

Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»

На правах рукописи

Кудряшова Анастасия Юрьевна

**Исследование и разработка эффективных методов кодирования источника
при преобразовании метрических пространств и действии помех**

Специальность 05.12.13 –

Системы, сети и устройства телекоммуникаций

ДИССЕРТАЦИЯ

на соискание ученой степени

кандидата технических наук

Научный руководитель:
доктор технических наук,
профессор Аджемов А.С.

Москва, 2022

Оглавление

ВВЕДЕНИЕ	3
1. АНАЛИЗ ОСОБЕННОСТЕЙ КОДИРОВАНИЯ СООБЩЕНИЙ ОТ РАЗЛИЧНЫХ ИСТОЧНИКОВ	13
1.1 Первичное кодирование сообщений источника	13
1.2. Кодирование при сопряжении изохронных и анизохронных сигналов с синхронным цифровым трактом	17
1.3 Кодирование значений отсчетов при аналого-цифровом преобразовании....	26
1.4. Кодирование в сигнально-кодовых конструкциях	33
1.5 Выводы по разделу.....	35
2. РАЗРАБОТКА ОБЩЕГО (УНИВЕРСАЛЬНОГО) МЕТОДА ОЦЕНКИ ЭФФЕКТИВНОСТИ ПЕРВИЧНОГО КОДИРОВАНИЯ.....	37
2.1. Постановка задачи.....	37
2.2. Предельная эффективность преобразования непрерывно-дискретного сигнала в цифровой при равновероятном появлении элементов различной длительности	38
2.3. Предельная эффективность преобразования непрерывно-дискретного сигнала в цифровой при неравновероятном появлении элементов различной длительности	52
2.4 Выводы по разделу.....	60
3. РАЗРАБОТКА МЕТОДА ИССЛЕДОВАНИЯ ВОЗНИКНОВЕНИЯ ДОПОЛНИТЕЛЬНЫХ ИСКАЖЕНИЙ ПРИ НАЛИЧИИ ОШИБОК В ЦИФРОВОМ КАНАЛЕ И ПРЕОБРАЗОВАНИИ РАЗЛИЧНЫХ МЕТРИЧЕСКИХ ПРОСТРАНСТВ.....	62
3.1 Исходные данные и определения	62
3.2. Особенности расстояний в пространстве Хэмминга.....	67
3.3. Особенности биекции пространств при передаче сообщений от источника к получателю	74
3.3.1. Биекция пространств A и B	75
3.3.2. Биекция типа A и B для регулярных структур.....	86
3.4 Выводы по разделу.....	98
4. РАЗРАБОТКА МЕТОДА ИССЛЕДОВАНИЯ ВОЗНИКНОВЕНИЯ ОШИБОК В ЦИФРОВОМ КАНАЛЕ ПРИ НАЛИЧИИ ИСКАЖЕНИЙ И ПРЕОБРАЗОВАНИИ РАЗЛИЧНЫХ МЕТРИЧЕСКИХ ПРОСТРАНСТВ	100

4.1. Постановка задачи и определения.....	100
4.2. Биекция пространств G и Q	101
2.3. Искажения при четырехкратных методах дискретной модуляции	111
2.4. Искажения при многократных методах дискретной модуляции	115
2.5. Выводы по разделу	119
5. РАЗРАБОТКА ПРОГРАММЫ РАСЧЕТА ДОПОЛНИТЕЛЬНЫХ ИСКАЖЕНИЙ ДЛЯ РАЗЛИЧНЫХ МОДЕЛЕЙ ОШИБОК.....	121
5.1. Постановка задачи	121
5.2. Описание программного алгоритма.....	121
5.3. Пользовательский интерфейс, установка ПО.....	127
5.4. Тестовые данные для матрицы 4x4.....	129
5.5. Тестовые данные для матрицы 8x8	130
5.6. Выводы по разделу	136
ЗАКЛЮЧЕНИЕ	137
Список литературы	141
Приложение А. Исходный код.	152
Приложение Б. Акт о внедрении.....	190
Приложение В. Свидетельство о государственной регистрации программы для ЭВМ	191

ВВЕДЕНИЕ

Актуальность темы исследования. Утвержденная Распоряжением Правительства Российской Федерации в июле 2017 года программа «Цифровая экономика Российской Федерации» определила важнейшие сквозные цифровые технологии, среди которых промышленный интернет, технологии беспроводной связи, технологии виртуальной и дополненной реальностей. Намечено значительное расширение использования современных инфокоммуникаций в социальной сфере, в образовании, в медицине и т.д. Все это сопровождается быстрым ростом информационных потоков, что требует постоянного совершенствования телекоммуникационных систем с целью обеспечения максимально возможной скорости передачи, высокой надежности при соблюдении различных требований к качеству передачи сообщений. Важным является также необходимость экономически обоснованной технической реализации различных систем и в частности систем первичного кодирования сообщений от различных источников, их последующую цифровую обработку и передачу с помощью модемов по разнообразным каналам связи.

При преобразовании различных сообщений от источников в цифровой сигнал в них вносится определенная погрешность, связанная с проводимыми дискретизацией и квантованием. Помимо этого, в каналах и линиях связи действуют помехи, вносящие дополнительные искажения, снижающие качество восстанавливаемого на приеме сигнала. Чтобы уменьшить эти искажения необходимо исследовать влияние различных преобразований, которые претерпевает сигнал в процессе его передачи от источника к получателю, что в аналитическом описании соответствует ряду преобразований в различных метрических пространствах.

При преобразовании сигналов в цифровой каждому передаваемому сообщению присваивается число, представленное, как правило, в виде кодовой комбинации двоичных символов 0 и 1 определенной длины. Далее, для повышения

верности приема цифровой последовательности, на передаче в неё вносятся дополнительная избыточность в виде служебных символов, позволяющих обнаруживать и исправлять возникающие из-за помех ошибки. Затем с целью согласования спектральных характеристик цифрового сигнала с частотными характеристиками канала применяют различные методы модуляции, при которых происходит еще одно преобразование сигнала, на этот раз из цифрового - в аналоговый и т.д.

На приеме осуществляются обратные преобразования. Однако происходят они с сигналами, получившими из-за вредного влияния помех определенные искажения, что приводит к ошибкам в восстанавливаемом цифровом сигнале, которые в свою очередь приводят к дополнительным искажениям в реконструируемом исходном сигнале и далее в сообщении, поступающем к получателю. При этом оказывается, что существует нелинейное влияние различных методов преобразования исходного сигнала в процессе его передачи и затем восстановлении на приеме на итоговую оценку качества переданного сигнала. В результате не всегда решения, оптимизирующие преобразования на отдельных этапах, приводят к наилучшим решениям в целом, хотя именно это и является основной целью при формировании системы передачи сообщений от источника к получателю.

Объект исследования: различные сообщения от источника, методы их цифрового отображения, а также методы их последующих преобразований и восстановление исходного сообщения на приемной стороне.

Предмет исследования: методы преобразования сигналов в условиях действия помех.

Степень научной разработанности темы. Начало разработок теории и методов кодирования сообщений от источников и их последующей передачи с помощью методов модуляции можно отнести к первой половине XX века.

Базовыми явились работы Клода Шеннона [1-6], заложившие основы математической теории связи, создавшего теорию информации, позволившей

определить потенциальные границы эффективности телекоммуникационных систем.

Фундаментальными явились также работы Владимира Александровича Котельникова [7-10], доказавшего возможность преобразования непрерывного (аналогового) сигнала в дискретный, заложившего основы теории потенциальной помехоустойчивости.

В статье Д. Хаффмана [1] описан метод построения минимально-избыточных кодов, рассмотрен алгоритм префиксного кодирования с минимальной избыточностью, известный как алгоритм Хаффмана.

В трудах Р. Хэмминга [2,3] описаны основы теории кодирования, рассмотрены коды для коррекции ошибок, в частности, конструкция блочного кода, который корректирует одиночные ошибки, возникающие при передаче сообщений, предложен конструктивный метод построения кодов с избыточностью и простым декодированием.

В трудах Р. Фано [4,5] рассмотрена связь средней потери информации через канал передачи с шумами с вероятностью ошибок при приёме сигнала, приведен способ вычисления нижней границы вероятности ошибки для любого декодера, а также способ получения границ для минимаксного риска в оценке плотности.

В трудах К.Шеннона [6-11] рассмотрены основы теории информации и криптографии, введено понятие энтропии источника, описана ее связь со средней и достижимой степени сжатия с помощью кодирования с потерями. Помимо этого, разобрана связь пропускной способности канала и существования кода, который возможно использовать для передачи с ошибкой, стремящейся к нулю (при увеличении длины блока), а также установлен предел максимального сжатия данных и числовое значение энтропии Шеннона, найдена пропускная способность канала, означающая теоретическую верхнюю границу скорости передачи данных.

Труды В.А. Котельникова [12-15] посвящены проблемам совершенствования методов радиоприёма, изучению радиопомех и разработке методов борьбы с ними. В частности, описана теорема отсчетов, теория потенциальной помехоустойчивости.

В трудах А.Г. Зюко [16-21] рассмотрены вопросы помехоустойчивости и эффективности систем связи, в частности методы повышения эффективности систем передачи дискретных сообщений, предложена оценка эффективности помехоустойчивых кодов.

В трудах Ю.Б. Окунева [22,23] рассматриваются общие принципы цифровой передачи сообщений фазоманипулированными сигналами, излагаются основы теории фазоразностной модуляции, включая вопросы формирования и обработки таких сигналов.

В трудах А.С. Аджемова [24-33] рассмотрены вопросы помехоустойчивости и эффективности передачи дискретных сигналов при их асинхронном сопряжении с синхронным цифровым трактом.

Однако современные исследователи и их предшественники не уделили достаточного внимания проблеме трансформации сигналов из одного метрического пространства в другое, в условиях действия помех в канале связи, что оказывается важным и достаточно существенным с точки зрения обеспечения качества передачи сообщений с учетом нелинейности данных преобразований.

Цель исследования. Целью исследования является разработка методов анализа и расчета, позволяющих оптимизировать типы биекции пространств, отображающих различные этапы передачи сообщений от источника к получателю.

Научная задача и частные задачи исследования.

Научная задача состоит в обосновании методов оценки эффективности кодирования передаваемых сигналов, позволяющей однозначно определять «наилучшие» варианты кодирования, минимизирующие искажения исходного сигнала при наличии ошибок в двоичном сигнале.

Для решения научной задачи и достижения цели диссертационного исследования в работе поставлены и решены 2 частные задачи:

1. Теоретическое обоснование эффективных методов сопоставления (кодирования) сообщений источника, заданных в некотором метрическом пространстве, с двоичными кодовыми комбинациями, определенных в

пространстве Хемминга с учетом матрицы потерь и модели ошибок в дискретном канале связи

2. Разработка программы для поиска эффективных методов кодирования с оценкой по среднему значению вносимых искажений в исходном метрическом пространстве.

Результаты исследования могут использоваться в технических системах передачи информации, а также при исследованиях в других областях науки: генетике, экономике, медицине и др., где в результате цифровой обработки происходит преобразование сообщений из некоторых метрических пространств в пространство Хемминга.

Научная новизна результатов исследования. Научная новизна полученных результатов состоит в следующем:

1. Разработан метод расчета потенциальной границы эффективности преобразования многоуровневого дискретно-непрерывного сигнала в цифровой.
2. Определены теоретически достижимые границы эффективности преобразования многоуровневого дискретно-непрерывного сигнала в цифровой в зависимости от требований к искажениям, вносимым в исходный сигнал при цифровом отображении, а также ограничениям на минимальную и максимальную длительность элементов в исходном сигнале.
3. Разработан метод анализа дополнительных искажений, возникающих из-за ошибок в цифровом сигнале при восстановлении исходного аналогового сигнала.
4. Получены оценки теоретически возможного уменьшения искажений за счет оптимизации типа биекции пространств, в которых отображается сигнал при его передаче от источника к получателю.

5. Предложен метод анализа битовых ошибок при трансформации аналогового сигнала в цифровой с учетом типа биекции соответствующих пространств.

Теоретическая и практическая значимость. Теоретическая значимость определяется недостатком комплексных исследований, учитывающих возможности уменьшения дополнительных искажений в восстанавливаемом на приеме сигнале, возникающих из-за ошибок в цифровом сигнале, а также учете битовых ошибок, появляющихся при восстановлении цифрового сигнала из непрерывного сигнала при различных видах модуляции и типах биекции пространств, отображающих различные этапы преобразований сигналов при их передаче от источника к получателю.

Теоретическая значимость исследования также состоит в следующем:

1. Рассчитаны границы, в рамках которых можно получить уменьшение искажений или битовых ошибок за счет оптимизации типа биекции пространств, отображающих сигналы при их передаче от источника к получателю.
2. Доказана эффективность использования кода Грэя и его аналогов при оценке искажений по среднему значению и преобладающему влиянии однократных ошибок для ряда методов дискретной модуляции. В то же время показано, что для иных методов дискретной модуляции предпочтение следует отдать методу взвешенного кодирования, обеспечивающему минимальную величину максимальных искажений в два раза меньшую по сравнению с кодом Грэя.

Практическая значимость исследования состоит в следующем:

1. Предложены инженерные методы расчета дополнительных искажений или битовых ошибок, позволяющие проводить соответствующие оценки при проектировании устройств кодирования источника и дискретной модуляции.

2. Разработана программа на ЭВМ, позволяющая проводить расчеты дополнительных искажений или битовых ошибок при большом числе вариантов биекции пространств, отображающих сигналы при кодировании источников или дискретной модуляции.

Методы исследования. Для решения поставленных задач в диссертационной работе применялся аналитический метод исследования с использованием теории вероятностей и комбинаторики, теории множеств и математического анализа.

Для проведения расчетов и моделирования процессов, не поддающихся точному аналитическому описанию, применялся метод расчета на ЭВМ.

Для реализации алгоритма использовался высокоуровневый язык C# и программная платформа Microsoft.NET. Для матричных вычислений использовалась специализированная библиотека Matrix Library .Net v2.0 (C) Anas Abidi, 2004. Для реализации интерфейса пользователя использовались стандартные возможности Microsoft Excel, в том числе по созданию встраиваемых (add-in) модулей.

Реализация и внедрение результатов работы.

Полученные в диссертационной работе результаты используются при чтении курсов лекций, проведении практических занятий, выполнении курсовых работ по дисциплинам: «Теория информации и кодирования», «Теоретические основы инфокоммуникаций» и «Теория информации» на кафедре «Общая теория связи» МТУСИ, что подтверждается соответствующим актом. Получено свидетельство о государственной регистрации программы для ЭВМ.

Апробация работы. Все результаты диссертации научно обоснованы и опубликованы в ведущих научных изданиях. Результаты диссертации докладывались на следующих конференциях:

Международная научно-техническая конференция «Фундаментальные проблемы радиоэлектронного приборостроения» (Intermatic - 2017), 21 - 23 ноября 2017, Москва; Международная научно-техническая конференция

«Фундаментальные проблемы радиоэлектронного приборостроения» (Intermatic – 2018), 19 - 23 ноября 2018, Москва; XII Международная отраслевая научно-техническая конференция «Технологии информационного общества» (ТИО-2018), 14-15 марта 2018, Москва; XIII Международная отраслевая научно-техническая конференция «Технологии информационного общества» (ТИО-2019), 20-21 марта 2018, Москва; International Scientific Conference IEEE «Systems of signals generating and processing in the field of on board communications» (ONBOARD), March 14-15, Moscow, Russia, 2018; International Scientific Conference IEEE «Wave Electronics and its Application in Information and Telecommunication Systems» (WECONF), November 26-30, S.-Petersburg, Russia, 2018; International Scientific Conference IEEE «Systems of Signal Synchronization, Generating and Processing in Telecommunications» (SYNCHROINFO), July 4-5, Minsk, Belarus, 2018; International Scientific Conference IEEE «Systems of signals generating and processing in the field of on board communications» (ONBOARD), March 20-21, Moscow, Russia, 2019; XIII Международная научно-техническая конференция «Технологии информационного общества». Москва, Московский технический университет связи и информатики (МТУСИ) 20-21 марта 2019, Москва; International Scientific Conference IEEE «Wave Electronics and its Application in Information and Telecommunication Systems» (WECONF), June 3-7, S.-Petersburg, Russia, 2019; International Scientific Conference IEEE «Systems of Signal Synchronization, Generating and Processing in Telecommunications» (SYNCHROINFO), July 1-3, Yaroslavl, Russia, 2019; 29-я Международная Крымская конференция «СВЧ-техника и телекоммуникационные технологии», 5 - 11 сентября 2019 г., г. Севастополь (КрыМиКО – 2019); International Scientific Conference IEEE «Systems of Signals Generating and Processing in the Field of on Board Communications» (ONBOARD - 2020), March 19-20, Moscow, Russia, 2020; International Scientific Conference IEEE «Systems of Signal Synchronization, Generating and Processing in Telecommunications» (SYNCHROINFO), July 1-3, Svetlogorsk, Russia, 2020; 30-я Международная Крымская конференция «СВЧ-техника и телекоммуникационные технологии», 6 - 12 сентября 2020, Севастополь (КрыМиКО – 2020);

Международная научно-техническая конференция IEEE «Systems of Signals Generating and Processing in the Field of on Board Communications» (ONBOARD - 2021), March 15-16, Moscow, Russia, 2021; International Scientific Conference IEEE «Wave Electronics and its Application in Information and Telecommunication Systems» (WECONF), May 31 –June 4, S.-Petersburg, Russia, 2021; International Scientific Conference IEEE «Systems of Signal Synchronization, Generating and Processing in Telecommunications» (SYNCHROINFO), June 30-July 2, Svetlogorsk, Russia, 2021.

Публикации. Результаты диссертационной работы были опубликованы в 24 работах: 5 публикаций в журналах Перечня ВАК, 11 публикаций в сборниках конференций, индексируемых в Scopus и WoS, 8 публикаций в журналах и сборниках конференций, индексируемых в РИНЦ.

Личный вклад. Результаты диссертационной работы получены автором самостоятельно, математические процедуры и программные средства разработаны при непосредственном участии автора.

Соответствие паспорту специальности.

Проведенное исследование соответствует п. 2 «Исследование процессов генерации, представления, передачи, хранения и отображения аналоговой, цифровой, видео-, аудио- и мультимедиа информации; разработка рекомендаций по совершенствованию и созданию новых соответствующих алгоритмов и процедур» и п. 11 «Разработка научно-технических основ технологии создания сетей, систем и устройств телекоммуникаций и обеспечения их эффективного функционирования» паспорта специальности 05.12.13 – «Системы, сети и устройства телекоммуникаций».

Положения, выносимые на защиту:

1. Оптимизация типа биекции сигналов при преобразовании между пространствами Хэмминга и Евклида, в которых описываются сигналы, позволяет в два и более раза уменьшить искажения, возникающие в процессе преобразования.
2. Предельная эффективность преобразования различных сигналов в цифровой может быть оценена путем построения предложенной комбинаторной

модели учета всех возможных реализаций исходного сигнала с последующей аппроксимацией их численности максимальным членом при бесконечной длительности обрабатываемого сигнала.

3. Практически достижимая эффективность преобразования непрерывно-дискретного сигнала в цифровой при ограниченной длительности отличается от предельно достижимой на 5-10%.

4. Разработан метод оптимизации типа биекции между пространствами Хэмминга и Евклида, позволяющий минимизировать количественную меру ошибок, возникающих при преобразовании сигналов, при заданном виде помех и методе дискретной модуляции.

Структура и объем работы. Диссертация состоит из введения, 5 разделов, заключения, списка литературы, и 3 приложений. Основные результаты изложены на 140 страницах. Диссертация содержит 45 рисунков и 42 таблицы. Дополнительные сведения изложены на 51 странице в приложениях. В библиографию включено 115 источников на русском и английском языках.

1. АНАЛИЗ ОСОБЕННОСТЕЙ КОДИРОВАНИЯ СООБЩЕНИЙ ОТ РАЗЛИЧНЫХ ИСТОЧНИКОВ

1.1 Первичное кодирование сообщений источника

Теоретические основы кодирования сообщений источника были заложены в фундаментальных работах К. Шеннона [6-11]. Известная первая теорема К. Шеннона для канала, где отсутствуют помехи, гласит, что при любой статистике сообщений, поступающих от источника, должен существовать код, который в условиях отсутствия ограничений на задержку сообщений, обеспечит среднюю длину кодового слова $\bar{n} = M\{n_l\}$, отображающего каждое сообщение, сколь угодно близкое к минимально возможному значению

$$n_{\min} = \frac{H(A)}{\log_b m}, \quad (1.1)$$

где $H(A) < \infty$ - энтропия источника, m - основание кода, а b - выбранная логарифмическая мера количества информации.

Для непрерывных источников в формуле (1.1) вместо значения энтропии $H(A)$ следует использовать значение ε - энтропии, введенное академиком А.Н. Колмогоровым [34].

При использовании двоичного кодирования, когда $m = 2$ и $b = 2$, формула (1.1) еще более упрощается и означает, что минимальная длина кодовой комбинации равна энтропии источника, измеряемой в битах.

Формула (1.1) показывает потенциальную теоретически достижимую границу. При этом, к сожалению, «механизм» достижения данного значения не раскрывается. На практике, стремясь достичь этого минимального значения, применяют различные способы, например, методы укрупнения алфавита, неравномерное кодирование и т.д. [1,5,35]. При этом образуемый дискретный канал как бы «настраивается» под соответствующие статистические особенности

сообщений от источника. Если же статистическое описание источника изменится, то проведенная «настройка» может привести к худшим результатам по обеспечению эффективного кодирования источника. В этом смысле образуемый дискретный канал является «непрозрачным», т.к. «настроен» только на определенные статистические особенности сообщений от источника. Более того в ряде случаев такой канал может быть вообще не пригоден для передачи сообщений, отличающихся по названным выше характеристикам от тех, под которые «настраивался» дискретный канал.

Исходя из этого во многих случаях, когда не известно статистическое описание источника, либо источник является не стационарным, используют равномерное блочное кодирование сообщений от источника в виде кодовых комбинаций одинаковой длины, определяемой по очевидному соотношению:

$$n = \lceil \log_2 N \rceil \quad (1.2)$$

где N – количество возможных сообщений, вырабатываемых источником, а знак $\lceil \cdot \rceil$ означает взятие целого с округлением в большую сторону.

Получаемый в этом случае канал называют «прозрачным», т.к. в этом случае нет зависимости эффективности первичного кодирования от статистических особенностей сообщений от источника.

Добиваясь минимальной скорости кодирования источника, совпадающей при двоичном кодировании с его энтропией, помимо этой характеристики важно оценивать сложность реализации и устойчивость к помехам.

Указанное выше блочное кодирование часто оказывается существенно проще в практической реализации, а также позволяет обеспечить большую помехоустойчивость при условии действия помех в канале связи, приводящих к ошибкам в дискретном канале. В [9] была предложена известная блок-схема системы передачи сообщений от источника, представленная на рисунке 1.1.



Рисунок 1.1 - Блок-схема системы передачи сообщений от источника

Согласно этой блок-схеме, сообщение от источника $a(t)$ преобразуется на передающей стороне в сигнал $s(t)$, который затем через канал связи поступает к приемнику. При этом в канале связи действуют помехи $\xi(t)$, в результате действия которых сигнал $s'(t)$, поступающий на приемник, может отличаться от переданного $s(t)$. Из-за этих отличий восстанавливаемое сообщение $a'(t)$, также может отличаться от переданного от источника $a(t)$.

Для более детального анализа процессов, происходящих при преобразовании сообщений в сигнал, и последующую их передачу по каналу связи, приведем более подробную блок-схему, поясняющую и детализирующую различные этапы преобразований. Блок-схема представлена на рисунке 1.2.

Сообщение $a(t)$ от источника преобразуется в цифровой сигнал в «Кодере источника» $s_q(t)$. Затем этот сигнал для повышения помехоустойчивости в блоке «Кодер канала» с добавлением определенного числа двоичных символов преобразуется в цифровой сигнал $s_{q,n}(t)$, который является модулирующим сигналом для последующего блока «Передатчик». В этом блоке обеспечивается согласование спектральных характеристик сигнала $s(t)$ с частотными характеристиками канала. При этом данный блок часто называют «Модулятор». На его выходе формируется сигнал $s(t)$, передаваемый по каналу связи. В результате действия помех $\xi(t)$ сигнал $s'(t)$ на входе «Приемника» («Демодулятора») может

содержать искажения, которые затем после демодуляции могут трансформироваться в ошибки в сигнале $s'_{u,n}(t)$.

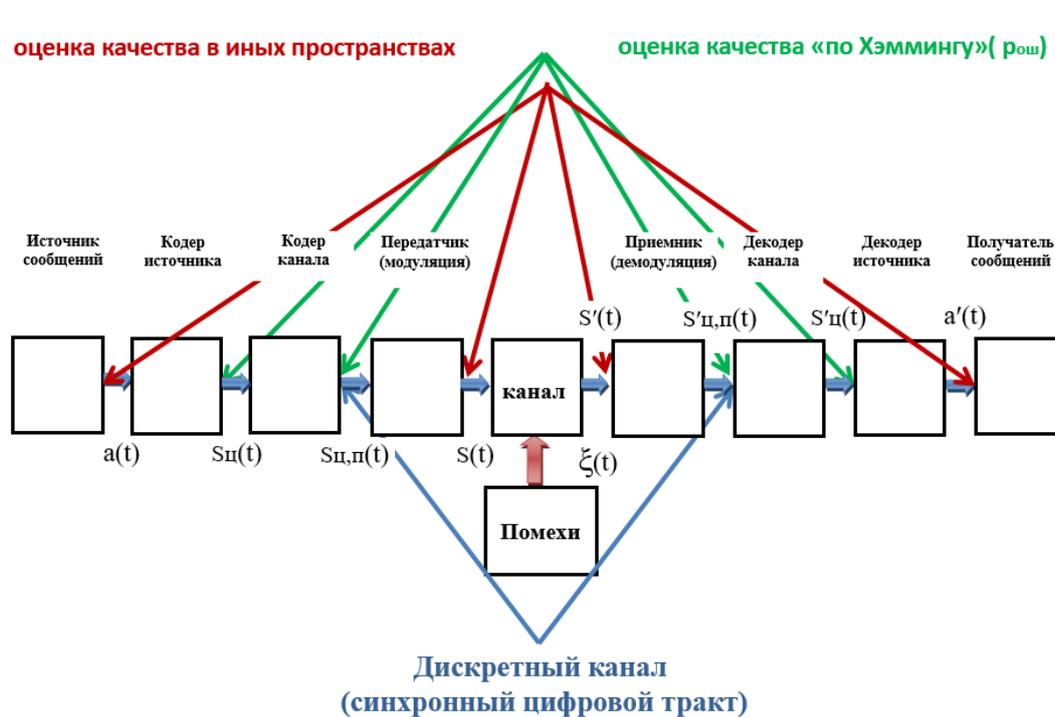


Рисунок 1.2 - Блок-схема системы передачи сообщений от источника

В блоке «Декодер канала» происходит обнаружение и даже исправление возникших ошибок согласно выбранному способу помехоустойчивого кодирования. Однако и после этого в цифровом сигнале $s'_u(t)$, поступающем на «Декодер источника», возможно наличие необнаруженных ошибок, которые затем после восстановления сообщения приведут к дополнительным искажениям в $a'(t)$.

