5 и 6

9 Все они являются решением недоопределенной системы линейных уравнений

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

Здесь первый столбец определяет три элемента e_2 в (15), два элемента e_5 сверху и два элемента e_6 снизу. Перечисленные элементы являются свободными, и т. к. эти три элемента являются битами, то число разных вариантов равно $2^3=8$, что и говорит о восьми разных векторах ошибок, приходящихся на один синдром. Из (15) следует, что для данного кода вектора 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, показанный в табл. 3.

Найденные вектора 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 $M=\{\vec{c},\{\vec{e}\}\}$ также называется **классами смежности.** Каждый класс характеризуется синдромом. Классу нулевого синдрома соответствуют необнаруживаемые ошибки, или кодовые слова линейного кода, в зависимости от того, как посмотреть.

Таблица 3 Все вектора 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 кода соответствует несколько векторов ошибок. Эти вектора являются решением недоопределенной системы линейных уравнений. Решение существует в том числе и для нулевого синдрома — этим решением будет множество кодовых слов кода.

b) Задание, код 03

Случайным образом задается порождающая матрица **G** и принятый кодовый вектор.

Требуется:

- По заданной порождающей матрице определить проверочную матрицу кода;
- По проверочной матрице определить кодовое расстояние кода;
- Декодировать заданный принятый кодовый вектор[10].

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 известных заранее количествах различных ненулевых синдромов

$$2^r-1 \ ,$$

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

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

Естественно, что для гарантированного исправления всех ошибок вплоть до кратности $q_{\text{и}}$ включительно, последняя величина должна быть меньше либо равна количеству разных ненулевых синдромов

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

$$\sum_{q=1}^{q_и} C_n^q \le 2^r - 1 \ ,$$

где C_n^q — количество сочетаний из n по q . Для удобства последнее неравенство можно упростить до следующего

$$\sum_{q=0}^{q_и} C_n^q \le 2^r \ . \qquad (16)$$

Здесь $q_и$ является искомой наибольшей величиной, при которой неравенство еще выполняется. Кратность исправления следует искать последовательным перебором, начиная с нуля.

Известно[11], 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

$$q_и \le \frac{r}{2} \ ,$$

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

$$q_и \approx \frac{r}{4} \ , \qquad (17)$$

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

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

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

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

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

Значит вероятность кодовой ошибки на выходе декодера равна

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

При малых $q_{\text{и}}$ относительно n вероятность (19) быстрее вычислить от обратного

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

Продолжим рассматривать код $(63,43)$. Зададим несколько значений p .

- Вероятность битовой ошибки в канале $p=0.002$. Вычисления по (20) дают следующий результат

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

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[12]

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

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

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

- Для вероятности битовой ошибки в канале $p=0.03$ последовательные вычисления по (20) и (21) дают

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

- Для вероятности битовой ошибки в канале $p=0.05$ последовательные вычисления по (20) и (21) дают

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

- Для вероятности битовой ошибки в канале $p=0.079$ последовательные вычисления по (20) и (21) дают

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

12 Это граница сверху (так ли?)

Таким образом, при приближении произведения $M[q]=np$, которое равно средней кратности ошибки в канале, к кратности исправления $q_{и}$, 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, $q_{и}$;
- 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 различных сигнальных импульсов, $s_i(t)$, $i=1,...,M$. Импульсам ставится в соответствие символ[13], который может принимать M значений. Часто вместо "значение символа" говорят "символ". Символы следуют через интервал времени T. 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 "бо́довой скоростью передачи информации". Так как символ модуляции может принимать M разных значений, то в одном символе

13 Его называют символом модуляции, однако, мы часто будем говорить "символ"

модуляции должно содержаться $\log_2 M$ битов. Множество из M значений образует **алфавит.**

В качестве сигнальных импульсов будем рассматривать радиоимпульсы[14]

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

заданные на интервале времени $0<t<T$ и имеющие энергию E_i, которая определена через интеграл от мгновенной мощности[15] $p_i(t)=s_i^2(t)$

$$E_i=\int_0^T s_i^2(t)\,dt\ . \tag{23}$$

Величина E_i в (22), в целом, 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, $i=1,\dots,M$, поэтому можно определить средние характеристики сигнала (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