Оценка качества передачи сигналов осуществляется в соответствующих метрических пространствах, в которых эти сигналы отображались. Цифровые сигналы $s_u(t)$, $s'_u(t)$, $s_{u,n}(t)$ и $s'_{u,n}(t)$ представляются в пространстве Хэмминга, тогда как сообщения от источника $a(t)$ и $a'(t)$, а также сигналы $s(t)$ и $s'(t)$ могут отображаться в иных пространствах, часто в пространстве Евклида.

Осуществляя эффективное первичное кодирование в блоке «Кодер источника», и, добиваясь, по возможности, наименьшего значения средней длины кодовой комбинации, следует каждому сообщению $a(t)$ поставить в соответствие

кодovou комбинацию $s_y(t)$. При этом то, как будет образовано это соответствие, не влияет на эффективность первичного кодирования, поскольку средняя длина кодовой комбинации не изменяется. Однако, как показали исследования данной проблемы, при условии, что в канале связи действуют помехи, и в результате этого в цифровом сигнале возникают ошибки, способ сопоставления кодовых комбинаций сигнала $s_y(t)$ с сообщениями $a(t)$, играет заметную роль, поскольку в зависимости от этого меняется величина искажений, вносимых в сообщение $a'(t)$ из-за ошибок в цифровом сигнале $s'_y(t)$ [24].

Использование корректирующих кодов позволяет уменьшить вероятность ошибок, возникающих из-за действующих помех и улучшить качественные характеристики дискретного канала, оцениваемых по вероятности ошибки, что характерно для отображения цифрового сигнала в пространстве Хэмминга. Однако сообщения $a(t)$ и $a'(t)$, а также сигналы $s(t)$ и $s'(t)$ описываются в иных пространствах. А это означает, что какая-либо оптимизация в пространстве Хэмминга в отношении цифровых сигналов не означает оптимизации в целом для, например, сообщений $a(t)$ и $a'(t)$ или сигналов $s(t)$ и $s'(t)$, поскольку преобразование одних пространств в пространство Хэмминга и обратно является нелинейной операцией.

Влияние ошибок в пространстве Хэмминга на искажения сигналов, отображаемых в иных пространствах и, наоборот, в общей постановке является задачей, не решенной до настоящего времени. Однако имеются отдельные результаты для конкретных видов сообщений и определенных размеров кодовых комбинаций двоичного кода, применяемых для формирования цифрового сигнала.

Проведем анализ имеющихся решений.

1.2. Кодирование при сопряжении изохронных и анизохронных сигналов с синхронным цифровым трактом

С началом распространения цифровых систем передачи возникла необходимость сопряжения (согласования) различных цифровых сигналов

(потоков), порождаемых в соответствии с частотой задающих генераторов отдельных систем передачи и конечных источников дискретных сигналов [36,37]. При этом, как правило, эти задающие генераторы, работающие, в том числе, с номинально равной частотой, не были синхронизированы между собой. В результате этого в силу реально имеющейся нестабильности частот задающих генераторов, возникали нарастающие фазовые расхождения, которые надо было компенсировать.

Для сопряжения изохронных дискретных сигналов с синхронным цифровым трактом наибольшее распространение получили методы стаффинга: «+ стаффинг» и «- стаффинг», получивших название методов одностороннего стаффинга, и «±стаффинг», называемый методом двустороннего стаффинга [36,37].

При этих методах двоичным элементам исходного изохронного сигнала от источника $a_i \in A$, где A - множество исходных элементов, состоящее из двух значений 0 и 1, а a_i - соответственно значение текущего элемента в i - момент времени, ставится в соответствие элементы синхронного цифрового тракта $b_i \in B$, где B - множество элементов синхронного цифрового тракта, состоящее также из двух элементов 0 и 1.

При методах стаффинга асинхронное сопряжение (кодирование) строится на основе формирования циклов определенной длины, начало которых определяется специальной синхрокомбинацией (СК) известного вида. В цикле также предусмотрены места для информационных символов $b_i = a_i$, места для команд управления (КУ) и при необходимости места или место для дополнительного элемента (ДЭ) $b_{i+1} = a_{i+1}$. Изобразим это на диаграммах для соответственно

«+стаффинга», представленного на рисунке 1.3 (а, б, с, d) и «-стаффинга», представленного на рисунке 1.4 (а, б, с, d).

При «+стаффинге» за время цикла T от источника поступает с учетом возможной нестабильности частот задающих генераторов либо n элементов (a_1, a_2, \dots, a_n) , либо $(n+1)$ элемент $(a_1, a_2, \dots, a_{n+1})$.

Согласно этому в дискретном канале в цикле эти элементы передаются значениями соответственно n элементов (b_1, b_2, \dots, b_n) , как показано на рисунке 1.3 а, б.

В случае, когда за время цикла от источника поступит $(n+1)$ элемент $(a_1, a_2, \dots, a_{n+1})$, то последний элемент a_{n+1} передается на месте ДЭ, как это показано на рисунке 1.3 с, d.

Поскольку как элементы a_i , так и элементы b_i принимают значение 0 или 1, то, как уже отмечалось, с помощью команд управления (КУ) передаются сведения о том, как используется ДЭ. Передается там значение элемента a_{n+1} или ДЭ «свободен» от a_{n+1} , и на приеме информация из этого элемента не считывается при восстановлении исходного сигнала [28].

При «-стаффинге» значения частот задающих генераторов выбирается так, чтобы за интервал цикла T от источника поступало с учетом возможной нестабильности частот задающих генераторов либо n элементов (a_1, a_2, \dots, a_n) , что представлено на рисунке 1.4 (а, б) либо $(n-1)$ элемент $(a_1, a_2, \dots, a_{n-1})$, что представлено на рисунке 1.4 (с, d).

В этом случае ДЭ в цикле не нужен, а с помощью команд управления КУ передаются сведения об элементе b_n , который в определенном смысле играет роль ДЭ. Соответственно на приеме в зависимости от КУ из элемента b_n либо считывается значение a_n , представленное на рисунке 1.4 (а, б), либо ничего не считывается, поскольку при сопряжении на месте элемента b_n не размещалось значение a_n , как представлено на рисунке 1.4 (с, d).

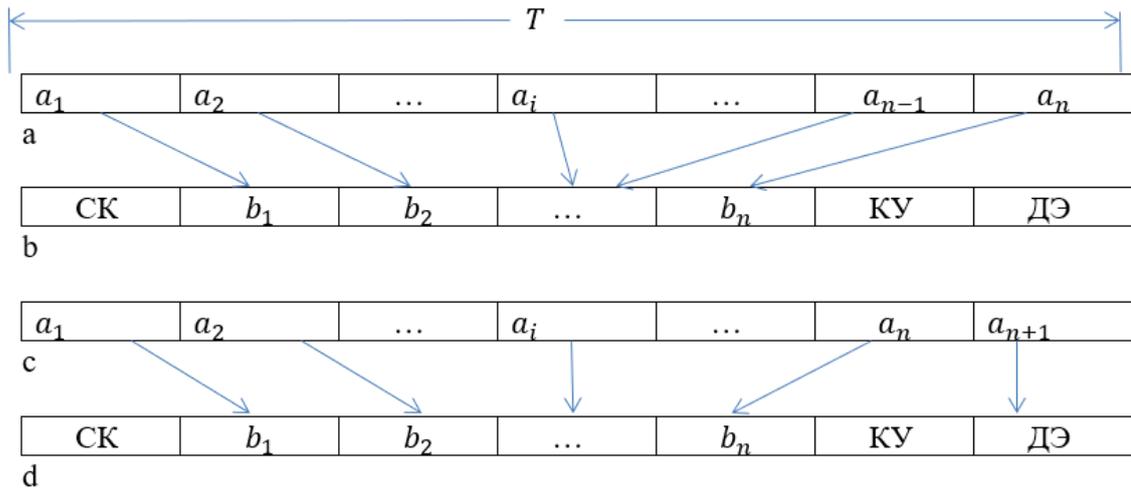


Рисунок 1.3 - Временные диаграммы «+стаффинга»

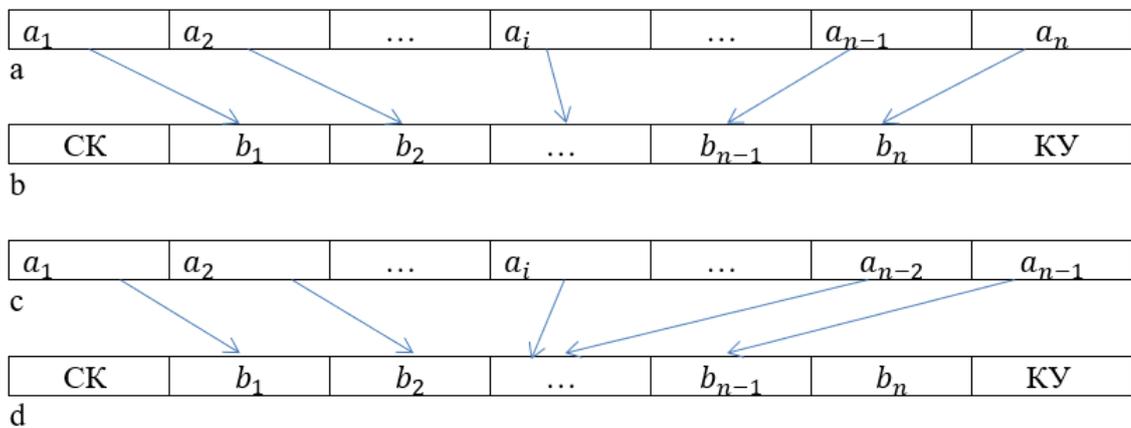


Рисунок.1.4 - Временные диаграммы «-стаффинга»

На практике распространение получил метод двустороннего стаффинга, предусматривающего работу как по принципу «+стаффинга», так и «-стаффинга».

Очевидно, что с точки зрения обеспечения требуемой помехоустойчивости наибольшее внимание следует уделить защите команд управления и СК, поскольку их неверный прием приведет к появлению вставки лишнего элемента, либо, наоборот, к изъятию истинного информационного элемента.

Ошибки же в элементах b_i без изменений будут отражаться в виде ошибок среди элементов a_i . При этом качество передачи и приема элементов a_i и b_i оценивается в пространстве Хэмминга, как правило, через соответствующие распределения вероятностей правильного или ошибочного приема.

При сопряжении изохронных сигналов с синхронным цифровым трактом образуется «закрытый» дискретный канал, настраиваемый на соответствующий вид дискретной последовательности от источника (a_1, a_2, \dots, a_n) и дискретной последовательности (b_1, b_2, \dots, b_n) после сопряжения. Эти последовательности описываются в пространстве Хэмминга и основной решаемой проблемой при сопряжении является сглаживание скоростей дискретных сигналов и минимизация фазового дрожания, называемого джиттером [38].

Анизохронный сигнал в отличие от рассмотренного изохронного может иметь длительность элементов любую, но не меньше некоторого минимального значения (τ_{\min}) , т.е. это, согласно принятой классификации [35] дискретно-непрерывный сигнал, который следует преобразовать в дискретный сигнал.

Ряд методов асинхронного сопряжения анизохронных сигналов описаны в [28,29]. Среди них наиболее распространение получил метод «Скользящего индекса с подтверждением» (СИП), обладающий наиболее высокой эффективностью преобразования (кодирования) дискретно-непрерывного сигнала посредством дискретного сигнала. Причем кодирования с определенным критерием качества, поскольку исходная последовательность элементов (a_1, a_2, \dots, a_n) имеет непрерывные значения длительностей, тогда как их отображение в виде последовательности (b_1, b_2, \dots, b_n) содержит элементы определенной длительности, отличающиеся от длительности исходных элементов на величину не более Δ .

В большинстве случаев критерием качества преобразования элементов дискретно-непрерывного сигнала в дискретный принимается максимальная абсолютная погрешность изменения длительности элемента исходного сигнала Δ , либо максимальная величина относительной погрешности, вычисляемая по формуле:

$$\delta_{\max} = \frac{\Delta}{\tau_{\min}} \cdot 100\% \quad (1.3)$$

Согласно введенной оценки качества преобразования непрерывно-дискретного сигнала в дискретный, минимальная скорость кодирования (R_{\min})

будет определяться эpsilon-энтропией источника $H(\varepsilon \leq \delta_{\max})$, когда относительная погрешность отображения элементов анизохронного сигнала не превысит значения δ_{\max} .

Как уже отмечалось, наилучшим методом сопряжения анизохронного сигнала с дискретным каналом был метод «скользящего индекса с подтверждением», при котором скорость кодирования была близка к теоретическому пределу. Однако оказалось, что этому методу свойственен процесс размножения ошибок, возникающий из-за ошибок в дискретном канале. При этом интенсивность этого процесса определялась также и методом кодирования (сопоставления) элементов исходного непрерывно-дискретного (анизохронного) сигнала посредством символов дискретного сигнала, передаваемых по дискретному каналу.

Рассмотрим примеры этих процессов на диаграммах, представленных на рисунке 1.5 a, b, c, d, e, f, g, h.

При методе «скользящего индекса с подтверждением» каждый фронт в непрерывно-дискретном сигнале (фронтом будем называть любой переход в анизохронном сигнале из состояния 1 в состояние 0 и наоборот) отображается кодовой комбинацией дискретного сигнала, состоящей из так называемого стартового символа A , указывающего на значение элемента исходного сигнала, уточняющих символов B_1, B_2, \dots, B_k и символов подтверждения C значение которых совпадает со значением предыдущего стартового символа и, следовательно, со значением передаваемого элемента исходного дискретно-непрерывного сигнала.

Рассмотрим подробнее передачу элементов непрерывно-дискретного сигнала по дискретному каналу на примере, показанном на рисунке 1.5, когда число уточняющих символов равно двум.

Стартовый символ A служит для обозначения факта появления фронта во входном сигнале, показанном на диаграмме (a). На диаграмме (b) приведена тактовая последовательность символов дискретного канала, значения которых меняются согласно методу «скользящего индекса с подтверждением».

Вслед за стартовым символом A следуют два уточняющих символа B_1, B_2 , с помощью которых кодируется часть тактового интервала, в который попадает фронт элемента анизохронного сигнала.

Поскольку в данном примере два уточняющих символа, тактовый интервал можно разбить на 4 зоны, пронумерованные как 1, 2, 3, 4, присвоив каждой соответствующее значение кодовых символов.

Причем присвоение значений уточняющих символов может быть различным для случая, когда стартовый символ A равен 1, или когда стартовый символ A равен 0.

Число различных вариантов способов задания значений уточняющих символов, кодирующих соответствующие зоны, очевидно, равно:

$$K = (2^k)! \quad (1.4)$$

Но с учетом того, что кодирование уточняющих символов при стартовом символе равном 1, не зависит от способа кодирования при стартовом символе, равном 0, формула (1.4) принимает вид:

$$K = [(2^k)!]^2 \quad (1.5)$$

Применив эту формулу для рассматриваемого примера, получаем, что число различных вариантов кодирования при двух уточняющих символах равно $K = 576$.

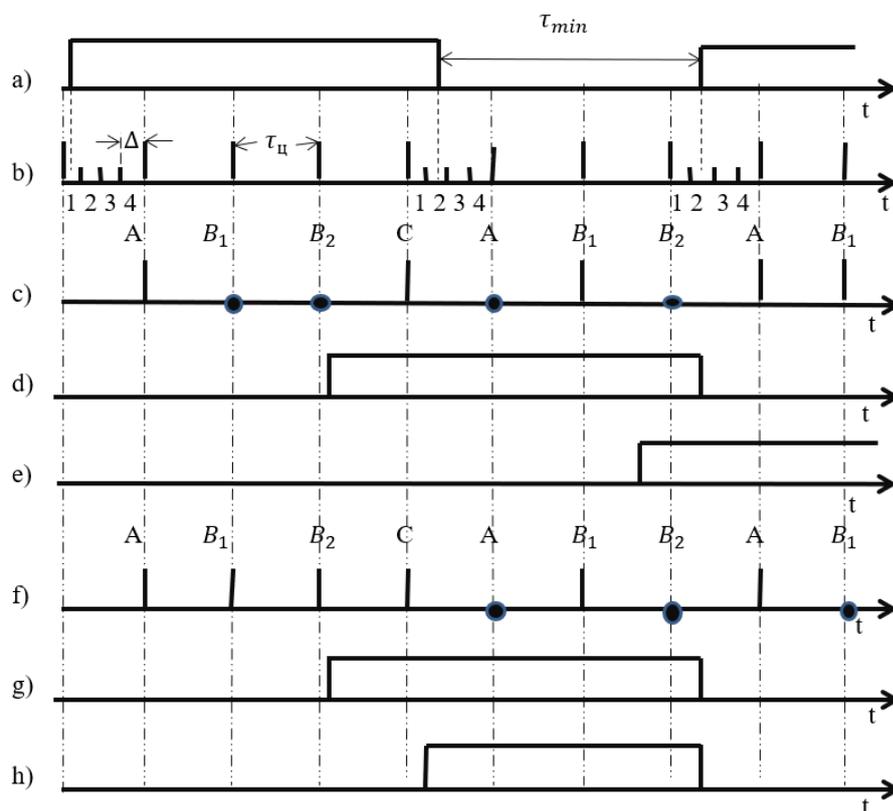


Рисунок 1.5 - Метод «скользящего индекса с подтверждением»

Таблица 1.1-Варианты кодирования

	Вариант 1			Вариант 2	
	$A = 1$	$A = 0$		$A = 1$	$A = 0$
Зона 1 - B_1, B_2	0,0	0,0	Зона 1 - B_1, B_2	1,1	0,0
Зона 2 - B_1, B_2	1,0	1,0	Зона 2 - B_1, B_2	0 1	1,0
Зона 3 - B_1, B_2	0,1	0,1	Зона 3 - B_1, B_2	1,0	0,1
Зона 4 - B_1, B_2	1,1	1,1	Зона 4 - B_1, B_2	0,0	1,1

Выберем для примера два варианта, приведенные в таблице 1.1

Отметим, что число уточняющих символов определяет точность, с которой можно закодировать местоположение фронта между символами тактового

интервала τ_u . В рассматриваемом примере абсолютная погрешность не превышает величины

$$\Delta = \frac{\tau_u}{2^k} = \frac{\tau_u}{4} \quad (1.6)$$

А каждый фронт элемента исходного сигнала, таким образом, отображается тремя символами кодовой комбинации, передаваемой по дискретному каналу.

Если после кодовой комбинации A, B_1, B_2 в исходном сигнале не появляется фронта, то формируется подтверждающий символ C , совпадающий по знаку с предыдущим символом A . В дальнейшем при появлении фронта вновь будет сформирована комбинация A, B_1, B_2 . Если фронта нет, то продолжается формирование подтверждающих символов C .

На диаграммах (с) и (f) соответственно показаны закодированные последовательности по варианту 1 и варианту 2 из таблицы 1.1.

На диаграммах (d) и (g) изображены восстановленные исходные сигналы, переданные методом «скользящего индекса с подтверждением» при использовании различных вариантов кодирования зон уточняющими символами. Сравнение показывает их полное совпадение.

Однако, если из-за действия помех произойдет ошибка, то ее последствия могут оказаться совершенно разными. Продемонстрируем это на примере, когда из-за помех произойдет ошибка в первом стартовом символе A . Для варианта 1 кодовая последовательность $A, B_1, B_2, C, A, B_1, B_2, A, B_1, \dots$ будет на приеме восприниматься, как $C, C, C, A, B_1, B_2, A, B_1, B_2, \dots$.

Сопоставляя эти последовательности, можно отметить начавшийся процесс размножения ошибок, когда из-за одной ошибки в дискретном канале будет неверно восстановлено несколько элементов исходного сигнала. Примером этого служит сравнение диаграмм (d) и (e).

Для варианта 2 та же кодовая последовательность $A, B_1, B_2, C, A, B_1, B_2, A, B_1, \dots$ будет на приеме восприниматься, как $C, A, B_1, B_2, A, B_1, B_2, A, B_1$. Это приведет к искажению одного элемента исходного сигнала, но размножения ошибок не будет. Свидетельством этого является сравнение диаграмм (g) и (h).

В [24,25] проведены исследования помехоустойчивости метода «скользящего индекса с подтверждением» для всех возможных способов кодирования уточняющих символов при $k=1$ и $k=2$, что соответствовало случаям практического применения данного метода.

Исследования показали, что рациональный выбор позволяет в 2-3 раза повысить помехоустойчивость в зависимости от действующих помех, соотношений $\frac{\tau_u}{\tau_{\min}}$ и способов кодирования уточняющих символов. При этом, как отмечалось эффективность преобразования непрерывно-дискретного (анизохронного) сигнала в дискретный сигнал не меняется.

1.3 Кодирование значений отсчетов при аналого-цифровом преобразовании

Появление интегральных микросхем, определивших перспективу цифровых методов передачи сигналов, привели к появлению систем, преобразующих, в том числе и аналоговые (непрерывные) сигналы в цифровую форму.

В основе такого преобразования находилась известная теорема В.А. Котельникова, определившая интервал дискретизации с последующим квантованием значений отсчетов [12-15].

Не уменьшая общности рассуждений, покажем на примере, изображенном на рисунке 1.6, особенности и эффекты от различных вариантов кодирования значений отсчетов в случае, когда число уровней квантования равно $N=8$.

Среди всех возможных вариантов трехсимвольного кодирования с целью иллюстрации разнообразия возникающих искажений были рассмотрены натуральный код, код Грея, код в ИКМ, случайный код, код с минимальными

искажениями при двукратных ошибках (Ми-2) и код с минимальными искажениями (Ми-3) при трехкратных ошибках.

Коды и величина искажений каждого символа при однократных, двукратных и трехкратных ошибках представлены в таблицах 1.2-1.7.

Таблица 1.2 – Искажения при Натуральном коде

Натур. код (1)	000	001	010	011	100	101	110	111	Итого
3, 2, 1	4, 2, 1	4, 2, 1	4, 2, 1	4, 2, 1	4, 2, 1	4, 2, 1	4, 2, 1	4, 2, 1	56
3-2, 2-1, 3-1	6, 3, 5	6, 1, 3	2, 1, 5	2, 3, 3	2, 3, 3	2, 1, 5	6, 1, 3	6, 3, 5	80
321	7	5	3	1	1	3	5	7	32

Таблица 1.3 – Искажения при коде Грея

Код Грея (2)	000	001	011	010	110	111	101	100	Итого
3, 2, 1	7, 3, 1	5, 1, 1	3, 1, 1	1, 3, 1	1, 3, 1	3, 1, 1,	5, 1, 1,	7, 3, 1	56
3-2, 2-1, 3-1	4, 2, 6	4, 2, 6	4, 2, 2	4, 2, 2	4, 2, 2	4, 2, 2,	4, 2, 6	4, 2, 6	80
321	5	3	5	3	3	5	3	5	32

Таблица 1.4 – Искажения при коде в ИКМ

Код в ИКМ (3)	011	010	001	000	100	101	110	111	Итого
3, 2, 1	7, 2, 1	5, 2, 1	3, 2, 1	1, 2, 1	1, 2, 1	3, 2, 1	5, 2, 1	7, 2, 1	56
3-2, 2-1, 3-1	5, 3, 6	3, 1, 6	5, 1, 2	3, 3, 2	3, 3, 2	5, 1, 2	3, 1, 6	5, 3, 6	80
321	4	4	4	4	4	4	4	4	32

Таблица 1.5 – Искажения при Случайном коде

Случ. код (4)	000	011	111	001	100	010	101	110	Итого
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-------

3, 2, 1	4, 5, 3	1, 2, 4	1, 4, 5	3, 2, 3	4, 3, 2	2, 5, 4	3, 4, 2	2, 3, 5	76
3-2, 2-1, 3-1	7, 1, 6	5, 1, 6	1, 2, 3	1, 2, 1	1, 2, 1	1, 2, 3	5, 1, 6	7, 1, 6	72
321	2	3	2	4	3	1	1	4	20

Таблица 1.6 – Искажения при коде Ми-3

Код Ми-3 (4)	000	111	001	110	010	101	011	100	Итого
3, 2, 1	7, 4, 2	5, 4, 2	3, 4, 2	1, 4, 2	1, 4, 2	3, 4, 2	5, 4, 2	7, 4, 2	80
3-2, 2-1, 3-1	3, 6, 5	1, 6, 3	1, 2, 5	3, 2, 3	3, 2, 3	1, 2, 5	1, 6, 3	3, 6, 5	80
321	1	1	1	1	1	1	1	1	8

Таблица 1.7 – Искажения при коде Ми-2

Код Ми-3 (4)	000	110	011	101	111	001	100	010	Итого
3, 2, 1	6, 7, 5	6, 5, 3	2, 3, 5	2, 1, 3	2, 1, 3	2, 3, 5	6, 5, 3	6, 7, 5	96
3-2, 2-1, 3-1	1, 2, 3	1, 2, 1	1, 2, 1	1, 2, 3	1, 2, 3	1, 2, 1	1, 2, 1	1, 2, 3	40
321	4	4	4	4	4	4	4	4	32

На рисунке 1.6.1 наглядно показано общее количество искажений из-за однократных, двукратных и трехкратных ошибок по каждому варианту кодирования. Количество искажений обозначено как k .