14 Это используется в **радио**связи
15 На один ом сопротивления

$$\Delta f_i \ll f_0$$

отвечает за узкополосность формируемого сигнала. Сигнал (24) принято записывать в виде квадратурных компонентов[16]

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

где

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

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

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

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

$$E_b=\frac{E_s}{R\log_2 M} \quad . \tag{26}$$

Здесь $R=k/n$ — скорость кодирования[17]. 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[18] $\sqrt{E_i}$, то такой вид модуляции называют **амплитудной манипуляцией,** АМн. Дискретные уровни,

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

17 Смотри стр. 5

18 Амплитуда пропорциональна корню из энергии (или мощности)

как правило, задают симметрично относительно нуля. Так экономится средняя мощность, затрачиваемая на передачу одного бита; также, если посмотреть со стороны частотного спектра, симметрия относительно нуля дает лучшее распределение мощности по частотному диапазону.

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[19] 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, 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. Как правило, среднее значение комплексной амплитуды $\sqrt{E_i}\exp(j\varphi_i)$ равно нулю.

Будем считать, что сигнал в процессе передачи и приема не искажается, а единственным мешающим фактором является тепловой шум приемника $n(t)$. Амплитуда n теплового шума распределена по гауссовскому закону с нулевым средним и некоторой дисперсией σ^2 . Дисперсия шума пропорциональна его средней мощности, т. е., фактически, дисперсия и мощность — это синонимы. Всякий шум, который измеряется[20], ограничен по полосе частот, — как минимум полосой пропускания измерительного прибора. Тепловой шум приемника устроен так, что его средняя мощность равномерно распределена по всей рабочей полосе частот приемника Δf , какой бы большой эта полоса ни была[21]. По этой причине считают, что на входе приемника имеется шум, мощность которого распределена

19 На фоне аддитивного белого гауссовского шума, АБГШ
20 Допустим, анализатором спектра
21 Ограничения наступают для частот в терагерцовой области и выше; смотри закон Планка

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 называют белым с ограниченной полосой. Мощность такого шума равна

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

Здесь N_0 — спектральная плотность мощности белого шума, Вт/Гц или $В^2$/Гц. Спектральная плотность 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 $\Delta f = \frac{1}{2\Delta t}$. Выбранный шаг дискретизации фактически определяет полосу 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

$$v(t) = s_i(t) + n(t) \quad . \qquad (27)$$

Задача приемника заключается в угадывании номера $i = 1, \ldots, M$ переданного символа, причем с минимальной вероятностью ошибки. Приемник наблюдает $v(t)$ на интервале $0 < t < T$, заранее знает множество $s_i(t)$ и параметры шума $n(t)$. Доказано, что оптимальный прием сигнала на фоне **аддитивного белого гауссовского шума** (АБГШ) основан на вычислении всех скалярных произведений и выборе наибольшего

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

Здесь $i_{\text{опт.}}$ — номер опорного сигнала, дающего максимум скалярного произведения. Скалярное произведение по смыслу является степенью похожести двух сигналов — коэффициентом корреляции, — но после того, как принятый

сигнал будет приведен к уровню, определяемому нормами опорных сигналов $s_i(t)$ для каждого i по отдельности. Эта тема не входит в наш круг и заслуживает отдельного внимания. Оптимальность здесь понимается в смысле минимума вероятности ошибки. Стратегия приема (28) получена в предположении одинаковой вероятности всех M значений символа; в системах передачи информации все символы с помощью кодирования[22] делают равновероятными.

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

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

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

$$v_{\text{с}}=\int_0^T s_i(t)s_j(t)dt=\sqrt{E_iE_j}r_{ij} \quad , \tag{31}$$

которое можно выразить через среднее геометрическое их энергий, умноженное на коэффициент корреляции r_{ij} этих импульсов. При $i=j$ коэффициент корреляции равен единице и $v_{\text{с}}=E_i=E_j$. При $i\neq j$, как правило, коэффициент корреляции равен нулю и поэтому $v_{\text{с}}=0$.