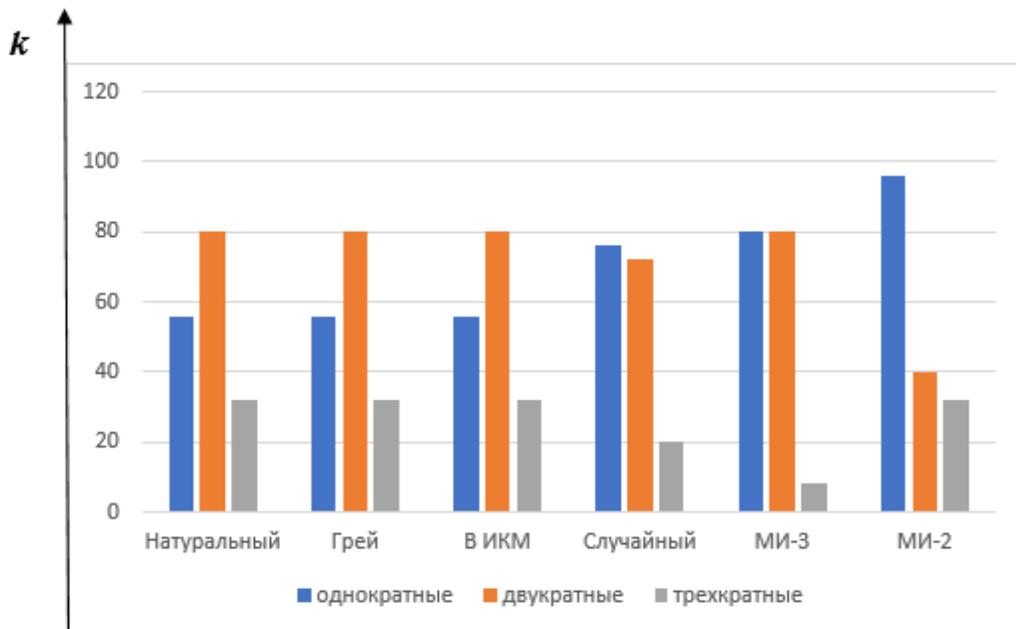


Рисунок 1.6 – Количество общих искажений из-за однократных, двукратных и трехкратных ошибок при различных вариантах кодирования

На рисунках 1.6.1, 1.6.2, 1.6.3 наглядно показано количество возникающих искажений из-за однократных, двукратных и трехкратных ошибок соответственно. Рассмотрены все 6 вариантов кодирования. Величина искажений обозначена как k .

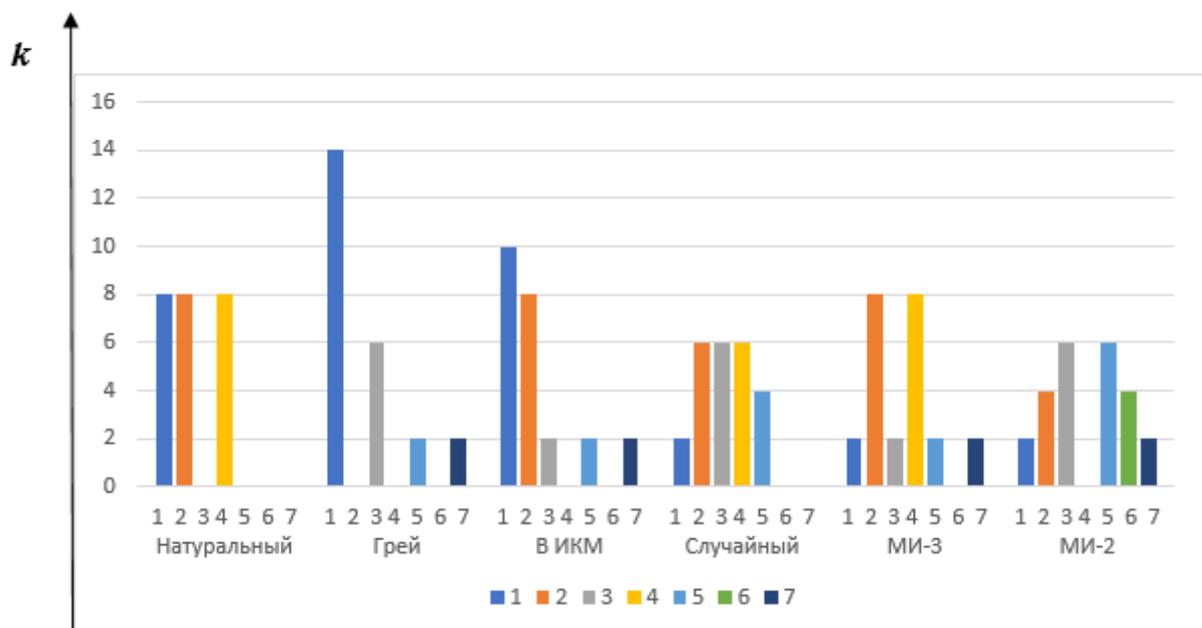


Рисунок 1.6.1 – Количество искажений каждого вида из-за однократных ошибок при различных вариантах кодирования

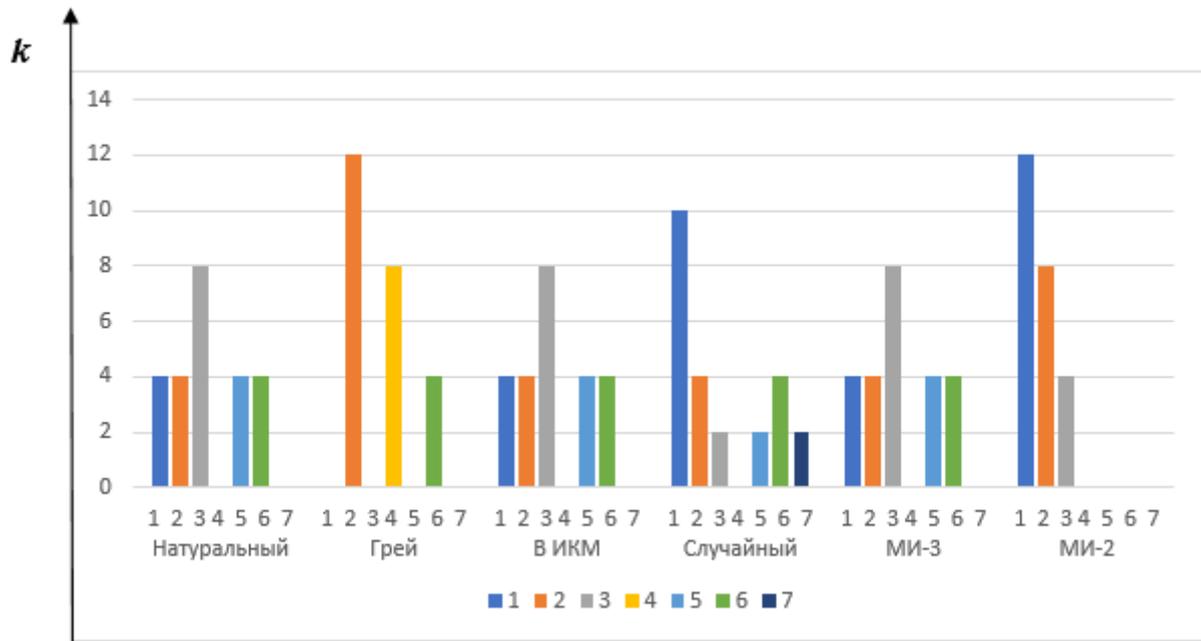


Рисунок 1.6.2 – Количество искажений каждого вида из-за двукратных ошибок при различных вариантах кодирования

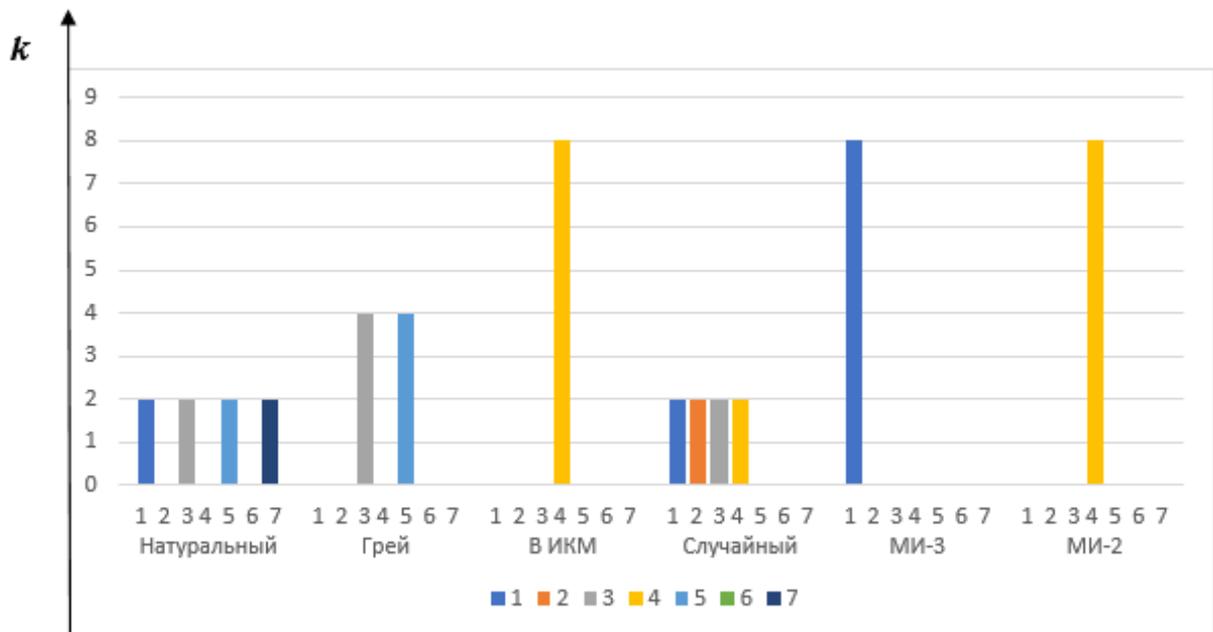


Рисунок 1.6.3 – Количество искажений каждого вида из-за трехкратных ошибок при различных вариантах кодирования

Анализ диаграмм подтверждает, что выбор варианта кодирования существенно влияет на качество восстановления исходного аналогового сигнала.

Для наглядности сведем итоговые цифры по рискам в таблице 1.8.

Таблица 1.8 – Сводные данные по искажениям

Код	1	2	3	4	5	6
1-кр	56	56	56	76	80	96
2-кр	80	80	80	72	80	40
3-кр	32	32	32	20	8	32

Так, например, для однократных ошибок преимущество за кодом Грея по сравнению с Мин-2 и Мин-3, тогда как при двукратных и трехкратных ошибках наоборот код Грея проигрывает. Важно так же отметить, что «хороший» выбор варианта не отличается по сложности реализации от других, что подчеркивает необходимость оптимизации выбора варианта кодирования в зависимости от вида ошибок в цифровом сигнале.

Исследованию данной проблемы были посвящены работы [39,40,41], при создании систем цифровой передачи аналоговых сигналов, в частности, речевых сигналов по каналам ИКМ-систем со скоростью 64 кБит/с.

В этих реализациях для повышения точности определения значений отсчетов было выбрано $N = 256$ уровней квантования, что соответствовало кодовой комбинации, состоящей из $n = 8$ двоичных символов.

Как было показано выше ошибки в кодовых комбинациях приводят к неверному восстановлению значения уровня отсчета, что приведет к соответствующим искажениям восстанавливаемого аналогового сигнала. При этом от выбора способа кодирования отсчетов зависит спектр возникающих искажений с учетом вероятности возникновения ошибок той или иной кратности, что будет подробно исследовано в Разделе 3.

Воспользовавшись формулой (1.4), несложно рассчитать число вариантов кодирования значений отсчетов для $n = 8$. Это число равно $(2^8)! = (256)!$.

Совершенно понятно, что с учетом возможностей вычислительной техники того времени полный перебор всех вариантов был невозможен. Поэтому исследованию были подвергнуты варианты чаще всего встречающиеся при

практическом использовании, а именно, взвешенный двоичный код и код Грэя, отличающийся тем, что находящиеся рядом двоичные кодовые комбинации отличаются друг от друга ровно в одном символе.

Исходя из предположений о существенно большей вероятности однократных ошибок в кодовой комбинации из $n=8$ символов, был сделан вывод о преимуществе кода Грэя. Проведенные в этих условиях экспериментальные исследования подтвердили правильность сравнения кода Грэя и взвешенного кода.

В дальнейшем было замечено, что с учетом особенности восприятия человеком восстанавливаемого речевого сигнала в условиях возникающих ошибок, целесообразно реализовывать нелинейную шкалу квантования, что также сносит особенности при выборе метода кодирования отсчетов.

Дальнейшее совершенствование методов аналого-цифрового преобразования речевых сигналов позволило разработать решения, когда скорость цифрового (дискретного) сигнала было значительно уменьшена, а именно, от 64 кБит/с в системах ИКМ до 6,3 кБит/с, это показано в таблице 1.9.

Средняя субъективная оценка (MOS – mean opinion score) или психологическая реакция человека на воспроизводимую речь. Оценка по шкале MOS определяется путем обработки оценок, даваемых группами слушателей. Наиболее предпочтительным среди приведенных методов кодирования с точки зрения соотношения качество речи / скорость потока является алгоритм G.723.1.

Таблица 1.9 – Средняя субъективная оценка для разных алгоритмов кодирования

Название алгоритма	MOS
G.711 (PCM; 64 кбит/с)	4,1
G.726 (ADPCM; 32 кбит/с)	3,8
G.728 (LD-CELP; 16 кбит/с)	3,6
G.723.1 (ACELP; 5,3 кбит/с)	3,7
G.723.1 (MP-MLQ; 6,3 кбит/с)	3,9

1.4. Кодирование в сигнально-кодовых конструкциях

После формирования цифрового сигнала для передачи по каналу связи необходимо осуществить согласование спектральных характеристик сигнала с частотными характеристиками канала. Этой цели служат методы модуляции, получившие за последние десятилетия интенсивное развитие. От простых методов амплитудной, частотной и фазовой модуляции, применяемых в телеграфии [44,45], современные методы являют собой методы многократной модуляции, сочетающие несколько видов модуляции, что в конечном итоге позволяет получить большую скорость передачи, приближающуюся к теоретически достижимому пределу по пропускной способности канала связи, обладающему определенной полосой пропускания ΔF .

Методы многократной дискретной модуляции, называемые зачастую сигнально-кодовой конструкцией, весьма разнообразны [16]. Однако общим для них является переход на передаче от дискретного сигнала к аналоговому, а на приемной стороне наоборот. На передаче, в зависимости от используемой конструкции, дискретной кодовой комбинации из n двоичных символов ставится в соответствие определенный вид модулируемого сигнала, а на приеме происходят обратные преобразования. При этом, также как это было рассмотрено выше, варианты сопоставления, в условиях действия помех, играют заметную роль с точки зрения обеспечения большей помехоустойчивости.

Рассмотрим это на примере. На рисунках 1.6, 1.7 изображены основные конфигурации ансамблей с $M = 2^k$, $k = 2...7$. Линии, соединяющие сигнальные точки, позволяют установить способ построения ансамблей. Все ансамбли можно разделить на две группы.

К первой относятся сигналы поверхностно-сферической укладки, когда сигнальные точки расположены на поверхности N -мерной сферы. В рассматриваемом случае ($N = 2$) такой сферой является окружность, ансамбли 1.4, 1.2. Сигнальные точки расположены на одинаковом расстоянии от начала координат, т.е. все сигналы ансамбля имеют одинаковые энергии. Очевидно, что

расположение сигнальных точек на поверхности сферы с ростом объема ансамбля M все более отличается от плотнейшего [17].

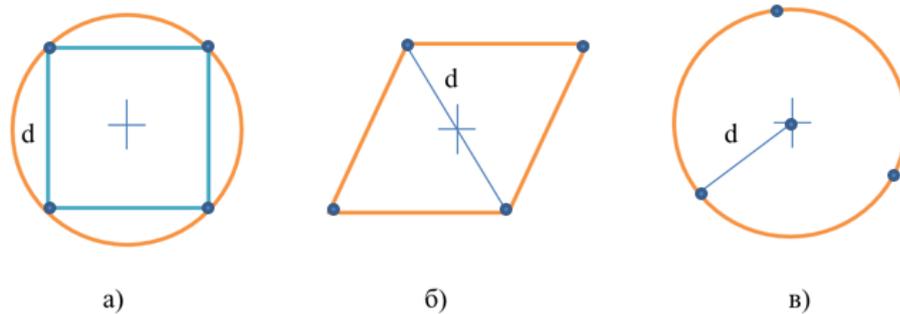


Рисунок 1.7 – Ансамбли двумерных сигналов при $M=4$

Большинство представленных ансамблей относится ко второй группе и характеризуется объемно-сферической укладкой сигнальных точек. Это и позволяет в ряде случаев получить плотность укладки, близкую к максимальной.

Рассмотрим ансамбль (а) на рисунке 1.7. Этот ансамбль является простейшим, полученным при расположении сигнальных точек в узлах квадратной сети.

Сигналы имеют одинаковые энергии и находятся на одинаковом расстоянии от начала координат. Сигналы отличаются только начальными фазами. Подобный ансамбль часто отождествляют с набором сигналов с фазовой модуляцией (ФМ) и числом позиций фазы $M = 4$ [18-21].

Ансамбль (в) на рисунке 1.7 состоит из трех сигналов, равномерно распределенных на окружности, и четвертого сигнала, расположенного в начале координат. Эти сигналы с амплитудно-фазовой модуляцией (АФМ).

Ансамбли (б), (в) построены на основе квадратной сети с различными расположениями сигнальных точек.

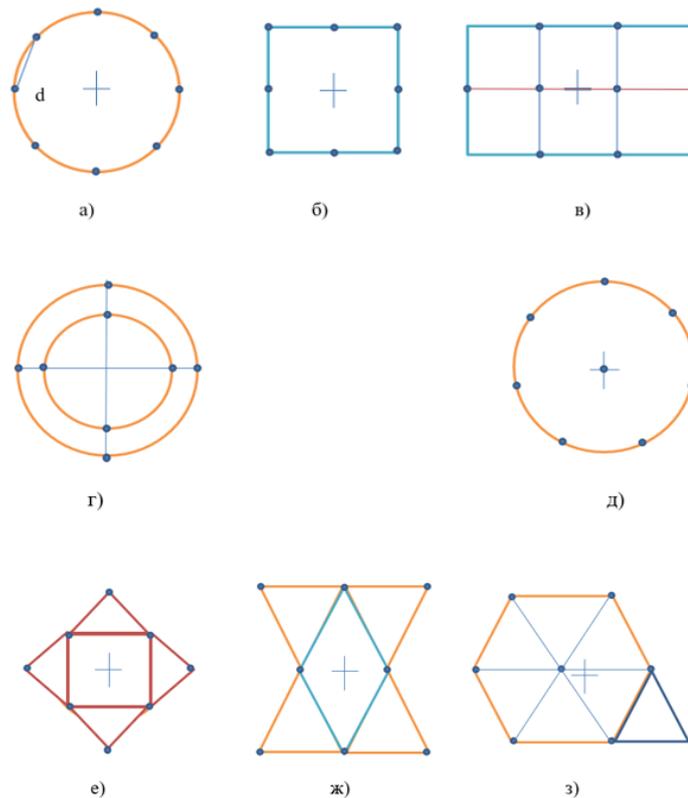


Рисунок 1.8 – Ансамбли двумерных сигналов при $M=8$

Манипуляционные коды для круговых расположений сигнальных точек на плоскости детально изучены в работе [17]. При $M=8$ существует шесть лучших вариантов манипуляционных кодов, эквивалентных друг другу. Один из них-код Грея.

1.5 Выводы по разделу

Показано, что при любых преобразованиях сигналов из одного вида в другой имеется возможность оптимизации данного выбора при сопоставлении элементов исходного сигнала с его отображением. При этом эффективность преобразования сигналов из одного вида в другой должна определяться не только скоростью кодирования, но и выбором метода сопоставления исходного сигнала и его отображения после преобразования.

Оптимизация метода отображения исходного сигнала в условиях действия помех позволяет получить заметный выигрыш по помехоустойчивости. При этом скорость кодирования остается неизменной, что является важным и существенным

фактором, поскольку достигаемый положительный эффект может быть получен без каких-либо дополнительных затрат, уменьшающих пропускную способность канала.

Виды помех, влияющие на появление ошибок в цифровом сигнале, необходимо учитывать, поскольку распределение ошибок оказывается существенным фактором при оптимизации выбора метода отображения исходного сигнала.

В известной литературе исследованы частные случаи анализа и оптимизации методов отображения исходного сигнала при его преобразованиях в процессе передачи от источника к получателю и нет общей теории исследования данной проблемы, которую в аналитическом виде можно представить как проблему оптимального отображения сигналов в различных метрических пространствах в условиях влияния мешающих факторов, искажающих элементы одного пространства, назовем его исходным, в результате чего элементы другого пространства, назовем его последующим, получают определенные искажения, величина которых нелинейно связана с ошибками или искажениями исходного пространства.

2. РАЗРАБОТКА ОБЩЕГО (УНИВЕРСАЛЬНОГО) МЕТОДА ОЦЕНКИ ЭФФЕКТИВНОСТИ ПЕРВИЧНОГО КОДИРОВАНИЯ

2.1. Постановка задачи

В известных работах [1,9] показано, что наибольшая эффективность первичного кодирования сообщений источника достигается, когда скорость кодирования равна энтропии источника. К сожалению, это важное положение не всегда поддержано конкретным решением по определению энтропии для тех или иных источников. Так в общей классификации возможных сообщений (сигналов) можно отметить, что все они подразделяются на непрерывные (аналоговые), непрерывно-дискретные, дискретно-непрерывные и дискретные сообщения (сигналы), а с учетом используемой элементной базы и возможностей вычислительной техники в дальнейшем эти сигналы преобразуются в дискретный (цифровой) сигнал.

Эффективность преобразования непрерывных сигналов в дискретные определяются, исходя из известной теоремы А.Н. Котельникова, а также статистическими особенностями получаемых отсчетов, точностью аппроксимации и прочие [12-15].

Ряд других исследований [25,27] определили эффективность преобразования анизохронных сигналов (непрерывно дискретных сигналов) в дискретные.

При этом указанные результаты появлялись в разное время, и не было предложено общего подхода для оценки эффективности преобразования названных различных сигналов в дискретный (цифровой).

В этой связи целью дальнейшего исследования явилась разработка общего метода оценки эффективности преобразований для различных сигналов в дискретный через расчет энтропии (эпсилон-энтропии) источника.

2.2. Предельная эффективность преобразования непрерывно-дискретного сигнала в цифровой при равновероятном появлении элементов различной длительности

Рассмотрим непрерывно-дискретный двоичный сигнал, поступающий от источника, и цифровую последовательность дискретного канала, с помощью которой необходимо передать этот сигнал, это показано на рисунке 2.1 (а, б).

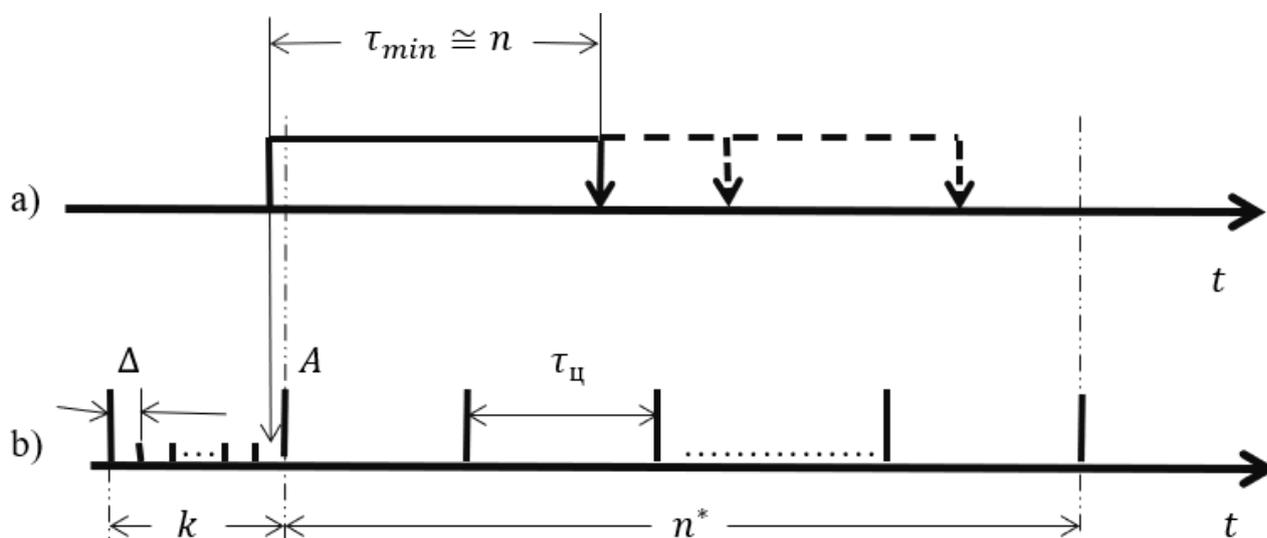


Рисунок 2.1 - Диаграммы преобразования непрерывно-дискретного сигнала в цифровой

Также как при методе «скользящего индекса с подтверждением» условимся, что первый фронт в непрерывно-дискретном сигнале от источника будет отображаться «стартовым» символом A , совпадающим по знаку со знаком поступающего элемента исходного сигнала. Пусть также данный фронт, как это изображено на рисунке 2.1, окажется в крайнем правом участке Δ рассматриваемого временного интервала длительностью $\tau_{ц}$.

Рассмотрим после символа A некий интервал, состоящий из n^* элементов длительностью Δ . Назовем этот интервал блоком и определим какое количество

различных реализаций исходного непрерывно-дискретного сигнала возможно над этим блоком.

В [28] было предложено решение данного вопроса.

Воспользовавшись этим, приведем полученные выражения, а именно, формулу для расчета числа возможных реализаций, имеющих над блоком n^* ровно l фронтов:

$$N(l) = C_{n^* - l + 1}^l, \quad (2.1)$$

а также формулу для расчета общего числа реализаций над блоком n^* :

$$N = \sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^* + j}{n} \rfloor} C_{n^* + j - in + i}^i, \quad (2.2)$$

в которой учтено, что первый фронт в непрерывно-дискретном сигнале может оказаться в любом из k интервалов Δ , это показано на рисунке 2.1, в результате чего блок n^* увеличивается на $(k-1)$ таких интервалов.

Непрерывно-дискретный сигнал от источника состоит из элементов, длительность которых может быть любой, но не меньше некоторого минимального значения τ_{\min} . Однако для отображения в виде последовательности символов дискретного канала необходимо осуществить дискретизацию длительности элементов с абсолютной погрешностью Δ , что составит максимальную относительную погрешность, оцениваемую по формуле (1.3). В итоге минимальную длительность элемента в исходном сигнале τ_{\min} можно сопоставить с соответствующей оценкой n , равной минимальному числу интервалов Δ .

Из [9] известно, что минимальная скорость кодирования сообщений от источника равна энтропии. При этом для дискретного источника с минимальным элементом, состоящем из n интервалов величиной Δ , энтропия максимальна, когда все сообщения от источника равновероятны. А, следовательно, энтропия в этом случае для непрерывно-дискретного сигнала может быть оценена по формуле:

$$H_{H-д}^2(\varepsilon \leq \delta_{\max}) = \log_2 N \quad (2.3)$$

где в обозначении энтропии $H_{H-д}^2$ непрерывно-дискретного сообщения в верхнем индексе цифра 2 обозначает, что сообщение двоичное.

С учетом (2.2) имеем выражение для расчета энтропии двоичного непрерывно-дискретного сигнала при заданном значении n , определяющем относительные искажения отображения исходного сигнала δ_{\max} , и соответствующей длине блока n^* .

$$H_{H-д}^2(\varepsilon \leq \delta_{\max}) = \log_2 \sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{(n^*+j)}{n} \rfloor} C_{n^*+j-in+i}^i \quad (2.4)$$

где

$$\delta_{\max} = \frac{1}{n} \cdot 100\% \quad (2.5)$$

Представим для наглядности зависимости $H_{H-д}^2(\varepsilon \leq \delta_{\max})$ от значений n и n^* для различных значений $k=2, 4$ и 8 (рисунок 2.2 а, б, в).

- а) $k=2$, для $n=4$ и далее, а $n^* \geq n$ и далее
- б) $k=4$, для $n=12$ и далее, а $n^* \geq n$ и далее
- в) $k=8$, для $n=32$ и далее, а $n^* \geq n$ и далее

Представленные на рисунке 2.2 зависимости позволяют судить об изменении энтропии источника, а, следовательно, и скорости кодирования с изменением длины блока n^* , различных n и разных k .

Поскольку непрерывно-дискретный сигнал от источника в дискретном канале отображается периодической последовательностью символов, следующих с интервалом $\tau_{ц}$, то число двоичных символов дискретного канала можно определить, как $n_{ок} = \frac{n^*}{k}$.

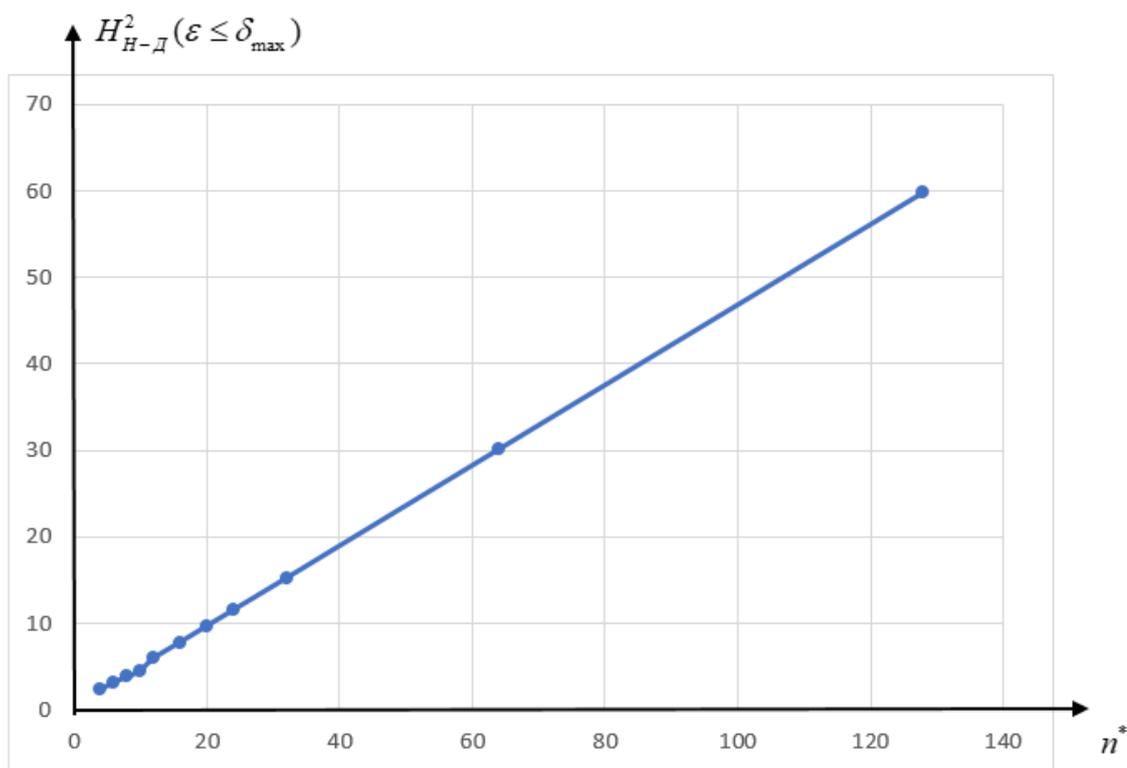


Рисунок 2.2 а - Зависимости энтропии непрерывно-дискретного двоичного сигнала $H_{H-Д}^2(\varepsilon \leq \delta_{\max})$ от n^* для различных значений $k=2$

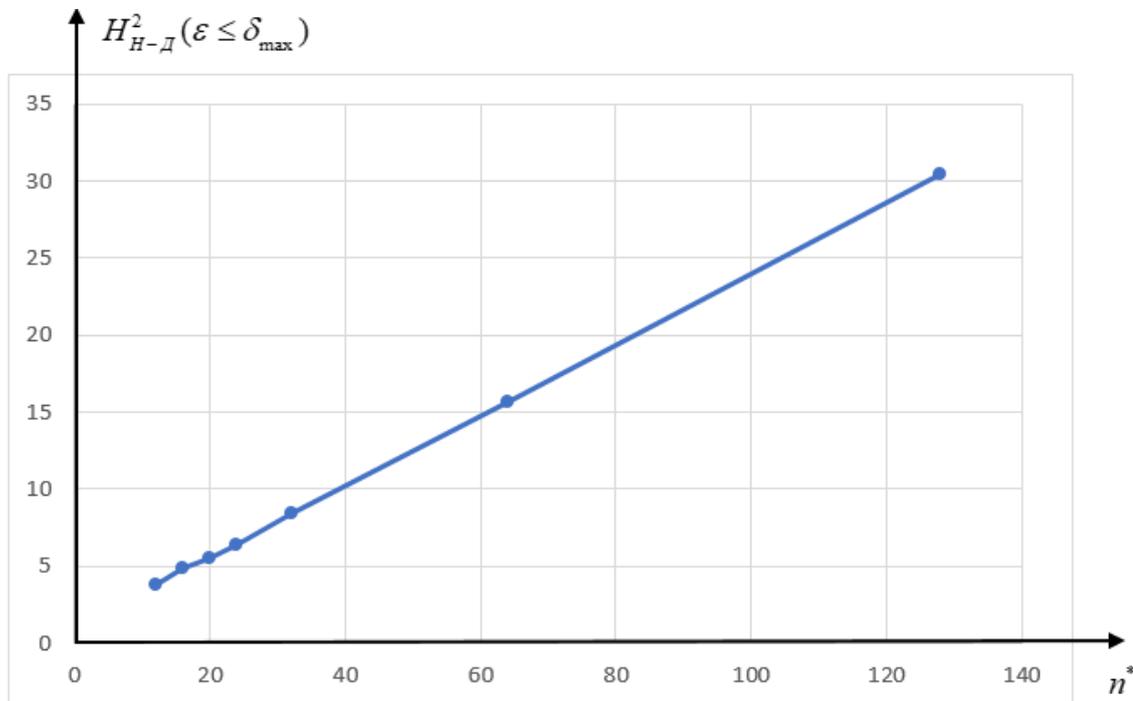


Рисунок 2.2 б - Зависимости энтропии непрерывно-дискретного двоичного сигнала $H_{H-Д}^2(\varepsilon \leq \delta_{\max})$ от n^* для различных значений $k=4$

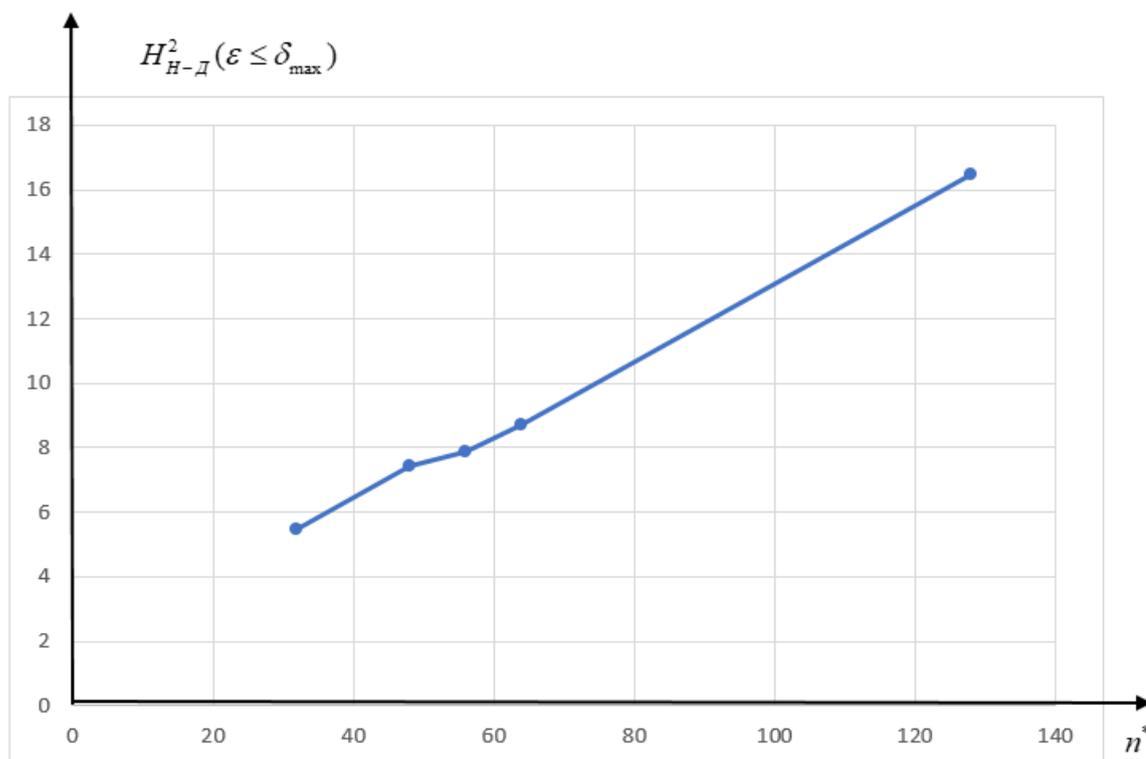


Рисунок 2.2 в - Зависимости энтропии непрерывно-дискретного двоичного сигнала $H^2_{H-D}(\epsilon \le \delta_{max})$ от n^* для различных значений $k=8$

Очевидно, что при безызбыточном кодировании число различных двоичных кодовых комбинаций в дискретном канале должно быть равно числу возможных реализаций исходных сигналов от источника. А потому

$$\sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{(n^*+j)}{n} \rfloor} C_{n^*+j-in+i}^i = 2^{\frac{n^*}{k}} \quad (2.6)$$

Что после логарифмирования и некоторых преобразований можно записать в следующем виде:

$$1 = \frac{H^2_{H-D}(\epsilon \le \delta_{max})}{\frac{n^*}{k}} \quad (2.7)$$

После подстановки в (2.7) выражения из (2.4) и некоторых преобразований получаем:

$$k = \frac{n^*}{\log_2 \sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} C_{n^*+j-in+i}^i} \quad (2.8)$$

На практике важным показателем является эффективность преобразования, указывающая количество двоичных символов дискретного канала, приходящихся на один элемент исходного сигнала. Для рассматриваемого случая это формально соответствует следующему отношению:

$$\mathfrak{E}_{H-D}^2 = \frac{k}{n} \quad (2.9)$$

где верхний индекс в эффективности \mathfrak{E}_{H-D}^2 указывает на двоичный характер сигнала от источника информации.

С учетом (2.8) получаем:

$$\mathfrak{E}_{n-D}^2 = \frac{n^*}{n \log_2 \sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} C_{n^*+j-in+i}^i} \quad (2.10)$$

Построим зависимость \mathfrak{E}_{H-D}^2 от длины блока n^* при заданных значениях n и k . Положим для определенности $n=12$, а $k=4$. График представлен на рисунке 2.3.

Представленная на рисунке 2.3 зависимость показывает, что с ростом длины блока n^* эффективность преобразования непрерывно-дискретного сигнала, поступающего от источника, в последовательность символов дискретного канала нарастает. В тоже время эта зависимость указывает на величину задержки, эквивалентную длине блока n^* , что, безусловно, следует учитывать при выборе соответствующего метода преобразования непрерывно-дискретного сигнала в цифровой.

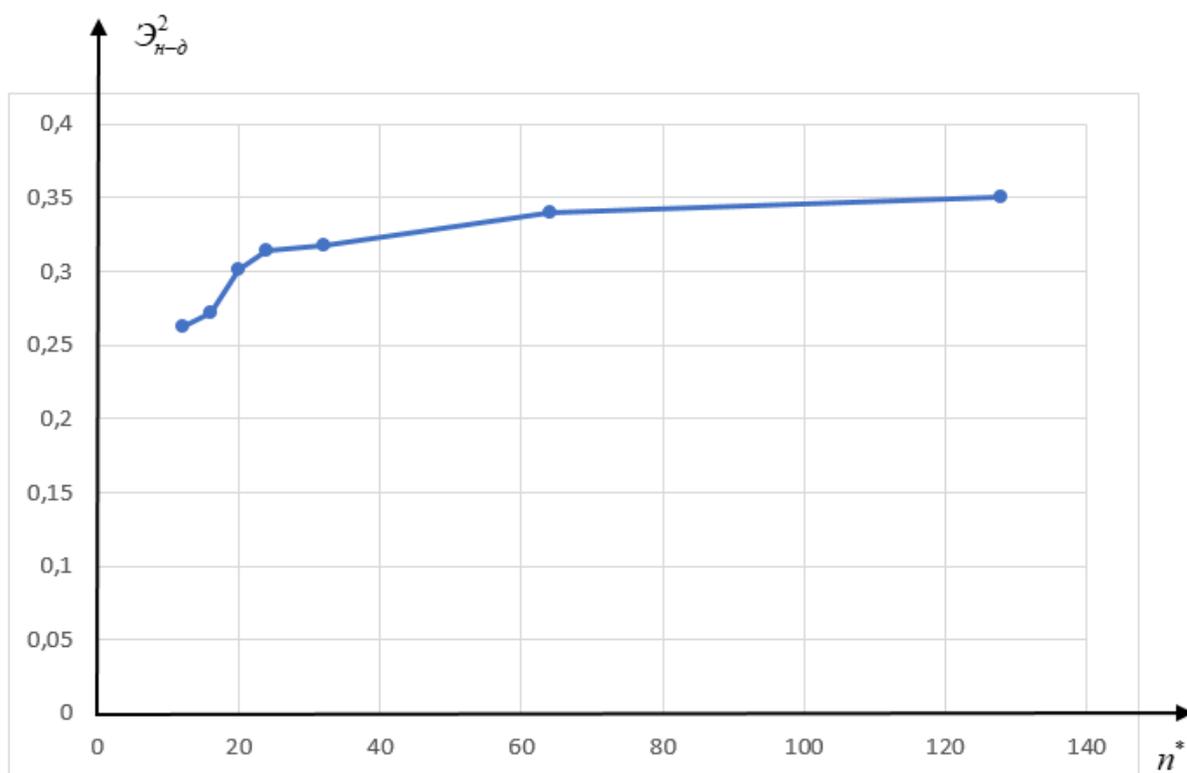


Рисунок 2.3 - Зависимость эффективности \mathcal{E}_{n-d}^2 от длины блока n^*

При рассмотрении зависимости, показанной на рисунке 2.3, естественно возникает вопрос о том, каким оказывается предельное значение эффективности при длине блока n^* , стремящейся к бесконечности. Для этого рассмотрим соответствующий предел.

$$\lim_{n^* \rightarrow \infty} \mathcal{E}_{n-d}^2 = \frac{n^*}{\lim_{n^* \rightarrow \infty} n \log_2 \sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} C_{n^*+j-in+i}^i} \quad (2.11)$$

В [31,33] было показано, что выражение $\sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} C_{n^*+j-in+i}^i$ может быть заменено его максимальным членом, в результате чего предел выражения (2.11) равен:

$$\mathcal{E}_{n-d}^2 = \frac{1}{n \cdot [(1 - \beta n + \beta) \log_2(1 - \beta n + \beta) - (1 - \beta n) \log_2(1 - \beta n) - \beta \log_2 \beta]} \quad (2.12)$$

где величина β определяется по формуле:

$$\beta = \frac{M}{1 + Mn} \quad (2.13)$$

а величина M - это корень решения уравнения:

$$M \cdot (M + 1)^{n-1} = 1 \quad (2.14)$$

Поскольку $\tau_{\min} \cong n$ с возможной максимальной абсолютной погрешностью Δ , то с ростом величины n погрешность отображения элементов непрерывно-дискретного сигнала от источника уменьшается согласно (2.5).

Этот процесс имеет зависимость, показанную на рисунке 2.4.

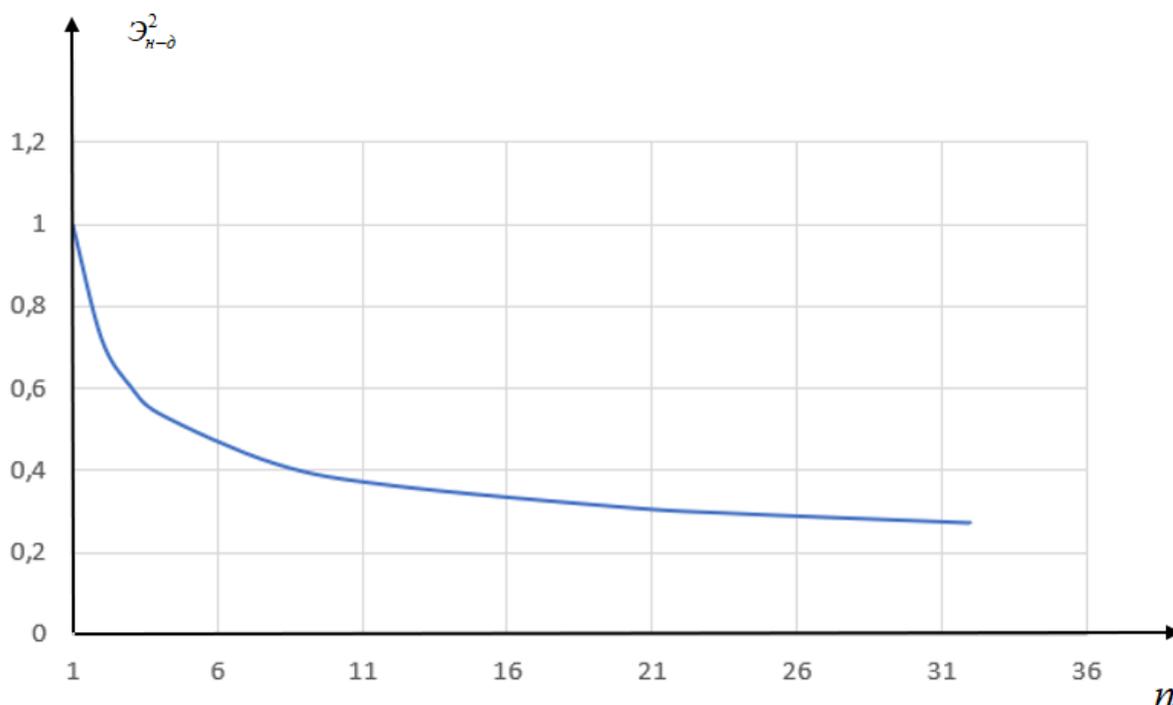


Рисунок 2.4 - Зависимость эффективности \mathcal{E}_{n-d}^2 от точности передачи длительности элементов исходного непрерывно-дискретного сигнала n

Приведенная на рисунке 2.4 зависимость указывает на потенциальную границу эффективности преобразования двоичного непрерывно-дискретного сигнала в цифровую последовательность дискретного канала. В частности, упоминаемый выше метод «скользящего индекса с подтверждением» обладает близкими к этой границе характеристиками эффективности, а именно, когда $n = 4$

эффективность равна 0,5 при $\mathcal{E}_{n-d}^2 = 0,538$, а когда $n = 12$ эффективность равна 0,3(3) при $\mathcal{E}_{n-d}^2 = 0,362$.

Отметим также, что при таких сравнениях следует обращать внимание на задержку при преобразовании непрерывно-дискретного сигнала в цифровой, учитывая, что предельное значение получено при условии длины блока n^* , стремящейся к бесконечности.

В общем случае непрерывно-дискретный сигнал от источника может содержать не два, а Q уровней.

Исследуем это на примере диаграмм, изображенных на рисунке 2.5

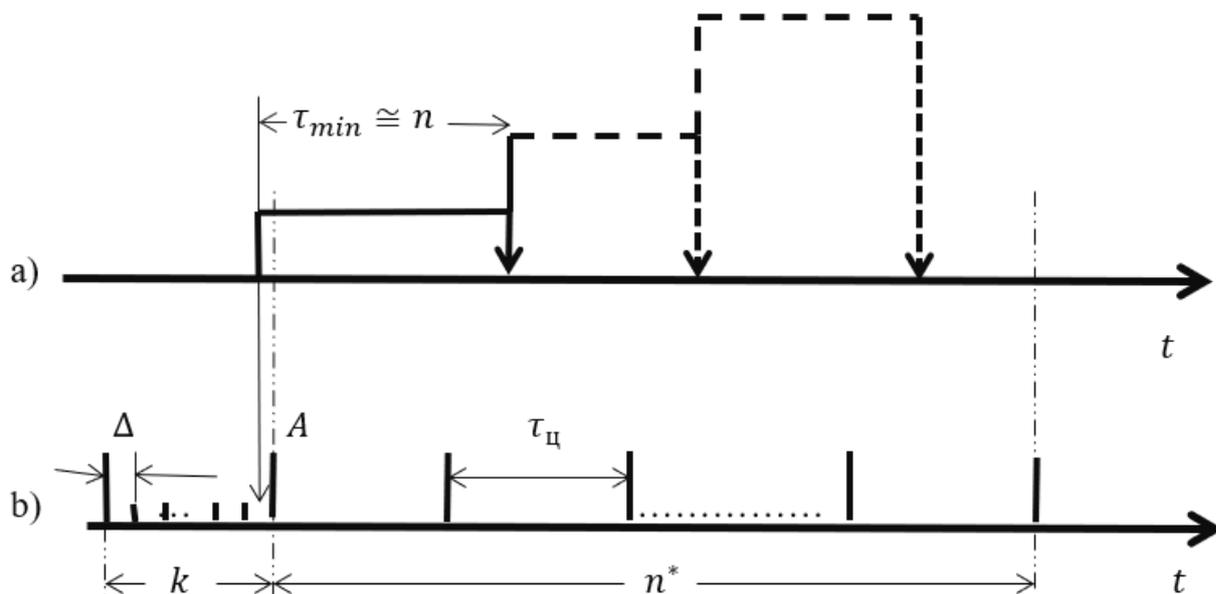


Рисунок 2.5 - Диаграммы преобразования Q -ичного непрерывно-дискретного сигнала в цифровой

В отличие от ситуации, рассмотренной на диаграммах рисунке 2.1, в данном случае в сигнале от источника может содержаться не два, а Q уровней. При этом будем полагать, что длительность этих элементов может быть любой, но не меньше некоторого минимального значения $\tau_{min} \cong n$. Причем условимся, что это требование выполняется для всех уровней в исходном сигнале от источника.

Предположим также, что появление того или иного уровня в исходном сигнале не зависит от предыдущих значений и является равновероятным событием.