22 Скремблирования

Так как шумовая компонента имеет гауссовский закон распределения с нулевым средним, то определим оставшийся параметр — дисперсию или среднюю мощность шумовой компоненты[23]

$$\sigma_{\text{ш}}^2 \equiv \overline{v_{\text{ш}}^{\,2}} = \overline{\left(\int_0^T n(t) s_i(t)\, dt \right)^2} = \int_0^T \int_0^T \overline{n(t_1) n(t_2)}\, s_i(t_1) s_i(t_2)\, dt_1\, dt_2 = \\ = \frac{N_0}{2} \int_0^T s_i^{\,2}(t)\, dt = \frac{E_i N_0}{2} \quad . \tag{32}$$

Видим, что мощность шумовой компоненты пропорциональна спектральной плотности мощности шума, а также энергии опорного импульса, участвующего в обработке[24]. Говорят так: "мощность шума после обработки", или: "мощность шума на выходе коррелятора". Мощность сигнальной компоненты после обработки

$$v_{\text{с}}^{\,2} = E_i^{\,2} \quad ,$$

поэтому разумно определить отношение сигнал-шум по мощности после обработки

$$q^2 = \frac{2 E_i}{N_0} \quad . \tag{33}$$

Видим, что мощность сигнальной компоненты растет как квадрат от энергии, в то время как мощность шумовой — линейно зависит от энергии. Линейный рост мощности шума вызван тем фактом, 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

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

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

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

в под знаком которого стоит произведение одинаковых импульсов.

Обработка (28) подразумевает, что параметры опорных сигналов $s_i(t)$ в приемнике известны. Амплитуда и частота стоят особняком, потому что они всегда известны, пусть и с небольшой ошибкой, ибо в противном случае прием будет невозможен. Начальная фаза — вот тот параметр, который может быть неизвестным в блоке обработки (28), т. е. фактически случайным, и прием при этом будет успешным. Если в процессе приема начальная фаза опорного сигнала никак не оценивается, то такие системы передачи информации называются **некогерентными.** Если же начальная фаза оценивается, то — **частично-когерентными.** Системы, в которых фаза известна априори, называются **когерентными.** 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 системы реализуются для амплитудной манипуляции и частотной. Для всех модуляций, в которых информация заложена в начальную фазу, в том числе и для КАМ[25], приемник всегда будет когерентным, но число корреляторов будет все равно удвоенным, т. к. одним коррелятором можно обойтись лишь для двухуровневой ФМн. Два квадратурных канала в приемнике — это своего рода две оси декартовой системы координат.

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 сводку формул для модуляций с числом символов $M=\{2,4,8\}$. Будем использовать 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[26]

25 Квадратурная амплитудная модуляция
26 Смотри Приложение А

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

- Число символов $M=2$. Здесь $E_s=E_b$.
  - Амплитудная, АМн.
    - Когерентная

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

    - Некогерентная[27]

$$P_{\text{ош. дем.}}=\frac{1}{2}\exp\left(\frac{-v_{\text{п}}^2}{2}\right)+\frac{1}{2}\left[1-\mathrm{Q}_1(q,v_{\text{п}})\right] \quad , \quad q^2=\frac{4E_s}{N_0} \quad , \quad \mathrm{I}_0(q\,v_{\text{п}})=\exp\left(q^2/2\right) \quad .$$

  - 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 декодированием битов

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

27 Подробности смотри в Приложении А

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

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

    - Частично-когерентная — ФАПЧ с дифференциальным декодированием дибитов[28]

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

- Число символов $M=8$. Здесь $E_s=3E_b$.
  - ФМн-8
    - Когерентная[29]

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

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[30] 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 $P_{\text{бит. дем.}}=2p(1-p)$. 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

28 Дибит — символ из двух битов, четырехзначный символ
29 Смотри Приложение Б
30 Смотри код Грея

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

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 которой[31] дана в п. 2.4.

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).

Требуется:

- Подобрать скорость кодирования R так, чтобы обеспечить минимум требуемого отношения сигнал-шум $\frac{E_b}{N_0}$.

4. Регенерация цифрового сигнала при передаче на большие расстояния

a) Сведения из теории

Данная тема актуальна при рассмотрении проблемы передачи цифровой информации[32] на большие расстояния — сотни и тысячи километров, как правило, по медному или оптическому кабелю, или по воздуху (например, радиоинтерфейсу радиорелейных линий).