Тогда с учетом (2.1) число различных реализаций над блоком n^* , имеющих ровно l фронтов, можно рассчитать по формуле:

$$N^Q(l) = (Q-1)^{l+1} \cdot C_{n^*-ln+l}^l \quad (2.15)$$

Общее число реализаций над блоком n^* для Q -ичного сигнала определится соответственно, как:

$$N^Q = \sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} (Q-1)^{i+1} \cdot C_{n^*+j-in+i}^i, \quad (2.16)$$

Максимальное значение энтальпии равно:

$$H_{H-d}^Q(\varepsilon \leq \delta_{max}) = \log_2 N^Q = \log_2 \left(\sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor (n^*+j)/n \rfloor} (Q-1)^{i+1} \cdot C_{n^*+j-in+i}^i \right) \quad (2.17)$$

Поскольку сообщения от источника отображаются посредством двоичных символов дискретного канала, то представляет интерес оценить эффективность такого преобразования, которая для данного случая определяется по формуле:

$$\mathfrak{E}_{H-d}^Q = \frac{n^*}{n \log_2 \left(\sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} (Q-1)^{i+1} \cdot C_{n^*+j-in+i}^i \right)} \quad (2.18)$$

Для определения предельного значения эффективности устремим величину блока n^* к бесконечности и рассмотрим предел:

$$\lim_{n^* \rightarrow \infty} \mathfrak{E}_{H-d}^Q = \lim_{n^* \rightarrow \infty} \frac{n^*}{n \log_2 \left(\sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} (Q-1)^{i+1} \cdot C_{n^*+j-in+i}^i \right)} \quad (2.19)$$

Осуществим некоторые упрощения, рассмотрев следующее неравенство:

$$\frac{\log_2 N_{max}^Q}{n^*} \leq \frac{\log_2 \left(\sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} (Q-1)^{i+1} \cdot C_{n^*+j-in+i}^i \right)}{n^*} \leq \frac{\log_2 \left(k \cdot \lfloor \frac{(n^*+k-1)}{n} \rfloor \cdot N_{max}^Q \right)}{n^*} \quad (2.20)$$

где N_{\max}^Q - максимальный член, входящий в выражение двойной суммы.

Рассмотрим предел левой части неравенства (2.20) при $n^* \rightarrow \infty$.

$$\lim_{n^* \rightarrow \infty} \frac{\log_2 N_{\max}^Q}{n^*} \quad (2.21)$$

и сравним его с пределом правой части неравенства (2.20) при условии, что величины k и n конечны.

$$\lim_{n^* \rightarrow \infty} \frac{\log_2(k \cdot \lfloor \frac{(n^* + k - 1)}{n} \rfloor \cdot N_{\max}^Q)}{n^*} = \lim_{n^* \rightarrow \infty} \frac{\log_2 k}{n^*} + \lim_{n^* \rightarrow \infty} \frac{\log_2 \lfloor \frac{(n^* + k - 1)}{n} \rfloor}{n^*} + \lim_{n^* \rightarrow \infty} \frac{\log_2 N_{\max}^Q}{n^*} = \lim_{n^* \rightarrow \infty} \frac{\log_2 N_{\max}^Q}{n^*} \quad (2.22)$$

Сравнивая (2.21) и (2.22) убеждаемся в том, что при $n^* \rightarrow \infty$ выражение стоящее в двойной сумме может быть заменено максимальным членом, который соответствует некоторому m - ому числу фронтов,

$$N_{\max}^Q = N^Q(m) = N^Q(\beta \cdot n^*) \quad (2.23)$$

где $0 < \beta < 1$ – коэффициент, определяющий максимальный член N_{\max}^Q .

Подставим в (2.19) вместо двойной суммы максимальный член и проведем упрощения:

$$\begin{aligned} \lim_{n^* \rightarrow \infty} \mathfrak{D}_{n^*}^Q &= \lim_{n^* \rightarrow \infty} \frac{n^*}{n \log_2 N^Q(m)} = \lim_{n^* \rightarrow \infty} \frac{n^*}{n \log_2 ((Q-1)^{m+1} \cdot C_{n^* - mn + m}^m)} = \\ &= \lim_{n^* \rightarrow \infty} \frac{n^*}{n \log_2 ((Q-1)^{(\beta \cdot n^* + 1)} \cdot C_{n^* - \beta \cdot n^* n + \beta \cdot n^*}^{\beta \cdot n^*})} = \lim_{n^* \rightarrow \infty} \frac{n^*}{n(\beta \cdot n^* + 1) \cdot \log_2(Q-1) + n \cdot \log_2 C_{n^* - \beta \cdot n^* n + \beta \cdot n^*}^{\beta \cdot n^*}} = \\ &= \lim_{n^* \rightarrow \infty} \frac{n^*}{\beta \cdot n \cdot \log_2(Q-1) + n \cdot \frac{\log_2 C_{n^* - \beta \cdot n^* n + \beta \cdot n^*}^{\beta \cdot n^*}}{n^*}} \end{aligned} \quad (2.24)$$

Исследуем последний член в знаменателе равенства (2.24), применив известную формулу Стирлинга $S! \cong \sqrt{2\pi S} \cdot e^{-S} \cdot S^S$.

$$\begin{aligned}
\frac{\log_2 C_{n^* - \beta \cdot n^* n + \beta \cdot n^*}^{\beta \cdot n^*}}{n^*} &= \frac{\log_2 \left(\frac{(n^* - \beta \cdot n^* n + \beta \cdot n^*)!}{(n^* - \beta \cdot n^* n)! (\beta \cdot n^*)!} \right)}{n^*} = \\
&= \frac{\log_2 (n^* - \beta \cdot n^* n + \beta \cdot n^*)! - \log_2 (n^* - \beta \cdot n^* n)! - \log_2 (\beta \cdot n^*)!}{n^*} = \\
&= \frac{\log_2 \left(\sqrt{2\pi (n^* - \beta \cdot n^* n + \beta \cdot n^*)} \cdot e^{-n^* - \beta \cdot n^* n + \beta \cdot n^*} \cdot (n^* - \beta \cdot n^* n + \beta \cdot n^*)^{n^* - \beta \cdot n^* n + \beta \cdot n^*} \right)}{n^*} - \\
&\quad - \frac{\log_2 \left(\sqrt{2\pi (n^* - \beta \cdot n^* n)} \cdot e^{-n^* - \beta \cdot n^* n} \cdot (n^* - \beta \cdot n^* n)^{n^* - \beta \cdot n^* n} \right)}{n^*} - \frac{\log_2 \left(\sqrt{2\pi (\beta \cdot n^*)} \cdot e^{-\beta \cdot n^*} \cdot (\beta \cdot n^*)^{\beta \cdot n^*} \right)}{n^*}
\end{aligned} \tag{2.25}$$

Найдем предел выражения (2.25) при $n^* \rightarrow \infty$. После логарифмирования и проведения несложных преобразований получаем:

$$\frac{\log_2 C_{n^* - \beta \cdot n^* n + \beta \cdot n^*}^{\beta \cdot n^*}}{n^*} = (1 - \beta \cdot n + \beta) \cdot \log_2 (1 - \beta \cdot n + \beta) - (1 - \beta \cdot n) \cdot \log_2 (1 - \beta \cdot n) - \beta \cdot \log_2 \beta \tag{2.26}$$

Подставив выражение, полученное в (2.24), окончательно получаем выражение предельной эффективности преобразования Q - ичного непрерывно-дискретного сигнала в дискретный двоичный (цифровой):

$$\mathfrak{E}_{n \rightarrow 0}^Q = \frac{1}{n \cdot [\beta \cdot \log_2 (Q - 1) + (1 - \beta \cdot n + \beta) \cdot \log_2 (1 - \beta \cdot n + \beta) - (1 - \beta \cdot n) \cdot \log_2 (1 - \beta \cdot n) - \beta \cdot \log_2 \beta]} \tag{2.27}$$

Отметим, что поскольку при выводе выражения (2.27) выбирался максимальный член, то это означает, что коэффициент β - это коэффициент, при котором для определенного n выражение (2.27) минимально. Для определения этого найдем максимум знаменателя для значения коэффициента β на интервале $(0, 1)$.

Найдем первую производную следующего выражения:

$$\begin{aligned}
[\beta \cdot \log_2 (Q - 1) + (1 - \beta \cdot n + \beta) \cdot \log_2 (1 - \beta \cdot n + \beta) - (1 - \beta \cdot n) \cdot \log_2 (1 - \beta \cdot n) - \beta \cdot \log_2 \beta]' &= n \cdot \log_2 (Q - 1) + \\
+(1 - n) \cdot \log_2 (1 - \beta \cdot n + \beta) + (1 - \beta \cdot n + \beta) \cdot \frac{(1 - n)}{(1 - \beta \cdot n + \beta) \cdot \ln 2} + n \cdot \log_2 (1 - \beta \cdot n) + (1 - \beta \cdot n) \cdot \frac{n}{(1 - \beta \cdot n) \cdot \ln 2} - &\tag{2.28} \\
-\log_2 \beta - \beta \cdot \frac{1}{\beta \cdot \ln 2} &= n \cdot \log_2 (Q - 1) + (1 - n) \cdot \log_2 (1 - \beta \cdot n + \beta) + n \cdot \log_2 (1 - \beta \cdot n) - \log_2 \beta
\end{aligned}$$

Составим уравнение:

$$n \cdot \log_2 (Q - 1) + (1 - n) \cdot \log_2 (1 - \beta \cdot n + \beta) + n \cdot \log_2 (1 - \beta \cdot n) - \log_2 \beta = 0 \tag{2.29}$$

Преобразуем его к следующему виду:

$$(1-n) \cdot \log_2(1-\beta \cdot n + \beta) + n \cdot \log_2(1-\beta \cdot n) - \log_2 \beta = -n \cdot \log_2(Q-1),$$

далее

$$\log_2 \frac{(1-\beta \cdot n)^n}{(1-\beta \cdot n + \beta)^{(n-1)} \cdot \beta} = \log_2 \frac{1}{(Q-1)^n},$$

далее

$$\frac{(1-\beta \cdot n)^n}{(1-\beta \cdot n + \beta)^{(n-1)} \cdot \beta} = \frac{1}{(Q-1)^n},$$

далее

$$\frac{(1-\beta \cdot n + \beta)^{(n-1)} \cdot \beta}{(1-\beta \cdot n)^n} = (Q-1)^n$$

Таким образом имеем:

$$\frac{\beta}{1-\beta \cdot n} \cdot \left(\frac{1-\beta \cdot n + \beta}{1-\beta \cdot n} \right)^{n-1} = (Q-1)^n \quad (2.30)$$

Обозначим через

$$x = \frac{\beta}{1-\beta \cdot n} \quad (2.31)$$

Тогда выражение (2.30) после подстановки (2.31) примет вид

$$x \cdot (x+1)^{n-1} = (Q-1)^n \quad (2.32)$$

Решая это трансцендентное уравнение при заданных Q и n , находим x .

После этого воспользовавшись соотношением

$$\beta = \frac{x}{1+x \cdot n} \quad (2.33)$$

Пример 1.

Для $Q=4$ и $n=4$, находим $x=2,285$, при котором выражение $x \cdot (x+1)^{n-1} = 81,001$, что с достаточной точностью совпадает с выражением $(Q-1)^n = 81$.

Далее, согласно (2.33) находим: $\beta = \frac{x}{1+x \cdot n} = 0,225$

Подставляя полученные значения в формулу (2.27), находим предельное значение эффективности $\mathcal{E}_{n-d}^Q = \mathcal{E}_{n-d}^4 = 0,387$. Аналогично можно рассчитать значения предельной эффективности для других значений Q и n .

На рисунке 2.6 представлены зависимости предельной эффективности, построенные по рассчитанным значениям при $Q = 2, 4, 8, 16$ и $n = 2, 4, 12, 24$.

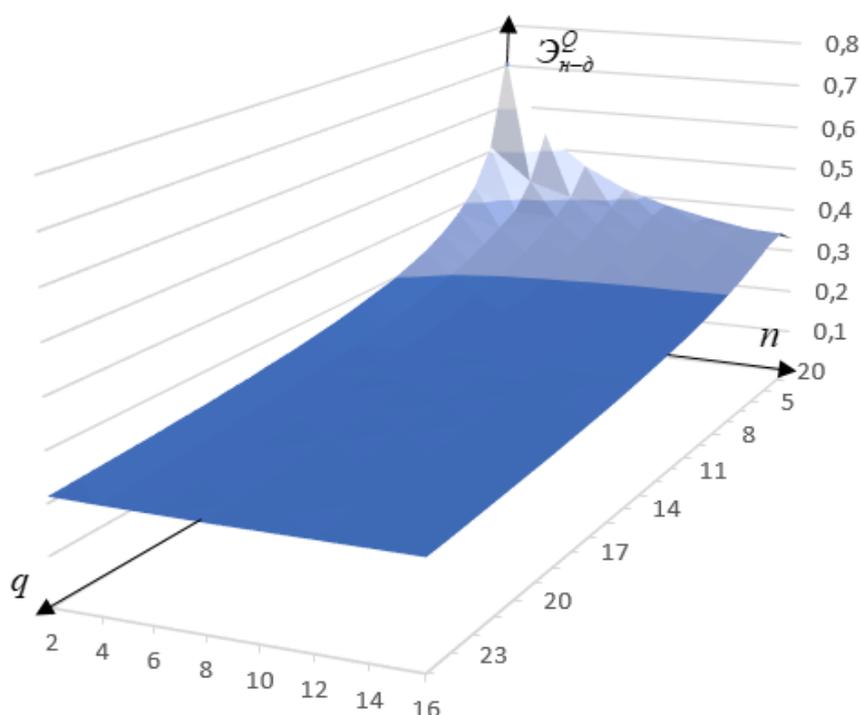


Рисунок 2.6 - Предельная эффективность преобразования Q-ичного непрерывно-дискретного сигнала в дискретный

Анализ полученных зависимостей указывает на границы предельной эффективности в зависимости от точности преобразования непрерывно-дискретного сигнала в дискретный и в зависимости от числа уровней в исходном сигнале от источника.

Получены адекватные оценки, показывающие необходимость больших «затрат» символов дискретного канала при передаче исходных сигналов с большим числом уровней, а также с большей точностью передачи значений длительностей элементов исходного сигнала.

2.3. Предельная эффективность преобразования непрерывно-дискретного сигнала в цифровой при неравновероятном появлении элементов различной длительности

Проведенное выше исследование предельной эффективности преобразования непрерывно-дискретного сигнала в цифровой осуществлялось при условии, что в сигнале от источника элементы могут иметь любую длительность, но не меньше, чем ($\tau_{\min} \cong n$). В тоже время из практических соображений интересно исследовать также случай, когда помимо этого ограничения существует и ограничение максимальной длительности элементов исходного сигнала (τ_{\max}).

В [28] предложено для двоичного непрерывно-дискретного сигнала оценку предельной эффективности проводить для оговоренных выше ограничений ($\tau_{\min} \cong n$) и (τ_{\max}), представляя исходный сигнал как $(S+1)$ – множество $X = \{x_0; x_1; \dots; x_s\}$ элементами - $\{x_0; x_1; \dots; x_s\}$ являющееся отображением соответствующих элементов двоичного непрерывно-дискретного сигнала с длительностью равной $\Delta \cdot n; \Delta \cdot (n+1); \dots; \Delta \cdot (n+S)$.

При таком описании каждой реализации двоичного непрерывно-дискретного сигнала над блоком n^* будет соответствовать $(S+1)$ – вектор $(a_0; a_1; \dots; a_s)$ над

множеством X с целочисленными координатами, такими, что $a_0 + a_1 + \dots + a_s = l$ при этом $a_i \geq 0$, $a_i = 0; 1; \dots; S$

Величины $(a_0; a_1; \dots; a_s)$ показывают, сколько раз встречаются элементы длительности $\Delta \cdot n; \Delta \cdot (n+1); \dots; \Delta \cdot (n+S)$ в реализации непрерывно-дискретного сигнала над блоком n^* .

Исходя из этого, для любой реализации двоичного непрерывно-дискретного сигнала будет справедливо неравенство:

$$a_0 n + a_1 (n+1) + \dots + a_s (n+S) \leq n^* \quad (2.34)$$

При отображении элементов двоичного непрерывно-дискретного сигнала его минимальный элемент (τ_{\max}) отождествляется с n интервалами величиной Δ . Тогда значение величины S , отождествляемой с элементами исходного двоичного непрерывно-дискретного сигнала, обладающими максимальной длительностью (τ_{\max}), могут быть определены следующим образом:

$$S = n^* - n \cdot l \quad (2.35)$$

где l - определенное число фронтов, для которых устанавливается значение S .

Учитывая, что $a_0 + a_1 + \dots + a_s = l$ и, учитывая (2.35), преобразуем неравенство (2.34). В результате получаем следующее:

$$a_1 + 2a_2 + \dots + Sa_s \leq S \quad (2.36)$$

В данной аналитической модели каждой реализации непрерывно-дискретного сигнала, имеющей ровно l фронтов над блоком n^* , соответствует $(S+1)$ - векторов над множеством X . При этом число реализаций непрерывно-

дискретного сигнала, имеющих ровно l фронтов, согласно [28] соответствует (l) -мультимножеству первичной спецификации $[X_0^{a_0}; X_1^{a_1}; \dots; X_s^{a_s}]$, порожденному множеством X . Таким образом, величина $N(l)$ может быть подсчитана по формуле:

$$N(l) = \sum_{\substack{a_0 + a_1 + \dots + a_s = l \\ a_1 + 2a_2 + \dots + Sa_s \leq S}} \frac{l!}{a_0! a_1! \dots a_s!} \quad (2.37)$$

где суммирование в формуле (2.37) осуществляется для всех целочисленных решений системы уравнений

$$\begin{cases} a_0 + a_1 + \dots + a_s = l \\ a_1 + 2a_2 + \dots + Sa_s \leq S \end{cases} \quad (2.38)$$

Рассматриваемый аналитический подход для расчета числа реализаций непрерывно-дискретного сигнала над блоком n^* , формула (2.37), в количественном отношении эквивалентен тому, что было сделано выше при выводе формулы (2.1). Сравнение этих выражений, а именно, (2.1) и (2.37), показывает, что первое - значительно проще в проводимых вычислениях, однако второе - позволяет с большей детализацией осуществить исследование потенциально достижимой эффективности.

В частности, с введенным ограничением не только минимальной длительности элемента в исходном сигнале, но и с учетом ограничения максимальной длительности.

Проведенные в [28] расчеты показали, что дополнительное ограничение максимальной длительности элемента в исходном сигнале позволяют получить большую предельную эффективность преобразования двоичного непрерывно-дискретного сигнала в цифровой, а именно:

$$\mathcal{E}_{n-d}^{2*} = \frac{1}{\beta n [\log_2(2 - 2\beta n + \beta) - \log_2 \beta]} \quad (2.39)$$

Графически отобразим это на рисунке 2.7.

Дополним эти результаты, перейдя к рассмотрению Q -ичного исходного сигнала от источника с ограничением как минимальной, так и максимальной длительности элементов исходного сигнала.

Как было показано выше, минимальная длительность элемента исходного сигнала τ_{\min} эквивалентна n , а максимальная длительность τ_{\max} определяется формулой (2.35), когда над блоком n^* все элементы реализации исходного сигнала, содержащие l фронтов, обладают минимальной длительностью n .

Очевидно, что с ростом длины блока n^* величина S пропорционально возрастает и стремится к бесконечности с аналогичным ростом длины блока n^* при фиксированном числе фронтов l .

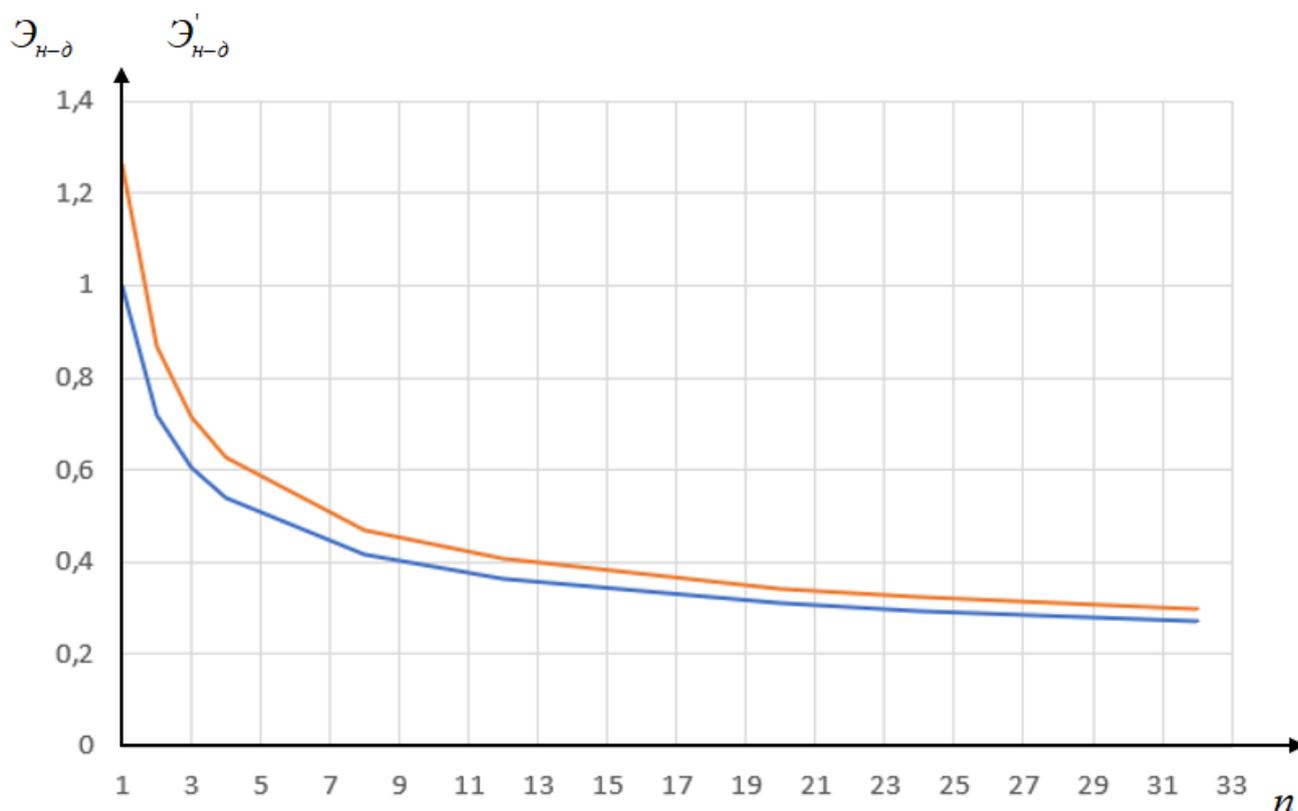


Рисунок 2.7 - Зависимости предельной эффективности преобразования двоичного непрерывно-дискретного сигнала в цифровой при ограничении минимальной длительности и при ограничении минимальной и максимальной длительности элементов исходного сигнала.

Введем ограничения максимальной длительности τ_{\max} через некий параметр p , принимающий целочисленные значения, согласно оговоренным выше дополнительным условиям к непрерывно-дискретному сигналу. Тогда

$$\tau_{\max} = \Delta \cdot (n + p) \quad (2.40)$$

Выражение (2.18), изменившее вид на:

$$\mathfrak{E}_{n-d}^Q = \frac{n^*}{n \log_2 \left(\sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} \left(\sum_{\substack{a_0+a_1+\dots+a_s=i \\ a_1+2a_2+\dots+Sa_s \leq S}} \frac{i!}{a_0!a_1!\dots a_s!} \right) \right)} \cdot (Q-1)^{i+1} \quad (2.41)$$

с учетом (2.40) запишем, как

$$\mathfrak{E}_{n-d}^Q = \frac{n^*}{n \log_2 \left(\sum_{j=0}^{k-1} \sum_{i=0}^{\lfloor \frac{n^*+j}{n} \rfloor} \left(\sum_{\substack{a_0+a_1+\dots+a_p=i \\ a_1+2a_2+\dots+Sa_s \leq S}} \frac{i!}{a_0!a_1!\dots a_p!} \right) \right)} \cdot (Q-1)^{i+1} \quad (2.42)$$

Для нахождения предельного значения эффективности устремим величину блока n^* к бесконечности и найдем предел выражения (2.42).

С целью упрощения преобразований покажем, что выражение, стоящее в знаменателе, может быть заменено его максимальным членом. Для этого рассмотрим очевидное неравенство:

$$N_{\max}^Q < N < C_{p+m}^m \cdot N_{\max}^Q \quad (2.43)$$

где N – общее число реализаций над блоком n^* , а C_{p+m}^m , как показано в [28], - число целочисленных решений уравнения $a_0 + a_1 + \dots + a_p = m$, причем без дополнительных ограничений, вытекающих из другого условия $a_1 + 2a_2 + \dots + pa_p \leq S$, а m - число фронтов, при которых достигается максимальный член.

Так же, как и в (2.23) определим:

$$N_{\max}^Q = N^Q(m) = N^Q(\beta \cdot n^*) \quad (2.44)$$

где β - коэффициент, связывающий величину m и длину блока n^* .

При $n^* \rightarrow \infty$ можно записать, что

$$\lim_{n^* \rightarrow \infty} C_{p+m}^m = \lim_{n^* \rightarrow \infty} \frac{m^p}{p!} \quad (2.45)$$

и так же, как при выводе выражения предельной эффективности (2.27) записать следующее выражение, внося очевидные изменения в (2.43):

$$\frac{N^Q(\beta \cdot n^*)}{\frac{n^*}{2^k}} < \frac{N}{\frac{n^*}{2^k}} < \frac{m^p}{p!} \cdot \frac{N^Q(\beta \cdot n^*)}{\frac{n^*}{2^k}} \quad (2.46)$$

Переходя к пределу, при $n^* \rightarrow \infty$, после логарифмирования и преобразований получаем

$$\frac{\log_2 N^Q(\beta \cdot n^*)}{n^*} < \frac{\log_2 N}{n^*} < \frac{\log_2 N^Q(\beta \cdot n^*)}{n^*} \quad (2.47)$$

Что свидетельствует о правомочности замены суммы в выражении (2.42) ее максимальным членом.

В результате имеем:

$$\mathfrak{E}_{n-\delta}^Q = \lim_{n^* \rightarrow \infty} \frac{n^*}{n \log_2 \left(\sum_{\substack{a_0+a_1+\dots+a_p=m \\ a_1+2a_2+\dots+pa_p \leq S}} \frac{m!}{a_0!a_1!\dots a_p!} \right) \cdot (Q-1)^{m+1}} \quad (2.48)$$

Проведя логарифмирование, и, вычисляя предел, окончательно получаем:

$$\mathfrak{E}_{n-\delta}^{Q*} = \frac{1}{\beta n [\log_2(Q-1) + \log_2(2 - 2\beta n + \beta) - \log_2 \beta]} \quad (2.49)$$

Полученное выражение, так же, как и (2.27) является обобщенной оценкой эффективности преобразования многоуровневого непрерывно-дискретного сигнала в двоичный дискретный (цифровой), но с дополнительным ограничением,

касающимся максимальной длительности элементов в исходном сигнале от источника сообщений.

Для вычисления коэффициента β найдем максимум выражения, стоящего в знаменателе (2.49).

Вычислим первую производную.

$$\beta n [\log_2(Q-1) + \log_2(2-2\beta n + \beta) - \log_2 \beta]' = n [\log_2(Q-1) + \log_2(2-2\beta n + \beta) + \frac{\beta(1-2n)}{(2-2\beta n + \beta) \ln 2} - \log_2 \beta - \frac{1}{\ln 2}] \quad (2.50)$$

После несложных преобразований составим уравнение для определения точки экстремума:

$$\begin{aligned} & (2-2\beta n + \beta) \cdot \ln 2 \cdot \log_2(Q-1) + \ln 2 \cdot (2-2\beta n + \beta) \cdot \log_2(2-2\beta n + \beta) + \beta(1-2n) - \\ & -(2-2\beta n + \beta) \cdot \ln 2 \cdot \log_2 \beta - (2-2\beta n + \beta) = (2-2\beta n + \beta) \cdot \ln 2 \cdot \log_2(Q-1) + \ln 2 \cdot \\ & \cdot (2-2\beta n + \beta) \cdot \log_2(2-2\beta n + \beta) - (2-2\beta n + \beta) \cdot \ln 2 \cdot \log_2 \beta - 2 = 0 \end{aligned} \quad (2.51)$$

После некоторого упрощения

$$(2-2\beta n + \beta) \cdot \log_2(Q-1) + (2-2\beta n + \beta) \cdot \log_2(2-2\beta n + \beta) - (2-2\beta n + \beta) \cdot \log_2 \beta = \frac{2}{\ln 2} \quad (2.52)$$

Далее, решая это уравнение при заданных n и Q , находим коэффициент β . Подставляя найденное значение в (2.49) получаем значение предельной эффективности.

Приведем пример расчета в продолжение Примера 1.

Пример 2

Для $Q = 4$ и $n = 4$, находим $\beta = 0,175$, при котором левая часть выражения (2.51) равна 2,892, тогда как правая часть этого же выражения равна 2,886. Полагаем что точность совпадения до третьего знака достаточная и подставляем найденное значение в (2.41). В итоге получаем значение предельной эффективности $\mathfrak{E}_{n-\delta}^{Q*} = \mathfrak{E}_{n-\delta}^{4*} = 0,383$.

Аналогично проводятся расчеты для других значений Q и n , что отображено в виде соответствующей зависимости на рисунке 2.8 построенный по рассчитанным значениям при $Q = 2, 4, 8, 16$ и $n = 2, 4, 12, 24$.

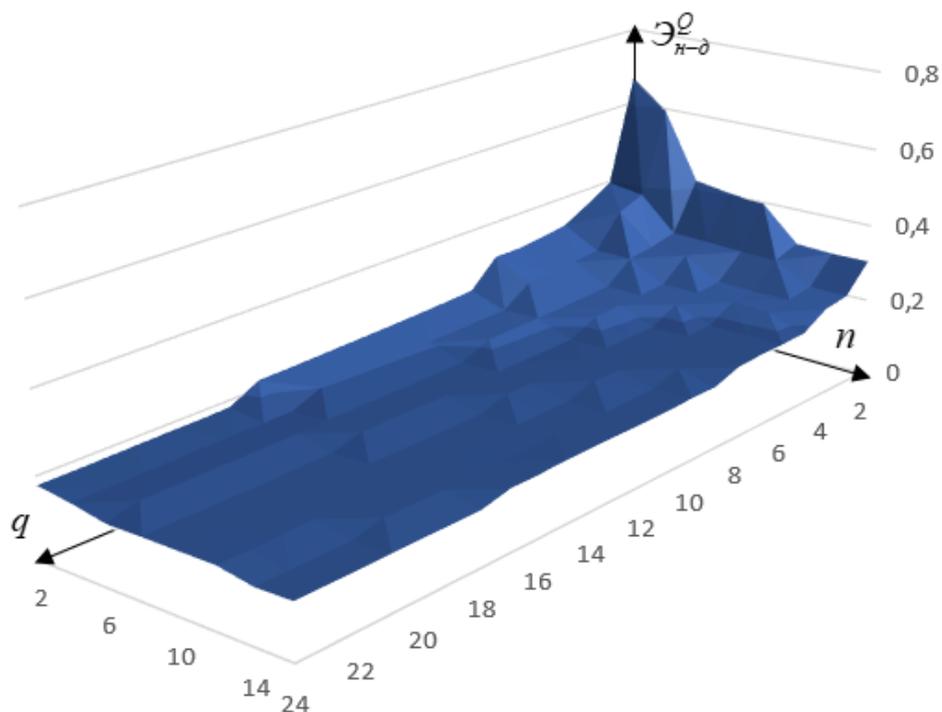


Рисунок 2.8 - Зависимости предельной эффективности преобразования Q-ичного непрерывно-дискретного сигнала в цифровой при ограничении минимальной и максимальной длительности элементов исходного сигнала.

Рассмотренные выше случаи эквивалентны ситуации, когда все реализации непрерывно-дискретного сигнала над блоком n^* равновероятны, что соответствует максимальной энтропии дискретного источника. Таким образом, найденные значения предельной эффективности определяют нижнюю предельную границу. Во всех других ситуациях энтропия будет меньше, а значит, эффективность преобразования будет больше. При этом конкретное значение эффективности будет зависеть от распределения вероятностей и его следует рассчитывать для каждого конкретного случая.

Исследуем эти вопросы более подробно.

Исходя из (2.4-2.10), формально можно записать выражение для эффективности Q-ичного непрерывно-дискретного сигнала

$$\mathcal{E}_{n-\delta}^Q(\varepsilon \leq \delta_{\max}) = -\frac{n^*}{n \cdot H_{n-\delta}^Q(\varepsilon \leq \delta_{\max})} = -\frac{n^*}{n \cdot \sum_{i=0}^N p_i \cdot \log_2 p_i} \quad (2.52)$$

где $H_{n-d}^Q(\varepsilon \leq \delta_{\max}) = \sum_{i=0}^N p_i \cdot \log_2 p_i$ – энтродия источника, а p_i – соответствующие значения вероятностей реализаций исходного сигнала над блоком n^* .

Зная распределение вероятностей в виде соответствующей модели, по формуле (2.52) можно рассчитать эффективность преобразования, переходя к пределу при $n^* \rightarrow \infty$ и, заменяя множество различных реализаций на соответствующее высоковероятное множество, что позволит перейти от суммирования множества членов к максимальному члену, представителю этого высоковероятного множества. При этом данное преобразование должно решаться в каждом конкретном случае, исходя из конкретной модели источника. Получаемые в этих условиях значения эффективности будут всегда выше по сравнению с границами, установленными формулами (2.27) и (2.49).

2.4 Выводы по разделу

Предложенная методика оценки эффективности преобразования различных сигналов в цифровой позволяет получить количественные оценки энтропии источников сигналов с различной структурой.

Показана зависимость эффективности преобразования сигналов в цифровой при различных значениях точности отображения в виде введенного критерия качества, оцениваемого по величине искажений элементов исходного сигнала.

Полученные аналитические выражения предельной эффективности преобразования многоуровневого непрерывно-дискретного сигнала в цифровой показывают количественные значения данного преобразования в зависимости от числа уровней и точности отображения элементов исходного сигнала.

Полученные аналитические выражения оценки предельной эффективности преобразования различных сигналов в цифровой можно использовать для оценки эффективности в случаях неравновероятного распределения появления различных

элементов в исходном сообщении. При этом эффективность преобразования возрастает, поскольку уменьшается энтропия источника.

Предельная эффективность преобразования различных сигналов в цифровой достигается при длине преобразуемого сигнала, стремящегося к бесконечности. Реально полученные значения эффективности при ограниченной длительности преобразуемого сигнала отличается от теоретически достижимой эффективности на 7-10%. Точное непреодолимое отличие реальной эффективности от теоретически достижимой, в условиях ограниченности длительности исходного сигнала, зависит от точности отображения и вида сигнала.

3. РАЗРАБОТКА МЕТОДА ИССЛЕДОВАНИЯ ВОЗНИКНОВЕНИЯ ДОПОЛНИТЕЛЬНЫХ ИСКАЖЕНИЙ ПРИ НАЛИЧИИ ОШИБОК В ЦИФРОВОМ КАНАЛЕ И ПРЕОБРАЗОВАНИИ РАЗЛИЧНЫХ МЕТРИЧЕСКИХ ПРОСТРАНСТВ

3.1 Исходные данные и определения

Согласно [44,45] метрическим пространством называется непустое множество, в котором между любой парой элементов, обладающих установленными свойствами, характерных для данного множества, определено расстояние, называемое метрикой.

Метрическое пространство принято обозначать парой (X, d) , где X - множество, а d - числовая функция, которая определена на декартовом произведении $X \times X$, принимающая значения на множестве вещественных чисел. При этом должны выполняться следующие аксиомы:

$$d(x, y) = 0 \Leftrightarrow x = y \text{ - аксиома тождества.}$$

$$d(x, y) = d(y, x) \text{ - аксиома симметрии.}$$

$d(x, z) \leq d(x, y) + d(y, z)$ - аксиома треугольника или неравенство треугольника.

Само множество X называют подлежащим множеством метрического пространства, а элементы множества X называются точками метрического пространства [44,45,46].

Функция d называется метрикой.

Как правило, расстояние между точками x и y в метрическом пространстве M обозначается через $d(x, y)$, что и будет использовано в дальнейшем, хотя в ряде источников используется - $\rho(x, y)$. При этом согласно введенным в Разделе 1 обозначениям аналогом $d(x, y)$ будет $d(a, b)$, либо $d(a_i, a_j)$ или $d(b_i, b_j)$.

Сигналы от источника сообщений могут быть, как указано в Разделе 1, различны. При этом для их описания используются соответствующие модели, позволяющие представить порождаемые сигналы в различных метрических пространствах, адекватных типу рассматриваемых сигналов. Так, например, для описания аналоговых сигналов используется пространство Евклида, тогда как для описания дискретного цифрового сигнала – пространство Хэмминга. При этом очевидно, что отображение сигналов вначале в пространстве Евклида, а затем в пространстве Хэмминга соответствует нелинейному преобразованию, частью которого является необходимость соответствия мощностей соответствующих множеств при условии однозначного (взаимно однозначного) отображения и при введенном критерии качества. Используя терминологию и определения из [44-46], названное выше преобразование пространств соответствует биекции пространств с исполнением условий инъективности и сюръективности. Однако при этом не выполняется условие изометрии.

Исходя из этого, обратное преобразование также является нелинейным, что выражается в том, что если в одном пространстве, согласно введенной метрике, расстояние $d(a_1, a_2) < d(a_1, a_3)$, то в другом пространстве, согласно иной введенной метрике, может быть как $d(b_1, b_2) > d(b_1, b_3)$, так и наоборот $d(b_1, b_2) < d(b_1, b_3)$. При этом оба пространства являются метрическими и для каждого из них выполняются аксиомы тождества, симметрии и треугольника.

Пример, поясняющий данное положение, показан на рисунке 3.1, где расстояние между точками a_1, a_2 и a_1, a_3 можно определить на основе их геометрического расположения и $d(a_1, a_2) > d(a_1, a_3)$. В тоже время, если этим точкам поставить в соответствие элементы пространства Хэмминга (b_1, b_2, b_3) , то в зависимости от вида этого соответствия, получаем разные результаты, а именно, когда $b_1 = (1000), b_2 = (0111)$, а $b_3 = (0000)$ имеем: $d(b_1, b_2) > d(b_1, b_3)$, однако при $b_1 = (1000), b_2 = (0000)$, а $b_3 = (0111)$ – наоборот $d(b_1, b_2) < d(b_1, b_3)$.

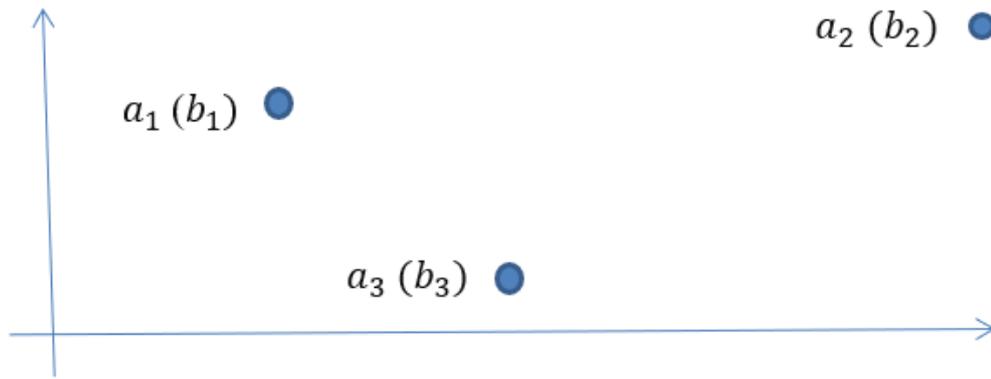


Рисунок 3.1 - Точки пространства

Дискретные сообщения от источника $a_i \in A$ появляются с некоторой вероятностью p_i , что принято обозначать как ансамбль сообщений $\{a_i, p_i\}$. Согласно этому можно определить и ансамбль сигналов $\{b_i, p_i\}$.

В результате действия помех в канале связи один сигнал может быть преобразован в другой, т.е. произойдет ошибка и $b_i \rightarrow b_j$ или, записывая иначе $b_j = b_i + e$, где e - вектор ошибок.

Ансамбль $\{b_i, p_i\}$ определен в пространстве Хэмминга, тогда как ансамбль $\{a_i, p_i\}$ - в ином пространстве, чаще всего в пространстве Евклида. Поэтому возникшая в сигнале ошибка, когда $b_i \rightarrow b_j$, в сообщении $a_i \rightarrow a_j$ отобразится в виде искажения, величину которого обозначим через z_{ij} .

Вероятности преобразования типа $b_i \rightarrow b_j$ будем обозначать p_{ij} . Тогда в целом этот процесс можно характеризовать матрицей переходных вероятностей:

$$P_{ij} = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{pmatrix} \quad (3.1)$$

Согласно (3.1) определяются вероятности трансформаций $b_i \rightarrow b_j$, возникающих из-за ошибок в дискретном сигнале. При этом в восстанавливаемом

сообщении появляются искажения, который можно количественно оценить с помощью матрицы искажений (погрешностей)

$$Z_{ij} = \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \dots & \dots & \dots & \dots \\ z_{n1} & z_{n2} & \dots & z_{nn} \end{pmatrix} \quad (3.2)$$

Очевидно, что формула (3.2) может быть преобразована к виду:

$$Z_{ij} = \begin{pmatrix} 0 & z_{12} & \dots & z_{1n} \\ z_{21} & 0 & \dots & z_{2n} \\ \dots & \dots & \dots & \dots \\ z_{n1} & z_{n2} & \dots & 0 \end{pmatrix} \quad (3.3)$$

При аналитическом описании сообщений, как правило, расстояние между сообщениями в определенном метрическом пространстве соответствует величине искажения z_{ij} при возникновении трансформации вида $b_i \rightarrow b_j$ и далее $a_i \rightarrow a_j$, что, собственно говоря, и определяется введенной метрикой. Однако в общем случае это может быть и не так, что следует каждый раз оговаривать отдельно. В этом случае следует пересмотреть условия определения метрики, либо пространство не будет отвечать в полной мере аксиомам метрического пространства и его целесообразно рассматривать как топологическое пространство.

В тоже время между дискретными сигналами b_i и b_j можно точно определить расстояние $d(b_i, b_j)$, которое удобно отобразить в виде следующей матрицы кодовых расстояний:

$$D_{ij} = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{pmatrix} \quad (3.4)$$

Или после очевидного упрощения:

$$D_{ij} = \begin{pmatrix} 0 & d_{12} & \dots & d_{1n} \\ d_{21} & 0 & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & \dots & 0 \end{pmatrix} \quad (3.5)$$

Матрица кодовых расстояний фактически предсказывает вероятность возможных ошибок из-за действия помех в канале связи, исходя из общей логики того, что чем меньше кодовое расстояние, тем больше вероятность того, что помеха трансформирует один сигнал в другой, хотя в общем случае это не верно. Однако, очевидно, что при биекции пространств A и B следует стремиться к тому, чтобы более вероятные трансформации $b_i \rightarrow b_j$ приводили к наименьшим искажениям вида $a_i \rightarrow a_j$.

При общей постановке данной задачи следует определить целевую функцию, а именно, какого результата следует добиваться в плане минимизации искажений при биекции пространств.

Выберем таким минимизируемым параметром среднее значение искажений при трансформациях $a_i \rightarrow a_j$, возникающих из-за ошибок вида $b_i \rightarrow b_j$.

При известной матрице погрешностей (3.3) и матрице переходных вероятностей можно определить матрицу – столбец, элементами которой будут средние значения искажений для каждого исходного значения сообщения a_i

$$\bar{Z}_i = \begin{pmatrix} M[a_1] \\ M[a_2] \\ \dots \\ M[a_n] \end{pmatrix}, \quad (3.6)$$

где $M[a_i] = \sum_{j=1}^n p_{ij} \cdot z_{ij}$.

Среднее значение искажений с учетом вероятностей появления сообщений от источника равно

$$\bar{Z} = P_i \times \bar{Z}_i = \sum_{i=1}^n p_i \cdot \left(\sum_{j=1}^n p_{ij} \cdot z_{ij} \right) \quad (3.7)$$

где $P_i = (p_1 p_2 \dots p_n)$ – матрица - строка, отражающая распределение вероятностей появления сообщений a_i от источника и, соответственно, сопоставленных с ними сигналов b_i .

3.2. Особенности расстояний в пространстве Хэмминга

Матрица (3.5) определяет в общем виде расстояния, будем их называть кодовыми расстояниями, между любой парой элементов b_i и b_j дискретного множества B . А сами элементы будем называть кодовыми комбинациями некоторого кода по основанию 2.

Будем также полагать, что каждая кодовая комбинация состоит из n двоичных символов.

В зависимости от способа сопоставления элементов $a_i \in A$ элементам $b_i \in B$, т.е. способов биекции пространств A и B , матрица (3.5) приобретает различное расположение своих элементов. При этом ни число этих элементов, ни их значения не изменяются [84-86].

Рассмотрим в качестве исходного варианта размещение элементов матрицы (3.5) для кодовых комбинаций натурального или, как еще говорят, взвешенного кода, когда последовательность кодовых комбинаций соответствует нарастающей численной последовательности от 0 до $N = (2^n - 1)$.

Согласно первой аксиоме тождества в главной диагонали матрицы (3.5) располагаются 0. А согласно второй аксиоме - аксиоме симметрии можно отметить и соответствующую симметрию всех элементов матрицы (3.5) относительно главной диагонали. Данное свойство доказывает, что для любого метрического пространства для определения всех элементов матрицы вида (3.5) необходимо определить только половину элементов матрицы (3.5), поскольку остальные могут быть достроены, исходя из принципа симметрии.

При этом очевидно, что расстояние Хэмминга всегда соответствует условию, что $0 \leq d(x, y) \leq n$.

Рассмотрим данный пример и построим последовательно матрицы кодовых расстояний $D_n(d)$, в которой укажем кодовые расстояния $0 \leq d(x, y) \leq n$ между всеми возможными парами кодовых комбинаций, при разных значениях n :

При $n = 1$ имеем:

$$D_1(d) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (3.8)$$

а для $n = 2$ –

$$D_2(d) = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} \quad (3.9)$$

Рассматривая структуру матриц (3.8) и (3.9) для натурального кода, замечаем, что с последовательным ростом n добавляется единица в старшем разряде, в результате чего можно сформировать следующее рекуррентное соотношение:

$$D_n(d) = \begin{pmatrix} D_{n-1}(d) & D_{n-1}(d) + J \\ D_{n-1}(d) + J & D_{n-1}(d) \end{pmatrix}, \quad (3.10)$$

где J – матрица, состоящая из единиц, и имеющая размерность, совпадающую с размерностью матрицы $D_{n-1}(d)$.

Матрица (3.10) обладает рядом очевидных свойств. А именно: в главной диагонали стоят 0, а в побочной – значения $d = n$.

Все строки и все столбцы этой матрицы состоят из одних и тех же чисел, но каждый раз в ином порядке, определяемом соотношением (3.10). При этом справедливо, что порядок чисел в i – ой строке совпадает с порядком чисел в j – ом столбце тогда и только тогда, когда $i = j$. Помимо этих свойств имеется и ряд других, которые также вытекают из соотношения (3.10). Например, это могут быть определенные виды симметрии, в том числе обратные симметрии, относительно различных секущих, примером чего может служить то, что первая строчка инверсна последней, вторая – предпоследней и т.д.

Несложно показать, что в каждой строке имеется C_n^d чисел, отражающих кодовое расстояние равное d в комбинации из n элементов.

Для всех чисел матрицы (3.10) справедливы соотношения:

$$N_n(0) = N_n(n), N_n(1) = N_n(n-1), \dots, N_n(i) = N_n(n-i), \dots$$

$$N_n(d) = 2 \cdot [N_{n-1}(d) + N_{n-1}(d-1)] \text{ или} \quad (3.11)$$

$$N_n(d) = C_n^d \cdot 2^n,$$

где $N_n(d)$ - число комбинаций из n элементов, имеющих кодовое расстояние равное d .

Рассмотрим пример, когда $n = 5$. Запишем матрицу (3.10)

Не уменьшая общности рассуждений, обратимся к первой строке. В ней приводятся кодовые расстояния относительно нулевой кодовой комбинации.

Отметим, что все строчки кодовых расстояний (3.12), а в общем случае матрицы (3.10), равноценны по количеству кодовых расстояний того или иного значения и отличаются друг от друга только порядком следования кодовых расстояний в матрице $D_n(d)$. Пример, показанный в (3.12) наглядно демонстрирует это. Отметим также еще одну удобную закономерность.

Если известна первая строчка, указывающая на кодовые расстояния относительно нулевой кодовой комбинации, то можно последовательно построить все остальные строчки, в которых будет отражен порядок кодовых расстояний относительно всех других кодовых комбинаций.

Попутно отметим, что первая строчка может также рассматриваться, как строчка, состоящая из весов кодовых комбинаций. Количество единиц в двоичной кодовой комбинации будем называть весом кодовой комбинации, и обозначать через (w) .

Вторая строчка строится путем «диагональных смещений» пар относительно первой строки. В первой строке имеем пару (0 1), значит, во второй строке будет (1 0). Далее следующая пара в первой строке (1 2) и соответственно во второй – (2 1).

Третья пара в первой строке (1 2) и четвертая пара (2 3). Соответственно во второй строке имеем (2 1) и (3 2). И т.д.

Третья и четвертая строка строятся путем «диагональных смещений» уже не пар, а четверок цифр из первой и второй строки.

Действительно, из первой и второй строки имеем первую четверку $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, а затем вторую четверку $\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$. Путем «диагональных смещений» соответственно имеем первую четверку третьей и четвертой строк: $\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$. А затем вторую четверку третьей и четвертой строк: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ и т.д. Затем строятся «диагональные смещения» уже из шестнадцати элементов матрицы кодовых расстояний, когда следует взять для построения четыре строки, и т.д.

Правило такого построения вытекает из матрицы (3.10), а пример (3.12) наглядно демонстрирует это.

Матрица кодовых расстояний определяет данный параметр между любыми парами кодовых комбинаций. Но если взять не две, а большее количество кодовых комбинаций, то для их характеристики часто используют показатель минимального кодового расстояния (d_{\min}), равный наименьшему кодовому расстоянию среди рассматриваемых кодовых комбинаций. (Попутно заметим, что если среди этих комбинаций есть нулевая кодовая комбинация, то минимальное кодовое расстояние (d_{\min}) будет равно минимальному весу (w) кодовой комбинации из этой группы).

(3.12)

При сравнении какой-либо j -ой кодовой комбинации с нулевой кодовой комбинацией ее вес (w) будет соответствовать кодовому расстоянию (d_{0j}) по отношению к нулевой кодовой комбинации, что наглядно демонстрирует первая

строка матрицы кодовых расстояний (3.12). Данные этой строки позволяют определить спектр весов кодовых комбинаций, приведенных в матрице кодовых расстояний $D_n(d)$. Таким образом, спектр весов указывает на количество кодовых комбинаций, имеющих тот или иной вес, что соответствует кодовому расстоянию по отношению к нулевой кодовой комбинации. Для других строчек матрицы кодовых расстояний, спектр кодовых расстояний будет таким же. И в этом смысле все строчки матрицы (3.12) эквивалентны, а спектр весов инвариантен по отношению к любой кодовой комбинации.

Отметим также, что кодовое расстояние (d_{ij}) – это число, определяемое, как правило, в десятичной системе счисления, равное весу кодовой комбинации, полученной в результате сложения по mod 2 двоичных кодовых комбинаций, отображающих число i и число j .

Для примера положим $i=17$, а $j=13$ в десятичной системе счисления и вычислим $d_{ij} = d_{17_{13}}$. Для этого переведем значения i и j в двоичную форму и найдем $i \oplus j = 10001 \oplus 01101 = 11100$. Полученное двоичное число имеет вес равный трем, что определяет значение $d_{ij} = d_{17_{13}} = 3$. Для проверки обратимся к матрице, представленной в (3.12), и убедимся в истинности полученного результата.

Как уже отмечалось в Разделе 1 число различных вариантов сопоставления кодовых комбинаций дискретного сигнала сообщениям от источника равно $(2^n)!$, где n - число двоичных символов в кодовой комбинации.