Регенерация означает восстановление чего-либо. В данном случае восстанавливаются двоичные символы — биты[33]. Восстанавливаются они из принимаемого сигнала, кодирующего биты согласно заданного закона модуляции.

31 Для жесткого декодера, hard-decision decoder

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

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

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. В связи с этим, регенерационные пункты — пункты, в которых производится регенерация импульсов — делят на **обслуживаемые** (ОРП) и **необслуживаемые** (НРП). В первых делается регенерация в полном смысле этого слова, т. е. с демодуляцией, детектированием и повторной модуляцией. 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 количества обслуживаемых регенераторов возможно 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[34]. Трассу передачи информации разобьем на $N>0$ участков, причем через каждые $M \le N$ 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 показана 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, включая Rx.

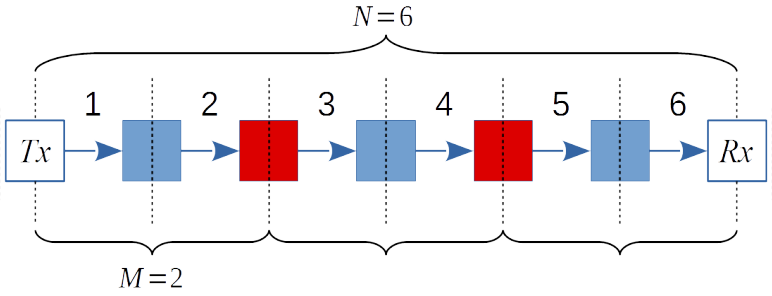

Рис 1. Структурная схема системы передачи информации с пятью регенераторами, два из которых — обслуживаемые.

Определим отношение сигнал-шум q^2.

34 Возможно, с преобразованием частоты

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. При приеме электрического сигнала вместо МШУ используется широкополосный видеоусилитель (например, технология Ethernet по витой паре), а в качестве преобразователя используется гальваническая развязка типа трансформатора.

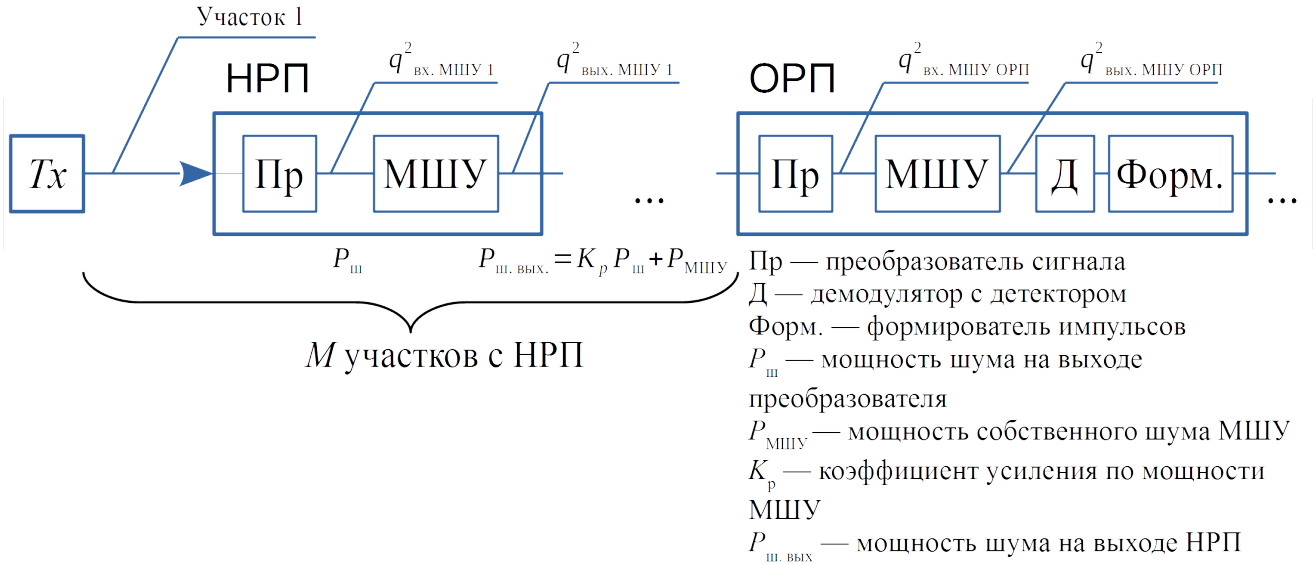

Рис 2. Разделение приемной части на преобразователь и МШУ. Отличие НРП от ОРП

- 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 $P_{\text{ш}}=\sigma^2_{\text{ш}}$

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

Мощность шума определяется через шумовую полосу[35] 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.