Каждое такое сопоставление – это определенный вид биекции пространств A и B .

Рассмотренное выше пространство B соответствует взвешенному или натуральному коду. Согласно этому назовем это пространство также, т.е. натуральное или взвешенное.

Всякое другое пространство, которые будем называть пространством B другого типа, может быть получено из данного пространства путем

соответствующих перестановок его элементов. При этом эти элементы определяются по рекуррентной формуле (3.10).

Как уже отмечалось, элементы первой строчки определяют элементы всех остальных строчек. При этом первая строчка, очевидно, формируется также аналогично формуле (3.10), когда $d_{00} = 0$, далее согласно (3.10) $d_{01} = d_{00} + 1 = 1$, $d_{02} = d_{00} + 1 = 1$, $d_{03} = d_{01} + 1 = 2$, $d_{04} = d_{00} + 1 = 1$, $d_{05} = d_{01} + 1 = 2$, $d_{06} = d_{02} + 1 = 2$, $d_{07} = d_{03} + 1 = 3$ и т.д.

Первая строчка кодовых расстояний соответствует также весу кодовых комбинаций, нарастающих по своему значению от 0 до $(2^n - 1)$.

При изменении типа пространства B изменяется порядок размещения элементов. Однако их число и их значения остаются неизменными.

Рассмотрим еще один тип часто встречающегося пространства B , а именно, пространство B , соответствующее коду Грэя.

В качестве примера приведем матрицу кодовых расстояний аналогичную матрице (3.12) для $n = 5$.

Первая строчка этой матрицы строится согласно логике кода Грэя, когда с увеличением длины кодовой комбинации n на единицу, число различных кодовых комбинаций возрастает вдвое, и при этом вновь формируемые кодовые комбинации могут быть получены из предыдущих, когда к новому старшему разряду, равному единице, справа приписываются ранее построенные комбинации, но в инверсном порядке.

В результате первая строчка приобретает вид: $d_{00}^G = 0, d_{01}^G = d_{01}, d_{02}^G = d_{03}, d_{03}^G = d_{02}, d_{04}^G = d_{06}, d_{05}^G = d_{07}, d_{06}^G = d_{05}, d_{07}^G = d_{04}, d_{08}^G = d_{012}, d_{09}^G = d_{013}, d_{010}^G = d_{015}, d_{011}^G = d_{014}, d_{012}^G = d_{010}, d_{013}^G = d_{011}, d_{014}^G = d_{09}, d_{015}^G = d_{08}$ и т.д.

(d_{00}^G – это кодовые расстояния, представленные в матрице (3.13) для кода Грэя, а d_{06} – это кодовые расстояния из матрицы (3.12) для взвешенного кода)

Дальнейшие построения аналогичны рассмотренному выше, а именно: вторая строчка получается путем диагональных смещений элементов первой строки. Затем 3-ья и 4-ая строчки получаются путем диагональных смещений

соответствующих четверок, взятых из первой и второй строк. 5-ая, 6-ая, 7-ая и 8-ая строчки формируются из элементов 1-ой, 2-ой, 3-ей и 4-ой строчек путем диагональных смещений, соответствующих 16 элементов так, как это было указано выше.

$$D_5^\Gamma(d) = \begin{pmatrix} 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 4 & 5 & 4 & 3 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 \\ 1 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 5 & 4 & 3 & 4 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 4 & 3 & 4 & 5 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 \\ 1 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 3 & 4 & 5 & 4 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 \\ 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 4 & 5 & 4 & 3 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 \\ 3 & 2 & 1 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 5 & 4 & 3 & 4 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 4 & 3 & 4 & 5 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 \\ 1 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 3 & 4 & 5 & 4 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 \\ 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 4 & 5 & 4 & 3 \\ 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 5 & 4 & 3 & 4 \\ 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 4 & 3 & 4 & 5 \\ 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 3 & 4 & 5 & 4 \\ 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 4 & 5 & 4 & 3 & 2 & 3 & 4 & 3 \\ 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 5 & 4 & 3 & 4 \\ 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 \\ 3 & 4 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 \\ 2 & 3 & 4 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 \\ 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 5 & 4 & 3 & 4 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 2 \\ 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 4 & 3 & 4 & 5 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 \\ 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 3 & 4 & 5 & 4 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 1 & 0 \end{pmatrix} \quad (3.13)$$

Сравнивая матрицы (3.12) и (3.13), отмечаем, что они состоят из одинакового числа элементов, определяющих соответствующие кодовые расстояния. Однако расположение их друг относительно друга имеет значительные отличия.

Если предположить, что вероятности перехода из одного состояния в другое одинаковы, т.е. матрица (3.1) состоит из одинаковых элементов, то типы пространств B при биекции пространств A и B не влияют на результат возникновения тех или иных искажений в элементах $a_i \in A$ при возникновении ошибок в элементах $b_i \in B$.

В противном случае, т.е. когда матрица переходных вероятностей (3.1) содержит различные элементы, выбор типа матрицы B может оказать существенное влияние при биекции с пространством A .

3.3. Особенности биекции пространств при передаче сообщений от источника к получателю

Дополним блок-схему, показанную на рисунке 1.2, еще одним функциональным элементом, а именно, скремблером, «выравнивающим» статистические особенности исходного сигнала, отметив также, что в современных системах связи к этому элементу зачастую добавляются еще и функции шифрования [47].

Будем обозначать через $a_i \in A$ элементы от источника, через $b_i \in B$ - элементы после кодера источника, через $c_i \in C$ - элементы после помехоустойчивого кодирования, через $g_i \in G$ - элементы после скремблирования и, наконец, через $q_i \in Q$ - элементы после модуляции.

Через A , B , C , G и Q будем обозначать соответствующие множества, биекцию которых и будем исследовать с учетом влияния типа биекции на конечный результат оценки качества передаваемого сообщения.

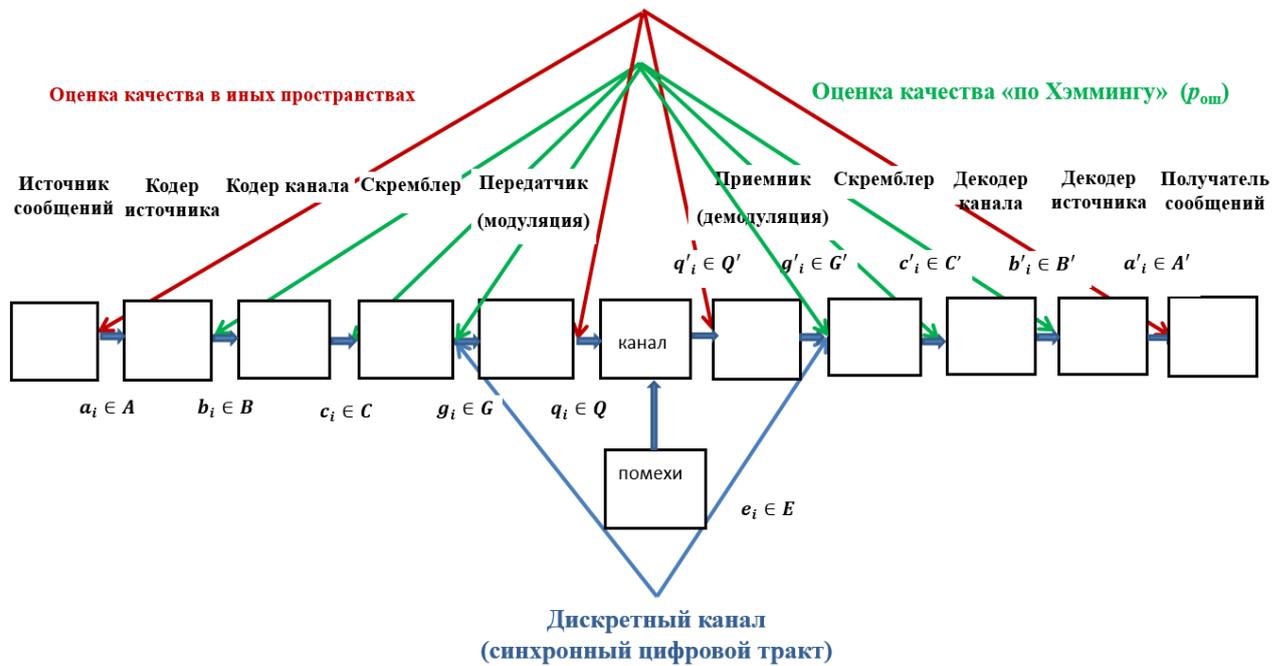


Рисунок 3.2 - Обобщенная блок-схема передачи сообщений от источника к получателю

3.3.1. Биекция пространств A и B

Рассматривая биекцию пространств A и B , будем исследовать ситуацию, когда пространство A является пространством Евклида, а пространство B - пространством Хэмминга.

Особенности кодовых расстояний в пространстве Хэмминга были рассмотрены выше, а в пространстве A (Евклида) рассмотрим модель, когда элементами $b_i \in B$ n – мерного множества B являются точки с определенными координатами, а расстояние между ними эквивалентно погрешности, возникающей, если один элемент множества A неверно восстанавливается на приеме, как какой-либо другой элемент этого множества.

Для описания этого достаточно задать матрицу погрешностей, определенную в (3.3), что не всегда может совпадать с геометрическими представлениями точек на плоскости.

Рассмотрим в качестве примера 8 элементов 3-х мерного пространства A с координатами $a_0 = (0,0,0)$, $a_1 = (3,0,0)$, $a_2 = (3,4,0)$, $a_3 = (0,4,0)$, $a_4 = (1,2,5)$, $a_5 = (1,0,5)$, $a_6 = (6,0,5)$, $a_7 = (6,5,6)$. Наглядно пример показан на рисунке 3.3.

Вычислим для данного примера матрицу погрешностей, сопоставляя их значения с геометрическим расстоянием между точками 3-х мерного пространства.

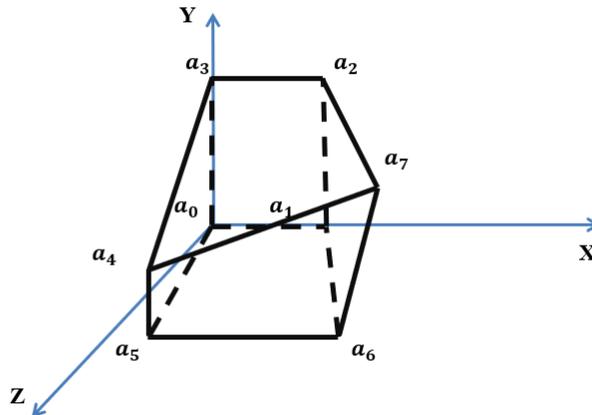


Рисунок 3.3 - Пример биекции пространства A и B

$$Z_{ij=} \begin{pmatrix} 0,00 & 3,00 & 5,00 & 4,00 & 5,48 & 5,10 & 7,81 & 9,85 \\ 3,00 & 0,00 & 4,00 & 5,00 & 5,74 & 5,39 & 5,83 & 8,37 \\ 5,00 & 4,00 & 0,00 & 3,00 & 5,74 & 6,71 & 7,07 & 6,78 \\ 4,00 & 5,00 & 3,00 & 0,00 & 5,48 & 6,48 & 8,77 & 8,54 \\ 5,48 & 5,74 & 5,74 & 5,48 & 0,00 & 2,00 & 5,39 & 5,92 \\ 5,10 & 5,39 & 6,71 & 6,48 & 2,00 & 0,00 & 5,00 & 7,14 \\ 7,81 & 5,89 & 7,07 & 8,77 & 5,39 & 5,00 & 0,00 & 5,10 \\ 9,85 & 8,37 & 6,78 & 8,54 & 5,92 & 7,14 & 5,10 & 0,00 \end{pmatrix} \quad (3.14)$$

Для наглядности осуществления биекции пространства A с пространством B различного типа приведем матрицы кодовых расстояний для взвешенного кода

$$D_3^B = \begin{pmatrix} 0 & 1 & 1 & 2 & 1 & 2 & 2 & 3 \\ 1 & 0 & 2 & 1 & 2 & 1 & 3 & 2 \\ 1 & 2 & 0 & 1 & 2 & 3 & 1 & 2 \\ 2 & 1 & 1 & 0 & 3 & 2 & 2 & 1 \\ 1 & 2 & 2 & 3 & 0 & 1 & 1 & 2 \\ 2 & 1 & 3 & 2 & 1 & 0 & 2 & 1 \\ 2 & 3 & 1 & 2 & 1 & 2 & 0 & 1 \\ 3 & 2 & 2 & 1 & 2 & 1 & 1 & 0 \end{pmatrix} \quad (3.15)$$

и для кода Грэя

$$D_3^r = \begin{pmatrix} 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 \\ 1 & 0 & 1 & 2 & 3 & 2 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 2 & 1 & 0 & 1 & 2 & 3 & 2 \\ 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 \\ 1 & 2 & 3 & 2 & 1 & 2 & 1 & 0 \end{pmatrix} \quad (3.16)$$

Осуществив биекцию пространства A с пространством B , построенного для взвешенного кода, и, осуществив биекцию пространства A с пространством A , построенного для кода Грэя, получаем отличающиеся варианты отображения элементов $a_i \in A$ в виде элементов $b_i \in B$.

Поскольку в процессе дальнейшей передачи сообщений помехи в канале связи приводят к ошибкам в элементах $b_i \in B$, то важно исследовать влияние этих ошибок на возникновение дополнительных искажений в элементах $a_i \in A$.

Рассмотрим данный процесс на примере двух пространств B , описываемых соответственно матрицами (3.15) и (3.16), при исходном пространстве A , заданном матрицей искажений (3.14).

Построим таблицу переходов одной комбинации b_i в другую b_j , при возникновении всевозможных ошибок, задаваемых вектором e (таблица 3.1). А далее с учетом полученных новых значений элементов $b_j = b_i \oplus e$, запишем в таблице 3.2 соответствующие искажения, взятые из матрицы 3.14.

Для построения таблиц 3.1 и 3.2 применялось следующее соответствие:

$$a_0 = (0,0,0) \rightarrow b_0 = 000, a_1 = (3,0,0) \rightarrow b_1 = 001, a_2 = (3,4,0) \rightarrow b_2 = 010, a_3 = (0,4,0) \rightarrow b_3 = 011, \\ a_4 = (1,2,5) \rightarrow b_4 = 100, a_5 = (1,0,5) \rightarrow b_5 = 101, a_6 = (6,0,5) \rightarrow b_6 = 110, a_7 = (6,5,6) \rightarrow b_7 = 111$$

Проведем аналогичные построения для кода Грэя, учитывая, что в данном случае между элементами исходного сообщения множества A и кодовыми комбинациями множества B будут установлены следующие соответствия:

$$a_0 = (0,0,0) \rightarrow b_0 = 000, a_1 = (3,0,0) \rightarrow b_1 = 001, a_2 = (3,4,0) \rightarrow b_2 = 011, a_3 = (0,4,0) \rightarrow b_3 = 001, \\ a_4 = (1,2,5) \rightarrow b_4 = 101, a_5 = (1,0,5) \rightarrow b_5 = 111, a_6 = (6,0,5) \rightarrow b_6 = 101, a_7 = (6,5,6) \rightarrow b_7 = 100$$

В результате получаем следующую таблицу 3.3, в которой приведены измененные кодовые комбинации в результате воздействия всевозможных ошибок.

Таблица 3.1 – Кодовые комбинации b_j после воздействия различных ошибок e

Исходные комбинации \ Ошибки	000	001	010	011	100	101	110	111
$e = 001$	001	000	011	010	101	100	111	110
$e = 010$	010	011	000	001	110	111	100	101
$e = 011$	011	010	001	000	111	110	101	100
$e = 100$	100	101	110	111	000	001	010	011
$e = 101$	101	100	111	110	001	000	011	010
$e = 110$	110	111	100	101	010	011	000	001
$e = 111$	111	110	101	100	011	010	001	000

Таблица 3.2 – Дополнительные искажения из-за ошибок e

Исходные комбинации \ Ошибки	000	001	010	011	100	101	110	111
$e = 001$	3,00	3,00	3,00	3,00	2,00	2,00	5,10	5,10
$e = 010$	5,00	5,00	5,00	5,00	5,39	7,14	5,39	7,14
$e = 011$	4,00	4,00	4,00	4,00	5,92	5,00	5,00	5,92
$e = 100$	5,48	5,39	7,07	8,54	5,48	5,39	7,07	8,54
$e = 101$	5,10	5,74	6,78	8,77	5,74	5,10	8,77	6,78
$e = 110$	7,81	8,37	5,74	6,48	5,74	6,48	7,81	8,37
$e = 111$	9,85	5,83	6,71	5,48	5,48	6,71	5,89	9,85

Далее, воспользовавшись матрицей 3.14 получаем

Таблица 3.3- Кодовые комбинации b_j после воздействия различных ошибок e

Исходные комбинации \ Ошибки	000	001	011	010	110	111	101	100
$e = 001$	001	000	010	011	111	110	100	101
$e = 010$	010	011	001	000	100	101	111	110
$e = 011$	011	010	000	001	101	100	110	111
$e = 100$	100	101	111	110	010	011	001	000
$e = 101$	101	100	110	111	011	010	000	001
$e = 110$	110	111	101	100	000	001	011	010
$e = 111$	111	110	100	101	001	000	010	011

Воспользовавшись данными из матрицы 3.14 с учетом нового способа соответствия, получаем таблицу искажений (таблица 3.4)

Таблица 3.4- Дополнительные искажения из-за ошибок e

Исходные комбинации \ Ошибки	000	001	011	010	110	111	101	100
$e = 001$	3,00	3,00	3,00	3,00	2,00	2,00	5,10	5,10
$e = 010$	4,00	4,00	4,00	4,00	5,92	5,00	5,00	5,92
$e = 011$	5,00	5,00	5,00	5,00	5,39	7,14	5,39	7,14
$e = 100$	9,85	5,83	6,71	5,48	5,48	6,71	5,89	9,85
$e = 101$	7,81	8,37	5,74	6,48	5,74	6,48	7,81	8,37
$e = 110$	5,48	5,39	7,07	8,54	5,48	5,39	7,07	8,54
$e = 111$	5,10	5,74	6,78	8,77	5,74	5,10	8,77	6,78

Заметим, что сумма искажений, приведенная в таблице 3.2 и 3.4, одинакова.

Это означает, что если переходы элементов $b_i \in B$ в другие элементы $b_j \in B$ из-за

действия помех и возникающих ошибок, будут происходить с одинаковыми вероятностями, то независимо от типа биекции пространства A и пространства B , итоговый результат возникающих дополнительных искажений будет один и тот же. В частности для данного примера эта величина суммарно составит 327,38 условных единиц трехмерного пространства A или при персчете на одну ошибку в одной комбинации 3,41 условных единицы.

Однако в реальности однократные, двух кратные и трехкратные ошибки возникают с разными вероятностями. В результате этого одни типы биекции пространства A и пространства B оказываются предпочтительней других.

Если предположить, что вероятность однократных ошибок много больше вероятности двукратных и трехкратных ошибок, и, следовательно, ими можно пренебречь, а также допустить, что вероятности передачи различных элементов $b_i \in B$ одинаковы, то суммарные искажения для взвешенного кода составят 124,22 условных единицы, а для кода Грэя – 119,78 условных единиц, что в персчете на одну ошибку и для одной комбинации $b_i \in B$ соответственно составит – 5,18 и 4,99 условных единицы искажения элементов $a_i \in A$.

Полученные результаты показывают для данного примера определенные преимущества кода Грэя по сравнению со взвешенным кодом. Однако данный вывод справедлив для принятого распределения вероятностей появления тех или иных элементов и соответственно вероятности возникновения ошибок различной кратности. Кроме того, следует обратить внимание, что сравнение проводилось по среднему значению, тогда как в определенных случаях критерием может являться не среднее значение, в ограничение в превышении некоторого максимального значения.

Обратившись к таблицам 3.2 и 3.4 несложно заметить, что при ранее оговоренных условиях при коде Грэя максимальная величина искажения может достигать 9,85 условных единиц, тогда как при взвешенном коде максимальная величина равна 8,54, что свидетельствует о преимуществе взвешенного кода по данному показателю.

Приведенные примеры показывают проблему не тривиального выбора типа биекции пространств A и B с учетом вероятностных характеристик источника сообщений, вероятностей возникновения ошибок той или иной кратности, а также принятого критерия качества.

В [26] был предложен квазиоптимальный тип биекции, который позволяет получить «хорошие» результаты. Однако доказательства его оптимальности для любых значений n не получено.

Применим этот метод для данного примера в предположении, что вероятности однократных ошибок много больше вероятностей ошибок большей кратности и, следовательно, ими можно пренебречь.

Согласно этому методу при биекции пространств наименьшие расстояния между элементами $b_i \in B$ ставятся в соответствие наименьшим расстояниям между элементами $a_i \in A$.

Учитывая, что в комбинациях $b_i \in B$ имеется C_n^1 кодовых комбинаций с кодовым расстоянием 1 от некоторой выбранной, C_n^2 кодовых комбинаций с кодовым расстоянием 2 от некоторой выбранной и т.д., а также полагая вероятность однократной ошибки много больше вероятностей ошибок большей кратности, отыщем среди $a_i \in A$ «точку сгущения», для которой отличительной чертой будет минимальное значение из $C_n^1 = C_3^1 = 3$ (трех) исходящих направлений по отношению к другим элементам. Для последующих элементов продолжим аналогичное исследование с учетом уже использованных кодовых комбинаций.

Обратимся к примеру, показанному на рисунке 3.2 и соответствующими численными оценками, представленными в 3.14.

Из точки a_0 минимальная сумма из трех направлений равна 12. Из точки a_1 – 12, из точки a_2 – 12, из точки a_3 – 12, из точки a_4 – 12,87, из точки a_5 – 12,1, из точки a_6 – 15,49, из точки a_7 – 20,87.

Полученные данные не позволяют сделать однозначный вывод при выборе «точки сгущения», т.к. имеется четыре одинаковых варианта. Однако для

конструктивного продолжения следует сделать определенный выбор. Поэтому обозначим a_0 , как «точку сгущения» и положим $a_0 \rightarrow b_0 = 000$. Далее через три самых коротких путей имеем точки, в отношении которых также установим соответствия $a_1 \rightarrow b_1 = 001$, $a_2 \rightarrow b_2 = 010$, $a_3 \rightarrow b_4 = 100$.

Из точки $a_1 \rightarrow b_1 = 001$ есть два самых коротких из имеющихся пути в точку a_2 и точку a_3 .

Из точки $a_2 \rightarrow b_2 = 010$ есть два самых коротких из имеющихся пути в точку a_1 и точку a_3 .

Из точки $a_3 \rightarrow b_4 = 100$ есть два самых коротких из имеющихся пути в точку a_2 и точку a_1 .

Однако поскольку в этих точках уже произошло установления соответствия, то никаких новых назначений в отношении этих точек не осуществляется. Т.е. $a_2 \rightarrow b_2 = 010$ и $a_3 \rightarrow b_4 = 100$.

Для следующих назначений найдем две точки, имеющие суммарно минимальное расстояние от уже назначенных точек.

От точки $a_1 \rightarrow b_1 = 001$ имеются пути в сумме 11,13.

От точки $a_2 \rightarrow b_2 = 010$ имеются пути в сумме 12,45.

От точки $a_3 \rightarrow b_4 = 100$ имеются пути в сумме 11,96.

Продолжение назначений очевидно: $a_1 \rightarrow b_1 = 001 \rightarrow a_4 \rightarrow b_3 = 011$ и $a_1 \rightarrow b_1 = 001 \rightarrow a_5 \rightarrow b_5 = 101$.

Из уже назначенных точек вычисляются расстояния до еще не назначенных точек и производится выбор на основании минимального суммарного расстояния.

В рассмотренном примере уже назначены точки $a_0 \rightarrow b_0 = 000$, $a_1 \rightarrow b_1 = 001$, $a_2 \rightarrow b_2 = 010$, $a_3 \rightarrow b_4 = 100$, $a_4 \rightarrow b_3 = 011$ и $a_5 \rightarrow b_5 = 101$. Соответственно далее $a_4 \rightarrow b_3 = 011 \rightarrow a_6 \rightarrow b_7 = 111$ и $a_4 \rightarrow b_3 = 011 \rightarrow a_7 \rightarrow b_5 = 110$.

Построим аналогично таблицам 3.1 и 3.3 результаты действия помех, приводящих к различным ошибкам.

Таблица 3.5 – Кодовые комбинации b_j после воздействия различных ошибок e

Исходные комбинации \ Ошибки	000	001	010	100	011	101	111	110
$e = 001$	001	000	011	101	010	100	110	111
$e = 010$	010	011	000	110	001	111	101	100
$e = 011$	011	010	001	111	000	110	100	101
$e = 100$	100	101	110	000	111	001	011	010
$e = 101$	101	100	111	001	110	000	010	011
$e = 110$	110	111	100	010	101	011	001	000
$e = 111$	111	110	101	011	100	010	000	001

Воспользовавшись матрице 3.14 построим таблицу искажений.

Таблица 3.6- Дополнительные искажения из-за ошибок e

Исходные комбинации \ Ошибки	000	001	010	100	011	101	111	110
$e = 001$	3,00	3,00	5,74	6,48	5,74	6,48	5,10	5,10
$e = 010$	5,00	5,74	5,00	8,54	5,74	5,00	5,00	8,54
$e = 011$	5,48	4,00	4,00	8,77	5,48	7,14	8,77	7,14
$e = 100$	4,00	5,39	6,78	4,00	5,39	5,39	5,39	6,78
$e = 101$	5,10	5,00	7,07	5,00	5,92	5,10	7,07	5,92
$e = 110$	9,85	5,83	3,00	3,00	2,00	2,00	5,89	9,85
$e = 111$	7,81	8,37	6,71	5,48	5,48	6,71	7,81	8,37

Полагая вероятность однократных ошибок много больше вероятностей ошибок большей кратности, как и ранее, подсчитаем сумму искажений.