- Выход первого МШУ приемника

Проходя через МШУ, как сигнальная часть, так и шумовая усиливаются по мощности в K_p раз, однако за счет усилительных элементов[36] добавляется внутренний шум мощностью $P_{МШУ}$, что определяет сигнал-шум на выходе МШУ

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

Здесь $F_{ш}=1+\frac{\sigma^2_{\text{внутр. ш}}}{K_p\sigma^2_{ш}}$ — шум-фактор[37] МШУ, причем $F_{ш}>1$. Для удобства шум-фактор выражают в децибелах $F_{ш\,[дБ]}=10\lg F_{ш}$. 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

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

На выходе МШУ ОРП формируется целевое отношение сигнал-шум

35 Энергетическую полосу. Эту полосу следует отличать от полосы по критерию половинной мощности
36 И других нелинейных элементов, в основном, смесителей при преобразовании частоты
37 Коэффициент шума

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

которое будет определять вероятность ошибки при детектировании в текущем ОРП. На входе первого МШУ, идущего после текущего ОРП, соотношение сигнал-шум вернется к исходной величине $q^2_{\text{вх. МШУ 1}}$. Однако при прохождении нескольких ОРП происходит накопление битовых ошибок, которые могут быть обнаружены и исправлены декодерами корректирующих кодов. Как правило, такие декодеры ставятся лишь в оконечный ОРП.

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

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

Отношение сигнал-шум на выходе второго МШУ

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

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 здесь все НРП — одинаковые.

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

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

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 может быть получена из условия, что битовая ошибка на выходе системы передачи информации будет лишь при нечетных сочетаниях битовых ошибок в наборе из $m+1$ ОРП

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

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 $p_{\text{ош. итоговая}}$, используя точную формулу
- Требуемое отношение сигнал-шум (в дБ) на входе первого МШУ $q^2_{\text{вх. МШУ 1}}$ для обеспечения найденной 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.

$$q^2_{\text{вых. МШУ ОРП}} = \frac{2E_s}{N_0} \quad \text{и} \quad p_{\text{ош. ОРП}} = \frac{1}{2}\exp\left(\frac{-q^2_{\text{вых. МШУ ОРП}}}{4}\right) .$$

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

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

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[38] 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 то же самое, среднюю скорость генерации информации. Единицей измерения энтропии чаще является бит/символ, в этом случае основание логарифма берется равным двум

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

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

---

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

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). Или по другому: емкость алфавита X объемом m равна $\log m$. Причем максимальная производительность $H(X) = H_{\max}$ достигается при равновероятных значениях символа $p(x_i) = \frac{1}{m}$. В этом случае алфавит используется максимально эффективно. Это 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 того, чтобы осознать почему емкость троичного символа $1{,}58$ бит, требуется составить несколько двоичных кодов, вычислить **среднюю арифметическую** длину кодового слова и сравнить ее с величиной $1{,}58$. Показано, что никакой код не даст среднюю арифметическую длину кодового слова меньше, чем $\log_2 3 \approx 1{,}58$ бит. Приведем примеры.

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

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

Пример — в таблице ниже.

Значение	Вероятность	Двоичный код	Длина слова	Двоичный код	Длина слова

символа X	$P(X)$	№1	l , бит	№2	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, но в новом образовавшемся алфавите и коде свойство префикса все равно обязано выполняться, как ни крути.

Коды Хаффмана

Существует алгоритм кодирования, доставляющий оптимальный код, т. е. код с минимально возможной средней длиной $L_{\text{опт. код}}$ такой, что

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

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 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 последних символа объединяются в один, вероятности складываются
3. Верхнему символу присваивается логическая единица, нижнему — ноль.

Процедура 1-2-3 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.

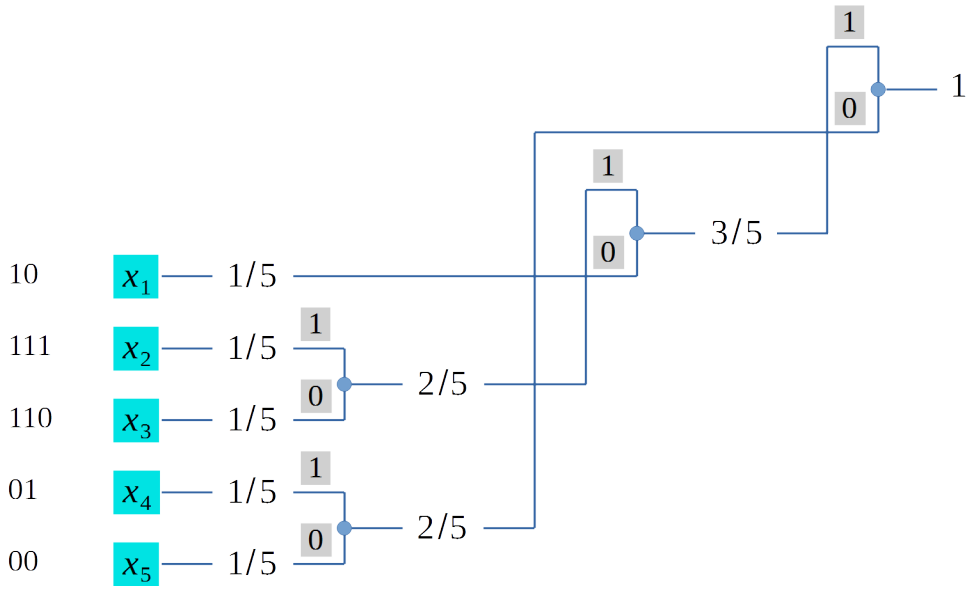

Рис. 3 Код и дерево Хаффмана при кодировании пятеричного символа, принимающего равновероятные значения

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[39], а код Хаффмана позволил обойтись $2{,}4$ битами[40]. 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 $\log_2 5 \approx 2{,}32$ 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.

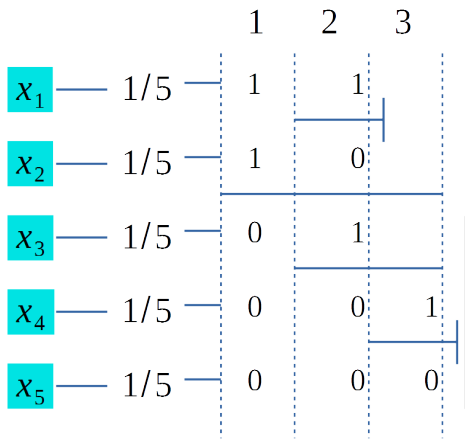

Рис. 4 Кодирование кодом Шеннона-Фано пятеричного символа с равновероятными значениями

Коды Шеннона-Фано

Коды Шеннона-Фано дают такой же результат, как и коды Хаффмана в плане средней длины кодового слова, или иногда — чуть хуже, т. е. они являются почти оптимальными[41], но свойство префикса в любом случае выполняется. Алгоритм кодирования следующий.

1. Символы сортируются в порядке убывания вероятностей

39 Здесь использовалось округление вверх, в "потолок"
40 Это справедливо и актуально при передаче большого количества символов и при условии, что вероятности сохраняют свое значение
41 Субоптимальными

2. Имеющееся множество символов делится на два подмножества так, чтобы сумма вероятностей в каждом из них была бы **максимально** одинаковой[42]
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 текущем 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 аналогично коду Хаффмана возможна неоднозначность. В частности на первом шаге в верхнее подмножество можно включить либо два символа, либо — три. Был выбран вариант с двумя. 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 Шеннона-Фано дает ту же самую среднюю длину $L=2,4$, 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 ничто не мешает последовательные значения x_i на выходе источника X сгруппировать по два $x_i x_j$ и обозначить двойной символ как некоторый вектор

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

42 Или, что эквивалентно, разность суммарных вероятностей для двух подмножеств была бы минимальной

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 $p(x_i/x_j)$

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

В случае, 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

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

Поясним 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

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

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

$$p(y_2)=p(x_1x_2)=p(x_1)p(x_2/x_1)=0,5\cdot 0,2=0,1\ ,$$

$$p(y_1)=p(x_1x_1)=p(x_1)p(x_1/x_1)=0,5\cdot 0,8=0,4\ .$$

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

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

Здесь требуется помнить, что символ — двойной, или записывать явно как бит/символ Y . Данную энтропию называют двухбуквенной, т. к. она определяется по парам значений базового источника X . Доказано, что n -буквенная энтропия неотрицательна и ограничена сверху

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

Относительно исходного символа 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 , т. е. за счет определенной выше парной зависимости путем объединения в пары возможно уменьшить среднюю длину кода до $0,86$ бит в пересчете на один символ X .

В итоге, к снижению энтропии и возможному увеличению эффективности кодирования приводит

- неравновероятность значений исходного источника X и
- статистическая зависимость этих значений,

однако последняя приводит к неравновероятности объединенных значений $(x_ix_j...x_s)$, что и определяет главенствующую роль распределения вероятностей или информационной энтропии соответствующего источника.

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 задание заново.

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

Дополнительная функция ошибок erfc() может быть без проблем вычислена в любой программе: Mathcad, Matlab, Octave, SciPy/Python путем вызова одноименной функции

```
import scipy.special as sp

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

Результат вычисления $\mathrm{erfc}(1)$

```
0.15729920705028516
```

Функция Маркума может быть вычислена в программах Matlab или Octave[43]

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

а также с помощью библиотеки SciPy языка Python

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

Также можно использовать любые программы, в которые встроена функция распределения cdf()[44] закона распределения Райса[45].

Данные для теста:

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

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

$Q(1,0)=1$,

$Q(1,2)=0.2690120600359135$,

$Q(2,1)=0.9181076963694064$.

Пороговый уровень $v_{\text{п}}$ находится путем численного решения уравнения[46]

$$\mathrm{I}_0(q\,v_{\text{п}})=\exp\left(\frac{q^2}{2}\right) .$$

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 $\text{minimize}(\mathrm{F},v_0)$. В качестве начального приближения рекомендуется брать $v_0=1$. 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]
```

7. 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

46 $\mathrm{I}_0(x)$ – функция Бесселя первого рода нулевого порядка

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

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

имеющий двустороннюю спектральную плотность мощности $N_0/2$, Вт/Гц. Допустим, 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 выходе корреляторов в каналах I и Q сформируются следующие уровни

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

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

Здесь уровень на выходе коррелятора состоит из 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 мы считаем, 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

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

$$\sigma_i^2 = \overline{(v_i - \overline{v_i})^2} = \int_0^T \int_0^T \overline{n(t_1) n(t_2)} A\cos(2\pi f t_1) A\cos(2\pi f t_2)\, dt_1\, dt_2 =$$

$$= \frac{A^2 N_0}{2} \int_0^T \cos^2(2\pi f t)\, dt = \frac{A^2 T N_0}{4} = \frac{E_s N_0}{2}$$

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

В данном случае в канале I сформировано так называемое отношение сигнал-шум после обработки

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

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) будет равна произведению плотностей отдельных величин

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

Далее, т. к. здесь удобнее оперировать длинами и углами, то перейдем в полярную систему координат

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

Тогда вероятность правильного приема

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

Далее, подставляя плотность в явном виде, получим

$$P_{\text{прав.}}=\int\limits_{-\pi/8}^{\pi/8}\int\limits_{0}^{\infty}\frac{1}{2\pi\sigma^2}e^{-\frac{1}{2\sigma^2}(r\cos\varphi-E_s)^2}e^{-\frac{1}{2\sigma^2}(r\sin\varphi)^2}r\,dr\,d\varphi \quad .$$

Переходя к переменной $q^2=\frac{2E_s}{N_0}$ 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 рассматривать нет смысла, т. к. они равновероятные и в таком случае сумма из восьми одинаковых вероятностей разделится на 8.

Получившаяся формула не удобна для численных расчетов, поэтому сделаем аппроксимацию. Для этого 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 [Федорюк, Метод перевала]

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

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

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

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