Эта величина оказывается равной 132,32 условным единицам, что превышает аналогичные показатели для взвешенного кода и кода Грэя, что доказывает то, что оптимизация последовательными шагами не приводит к глобальной оптимизации и может рассматриваться только как один из способов поиска наилучшего решения. Еще для одного сравнения применим пошаговые решения для поиска худшего варианта.

Для этого в качестве «точки сгущения» выберем точку, обладающую наибольшим значением трех длин к остальным точкам и назначим им кодовые комбинации с кодовым расстоянием 1 относительно исходной точки.

Как следует из матрицы 3.14 относительно точки a_0 максимальная сумма трех длин равна 23,14, относительно a_1 – 19,94, относительно a_2 – 20,56, относительно a_3 – 23,79, относительно a_4 – 17,14, относительно a_5 – 20,33, относительно a_6 – 23,65, относительно a_7 – 26,76.

Исходя из полученного результат выберем точку a_7 и поставим ей в соответствие комбинацию $b_0 = 000$.

Далее установим следующие соответствия и назначения кодовых комбинаций: $a_0 \rightarrow b_1 = 001$, $a_1 \rightarrow b_2 = 010$, $a_3 \rightarrow b_4 = 100$.

Выберем из числа назначенных точек пары, обладающие наибольшим расстоянием. Для a_0 – это 13,29, для a_1 – это 11,57, для a_3 – это 13,78.

Согласно полученным значениям следующими точками соответствия будут следующие: $a_5 \rightarrow b_5 = 101$, $a_6 \rightarrow b_6 = 110$.

Продолжив далее получаем, что $a_2 \rightarrow b_5 = 110$, $a_4 \rightarrow b_7 = 111$.

В итоге имеем: $a_0 \rightarrow b_1 = 001$, $a_1 \rightarrow b_2 = 010$, $a_2 \rightarrow b_7 = 111$, $a_3 \rightarrow b_4 = 100$, $a_4 \rightarrow b_7 = 011$, $a_5 \rightarrow b_5 = 101$, $a_6 \rightarrow b_6 = 110$, $a_7 \rightarrow b_0 = 000$.

Построим таблицу переходов кодовых комбинаций при возникновении различных ошибок в элементах $b_i \in B$.

Таблица 3.7 - Кодовые комбинации b_j после воздействия различных ошибок e

Исходные комбинации \ Ошибки	001	010	111	100	011	101	110	000
$e = 001$	000	011	110	101	010	100	111	001
$e = 010$	011	000	101	110	001	111	100	010
$e = 011$	010	001	100	111	000	110	101	011
$e = 100$	101	110	011	000	111	001	010	100
$e = 101$	100	111	010	001	110	000	011	101
$e = 110$	111	110	001	010	101	011	000	110
$e = 111$	110	101	000	011	100	010	001	111

Воспользовавшись матрицей 3.14 построим таблицу искажений.

Таблица 3.8 - Дополнительные искажения из-за ошибок e

Исходные комбинации \ Ошибки	001	010	111	100	011	101	110	000
$e = 001$	9,85	5,74	7,07	6,48	5,74	6,48	7,07	9,85
$e = 010$	5,48	8,37	6,71	8,77	5,48	6,71	8,77	8,37
$e = 011$	3,00	3,00	3,00	3,00	5,92	5,00	5,00	5,92
$e = 100$	5,10	5,83	5,74	8,54	5,74	5,10	5,89	8,54
$e = 101$	4,00	4,00	4,00	4,00	5,39	7,14	5,39	7,14
$e = 110$	5,00	5,83	5,00	5,00	2,00	2,00	5,10	5,10
$e = 111$	7,81	5,39	6,78	5,48	5,48	5,39	7,81	6,78

Для сравнения рассмотренных вариантов подсчитаем суммарные искажения, возникающие при однократных ошибках. По таблице 3.8 получаем 167,36 условных единиц.

Из рассмотренных выше вариантов наименьшее значение оказалось при коде Грэя, а наибольшее при пошаговом поиске плохого варианта. Отличия составляют 39,7%, что доказывает, по крайней мере для этого примера, целесообразность поиска лучшего типа биекции множеств A и B .

В общем случае можно рекомендовать при поиске наилучшего типа биекции в целях оценки нижней и верхней границы иметь данные, полученные для последовательной оптимизации в поиске «хорошего» и «плохого» вариантов типа биекции.

3.3.2. Биекция типа A и B для регулярных структур

На практике значительную часть множеств A составляют множества, элементы которых расположены друг относительно друга согласно определенному правилу. Будем в дальнейшем называть такие множества множествами с регулярной структурой.

При аналогово-цифровом преобразовании согласно методу импульсно-кодовой модуляции после получения отсчетов их значения можно рассматривать как элементы множества A с регулярной структурой, когда все элементы $a_i \in A$ располагаются «в линейку» на определенном расстоянии друг от друга.

Положим для иллюстрации расстояние между соседними элементами равным во всех случаях единице, как было показано на рисунке 1.6. Матрица искажений (погрешностей) согласно (3.3) для данного примера имеет вид

$$Z_{i,j} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 \\ 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{pmatrix} \quad (3.17)$$

Для общего случая очевидно

$$Z_{i,j} = \begin{pmatrix} 0 & 1 & 2 & 3 & \dots & (N-2) & (N-1) \\ 1 & 0 & 1 & 2 & \dots & (N-3) & (N-2) \\ 2 & 1 & 0 & 1 & \dots & (N-4) & (N-3) \\ & & & \dots & & & \\ (N-1) & (N-2) & (N-3) & (N-4) & \dots & 1 & 0 \end{pmatrix} \quad (3.18)$$

где $N = 2^n$, а n – число двоичных символов в элементах $b_i \in B$.

Несложно заметить, что строчки матрицы (3.17) и в более общей форме (3.18) представляют собой арифметическую прогрессию, сумма N членов которой вычисляется по известной формуле:

$$S_N = \frac{2 \cdot a_1 + d \cdot (N-1)}{2} \cdot N \quad (3.19)$$

где a_1 – 1-й член арифметической прогрессии, d – шаг арифметической прогрессии, N – количество членов в арифметической прогрессии.

Для матриц (3.17) и (3.18) шаг равен 1, а первый член равен 0.

Сумма искажений в первой и последней строчках равна

$$S_0 = S_{N-1} = \frac{N \cdot (N-1)}{2} \quad (3.20)$$

Во второй и предпоследней

$$S_1 = S_{N-2} = \frac{(N-1) \cdot (N-2)}{2} + 1 \quad (3.21)$$

и т.д.

В общем виде для j -ой строчки, когда $j = \overline{0, (N-1)}$ можно записать

$$S_j = \frac{(N-j) \cdot (N-j-1) + j \cdot (j+1)}{2} \quad (3.22)$$

Общая сумма искажений определяется по формуле:

$$S_{\text{общ}} = \sum_{j=0}^{N-1} \frac{(N-j) \cdot (N-j-1) + j \cdot (j+1)}{2} \quad (3.23)$$

Отметим, что, если рассмотреть все возможные ошибки, влияющие на различные кодовые комбинации $b_i \in B$, и, переводящие их в другие кодовые

комбинации этого множества, то будут представлены все возможные искажения матрицы 3.18. Поэтому еще раз следует отметить, что, если все виды ошибок равновероятны, то тип биекции A и B не влияет на результат искажений возникающих в элементах $a_i \in A$ из-за ошибок в $b_i \in B$. Однако на практике этого не происходит и ошибки различной кратности имеют существенно разные вероятности, что предопределяет отличия в возникающих искажения и обосновывает необходимость поиска оптимальных или по крайней мере «хороших» типов биекции.

Для оценки этого следует провести расчеты для наиболее известных методов двоичного кодирования, а именно: взвешенного кода, кода Грэя, а также для «квазиоптимального» и «квази не оптимального» кодов, как это было показано выше путем соответствующего выбора «точки сгущения».

Для взвешенного кода матрица искажений представлена в (3.17). Далее на основании таблицы 3.1 построим таблицу искажений для взвешенного кода.

Таблица 3.9 – Дополнительные искажения из-за ошибок e

Исходные комбинации \ Ошибки	000	001	010	011	100	101	110	111
$e = 001$	1	1	1	1	1	1	1	1
$e = 010$	2	2	2	2	2	2	2	2
$e = 011$	3	1	1	3	3	1	1	3
$e = 100$	4	4	4	4	4	4	4	4
$e = 101$	5	3	5	3	3	5	3	5
$e = 110$	6	6	2	2	2	2	6	6
$e = 111$	7	5	3	1	1	3	5	7

Полагая, как и ранее, вероятность однократных ошибок много больше вероятностей ошибок большей кратности получаем суммарную величину

искажений при условии равновероятной передачи элементов $b_i \in B$, равной 56. При этом максимальной искажение происходит при $e=100$ оно равно 4, что вполне объяснимо структурой формирования взвешенного кода.

Для кода Грея воспользуемся таблицей 3.3 и матрицей искажений (3.17), измененной согласно коду Грея. В результате получаем таблицу искажений при всех возможных ошибках.

Как следует из таблицы 3.10 суммарная величина искажений при однократных ошибках составляет 56 условных единиц, т.е. столько же, как и при взвешенном кодировании. Однако при однократной ошибке вида $e = 100$ возможны максимальные искажения равные 7, что говорит в пользу взвешенного кода по сравнению с кодом Грея, поскольку при одинаковой в среднем суммарной величине искажений из-за однократных ошибок, при коде Грея возможны большие искажения равные 7 и 5, тогда как при взвешенном коде искажения больше 4 не возможны. Однако код Грея превосходит взвешенный код по числу минимальных искажений в 1 –ну единицу. Их число равно 12 - ти, тогда как во взвешенном коде это число равно 8.

Таблица 3.10 - Дополнительные искажения из-за ошибок e

Исходные комбинации \ Ошибки	000	001	011	010	110	111	101	100
$e = 001$	1	1	1	1	1	1	1	1
$e = 010$	3	1	1	3	3	1	1	3
$e = 011$	2	2	2	2	2	2	2	2
$e = 100$	7	5	3	1	1	3	5	7
$e = 101$	6	6	2	2	2	2	6	6
$e = 110$	4	4	4	4	4	4	4	4
$e = 111$	5	3	5	3	3	5	3	5

Исследуем вариант «хорошего» решения, определив «точкой сгущения» среднее значение в линейной конструкции, поскольку в рамках исследуемой задачи положено, что расстояния между соседними элементами $a_i \in A$, одинаково и равно некоторой условной единице.

Для определенности выберем a_3 «точкой сгущения» и назначим этой точке комбинацию $b_0 = 000$. Соответственно точке a_2 назначим $b_1 = 001$, точке a_1 - $b_2 = 010$ точке, a_4 - $b_4 = 100$. Далее очевидно, что $a_0 = b_3 = 011$, $a_5 = b_5 = 101$, $a_6 = b_7 = 111$, $a_7 = b_6 = 110$.

Для полученного варианта построим таблицу переходов кодовых комбинаций при возникновении различных ошибок в элементах $b_i \in B$.

Видоизменив соответствующим образом матрицу искажений (3.17), построим таблицу искажений для $a_i \in A$.

Таблица 3.11 - Кодовые комбинации b_j после воздействия различных ошибок e

Исходные комбинации Ошибки	011	010	001	000	100	101	111	110
$e = 001$	010	011	000	001	101	100	110	111
$e = 010$	001	000	011	010	110	111	101	100
$e = 011$	000	001	010	011	111	110	100	101
$e = 100$	111	110	101	100	000	001	011	010
$e = 101$	110	111	100	101	001	000	010	011
$e = 110$	101	100	111	110	010	011	001	000
$e = 111$	100	101	110	111	011	010	000	001

Таблица 3.12- Дополнительные искажения из-за ошибок e

Исходные комбинации \ Ошибки	011	010	001	000	100	101	111	110
$e = 001$	1	1	1	1	1	1	1	1
$e = 010$	2	2	2	2	3	1	1	3
$e = 011$	3	1	1	3	2	2	2	2
$e = 100$	6	6	3	1	1	3	6	6
$e = 101$	7	5	2	2	2	2	5	7
$e = 110$	5	3	4	4	3	5	4	4
$e = 111$	4	4	5	3	4	4	3	5

Для однократных ошибок, полагая их вероятность много больше вероятности ошибок большей кратности, и, полагая равновероятным передачу кодовых комбинаций $b_i \in B$, получаем из таблицы 3.12 суммарную величину искажений равную 56 условным единицам.

При этом максимальная величина искажений может составлять 6 условных единиц, что больше, чем при взвешенном коде, но меньше, чем при коде Грэя.

С целью определения границ оптимизации исследуем вариант «плохого» кодирования при выборе той же «точки сгущения», но назначив при начальной точке $a_3 = b_0 = 000$ остальные точки с максимальным удалением по Хэммингу.

В результате получаем: $a_0 = b_5 = 110$, $a_1 = b_1 = 001$, $a_2 = b_7 = 111$, $a_3 = b_0 = 000$, $a_4 = b_3 = 011$, $a_5 = b_4 = 101$, $a_6 = b_3 = 010$, $a_7 = b_4 = 100$.

Построим таблицу переходов кодовых комбинаций при возникновении различных ошибок в элементах $b_i \in B$ для данного «плохого» варианта.

Таблица 3.13 - Переходы кодовых комбинаций при возникновении различных ошибок в элементах $b_i \in B$

Исходные комбинации \ Ошибки	110	001	111	000	011	101	010	100
$e = 001$	111	000	110	001	010	100	011	101
$e = 010$	100	011	101	010	001	111	000	110
$e = 011$	101	010	100	011	000	110	001	100
$e = 100$	010	101	011	100	111	001	110	000
$e = 101$	011	100	010	101	110	000	111	001
$e = 110$	000	111	001	110	101	011	100	010
$e = 111$	001	110	000	111	100	010	101	011

Для этой таблицы, воспользовавшись измененной матрицей искажений (3.17) построим таблицу искажений элементов $a_i \in A$.

Таблица 3.14 - Искажения элементов $a_i \in A$.

Исходные комбинации \ Ошибки	110	001	111	000	011	101	010	100
$e = 001$	2	2	2	2	2	2	2	2
$e = 010$	7	3	3	3	3	3	3	7
$e = 011$	5	5	5	1	1	5	5	5
$e = 100$	6	4	2	4	2	4	6	4
$e = 101$	4	6	4	2	4	2	4	6
$e = 110$	3	1	1	3	1	1	1	1
$e = 111$	1	1	1	1	3	1	1	3

При оговоренных выше условиях действия однократной ошибки суммарная величина искажений для данного типа биекции составит 80 условных единиц, что на 42,9% больше, чем в рассмотренных ранее методах. Недостатком также является то, что здесь, как и в коде Грэя, возможны максимальные искажения равные в данном примере 7 условным единицам.

Регулярность структуры пространства A при биекции с пространством B позволяет для рассмотренных вариантов ошибок сделать обобщения для любых длин кодовых комбинаций $b_i \in B$. Особенно явственно это проявляется для взвешенного кода, когда каждый i -ый символ в кодовой комбинации имеет определенный вес, а именно 2^i .

Именно поэтому ошибка вида $e = 001$, влияющая во всех комбинациях $b_i \in B$, вносит искажения равные 1-ой условной единице для всех $a_i \in A$. Ошибка вида $e = 010$ искажает $a_i \in A$ ровно на 2-е условные единицы, ошибка вида $e = 100$ на 4-е условные единицы и соответственно ошибка состоящая из n двоичных символов, где первый единица, а остальные нули, т.е. $e = 100\dots 0$ вносит искажения во всех $a_i \in A$ на 2^{n-1} условных единиц.

Таким образом, суммарные искажения из-за различных однократных ошибок для взвешенного кода равны

$$Z_{\text{сум-1-взвеш}} = 2^n \cdot \sum_{i=0}^{n-1} 2^i = 2^n \cdot (2^n - 1) \quad (3.23)$$

Рассмотрим, как формируются искажения для кода Грэя с ростом длины кодовой комбинации n .

Пронумеруем кодовую комбинацию справа-налево.

Кодовая комбинация при коде Грэя с увеличением длины кодовой комбинации на единицу образуется из предыдущих кодовых комбинаций с приписанным слева нулем, и далее из предыдущих кодовых комбинаций, взятых в обратном порядке с приписыванием слева единицы.

Такой формат, такая инверсная симметрия позволяет оценить искажения, возникающие при ошибках различной кратности и, в частности, при разных однократных ошибках.

Полагая, что вероятность однократных ошибок много больше вероятности ошибок большей кратности ограничимся исследованием возникающих искажений в $a_i \in A$ именно по этой причине для произвольных значений длины кодовой комбинации n , хотя с ее ростом это допущение становится все более слабым и не точным.

Если возникает ошибка в первом справа разряде кодовой комбинации элементов $b_i \in B$, то это приводит к искажению элемента $a_i \in A$ ровно на 1.

Далее при ошибке во втором разряде, в силу указанных выше правил построения кода Грэя, возникнут искажения вида: 3, 1 и 1, 3.

Ошибка в третьем разряде, опять же в силу инверсного принципа формирования кодовых комбинаций при коде Грэя, приведет к искажениям вида: 7, 5, 3, 1 и 1, 3, 5, 7.

Ошибка в четвертом разряде приводит к искажениям вида: 15, 13, 11, 9, 7, 5, 3, 1 и 1, 3, 5, 7, 9, 11, 13, 15.

Ошибка в i -ом разряде приводит к искажениям, которые можно записать в общем виде как: $(2^i - 1), (2^i - 3), (2^i - 5), \dots, 3, 1$ и $1, 3, \dots, (2^i - 5), (2^i - 3), (2^i - 1)$.

Ошибка в первом разряде, приводящая к искажениям в 1, может произойти в 2^n случаях, рассматривая все возможные кодовые комбинации $b_i \in B$, состоящие из n двоичных символов.

Ошибка во втором разряде приводит к суммарным искажениям для пары кодовых комбинаций равным 4. Если кодовая комбинация состоит из n символов, то таких пар будет 2^{n-1} .

Ошибка в третьем разряде дает «четверку» искажений суммой равной 16. И таких четверок будет 2^{n-2} .

Ошибка в четвертом разряде дает «восьмерку» искажений суммой равной 64. И таких восьмерок будет соответственно 2^{n-3} .

Для ошибки в i - ом разряде справедливо указать на возникновение соответствующих искажений суммой в 4^i в 2^{n-i+1} случаях.

Исходя из изложенного, суммарные искажения в $a_i \in A$ из-за различных однократных ошибок в $b_i \in B$ при одинаковой вероятности передачи этих кодовых комбинаций равны:

$$Z_{\text{сум-1-Грэй}} = 2^n \cdot \sum_{i=1}^n 4^i \cdot 2^{n-i+1} = \sum_{i=1}^n 2^{n+i-1} \quad (3.24)$$

Выражение (3.24) можно преобразовать к следующему виду:

$$Z_{\text{сум-1-Грэй}} = \sum_{i=1}^n 2^{n+i-1} = 2^n \cdot \sum_{i=0}^{n-1} 2^i = 2^n \cdot (2^n - 1) \quad (3.25)$$

Сравнение (3.25) и (3.23) показывает, что код Грэя и взвешенный код эквивалентны по показателю суммарных искажений $a_i \in A$, возникающих из-за одиночных ошибок в кодовых комбинациях $b_i \in B$.

При этом максимальные искажения при коде Грэя могут достигать величины $(2^n - 1)$, тогда как при взвешенном коде максимальная величина искажений не превосходит 2^{n-1} , т.е. почти в два раза меньше. В тоже время количество минимальных искажений, величиной 1, при коде Грэя равно $2 \cdot (2^n - 1)$, тогда как при взвешенном коде это число равно 2^n , т.е. в почти два раза меньше.

Для большей наглядности представим на рисунке 3.4 а,б,в,г,д,е спектры искажений $a_i \in A$ при соответственно $n = 3, 4, 5, 6, 7, 8$ для взвешенного кода и кода Грэя.

Для взвешенного кода построения ведутся по формулам, когда искажения равные 1 встречаются 2^n раз, искажения равные 2 - 2^{n-1} раз, искажения равные 4 - 2^{n-2} раз, и т.д. искажения 2^{n-1} - также 2 раз. Т.е. распределение носит равномерный характер.

Для кода Грэя искажения величиной 1 появляются в $2 \cdot (2^n - 1)$ случаях, искажение величиной 3 в $2 \cdot (2^{n-1} - 1)$ случаях, искажения величиной 5 в $2 \cdot (2^{n-2} - 1)$ случаях, искажения 7 также в $2 \cdot (2^{n-2} - 1)$ случаях.

Далее искажения 9, 11, 13, 15 в $2 \cdot (2^{n-3} - 1)$ случаях каждое, потом следующие 8 нечетных искажений 17, 19, 21, 23, 25, 27, 29, 31 в $2 \cdot (2^{n-4} - 1)$ случаях каждое, потом далее 16 нечетных искажений, 32 и т.д. в соответствующем числе случаев, которое определяется аналогично.

На рисунках ниже наглядно показана зависимость количества встречаемых искажений k от величины искажения v для Натурального кода и кода Грея (синим и оранжевым цветом соответственно).

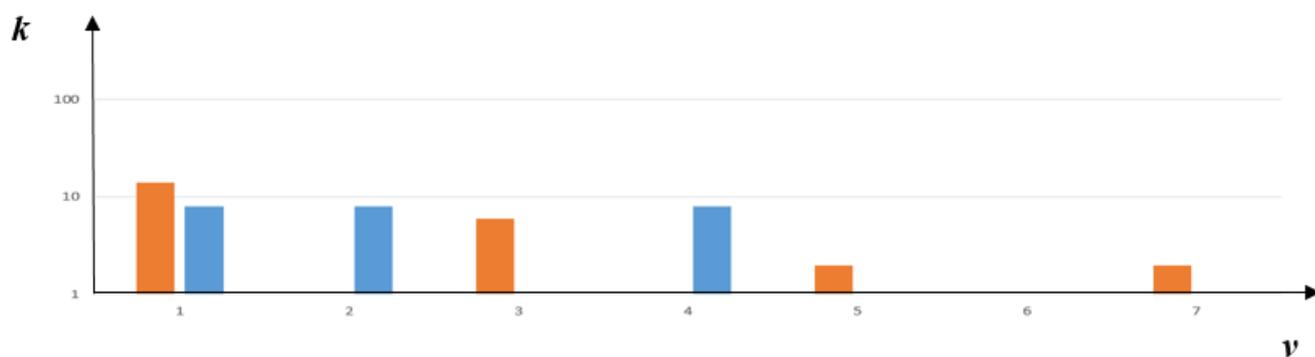


Рисунок 3.4а – при $n=3$

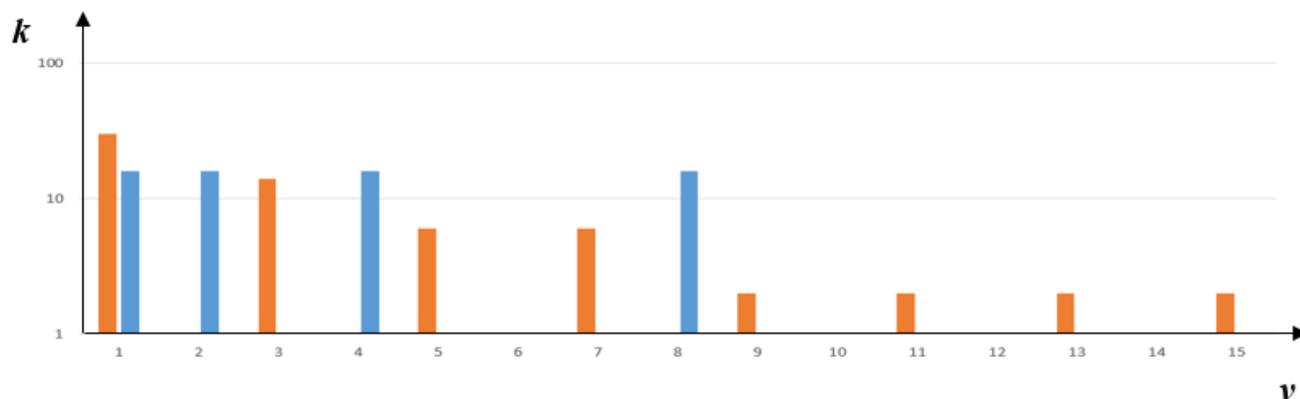
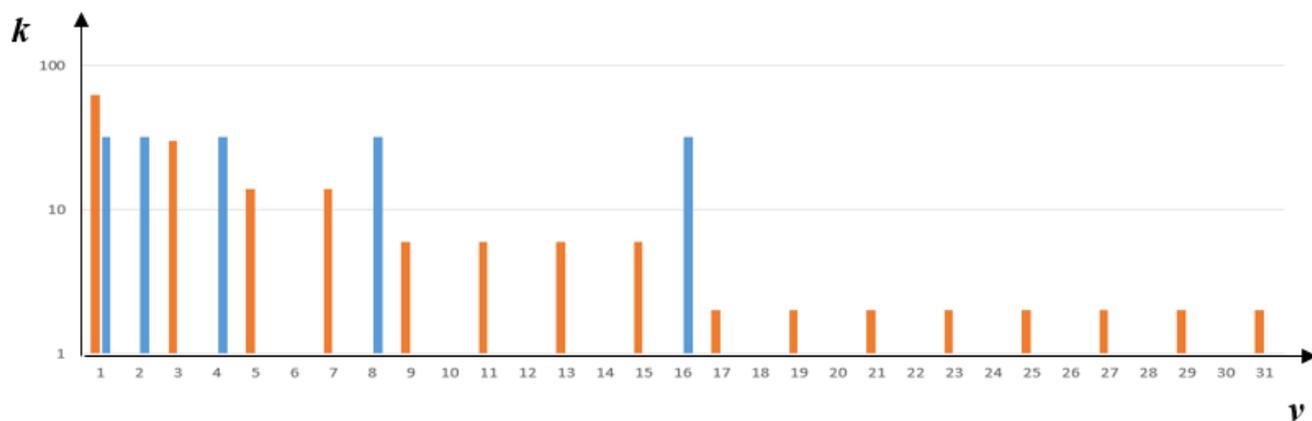
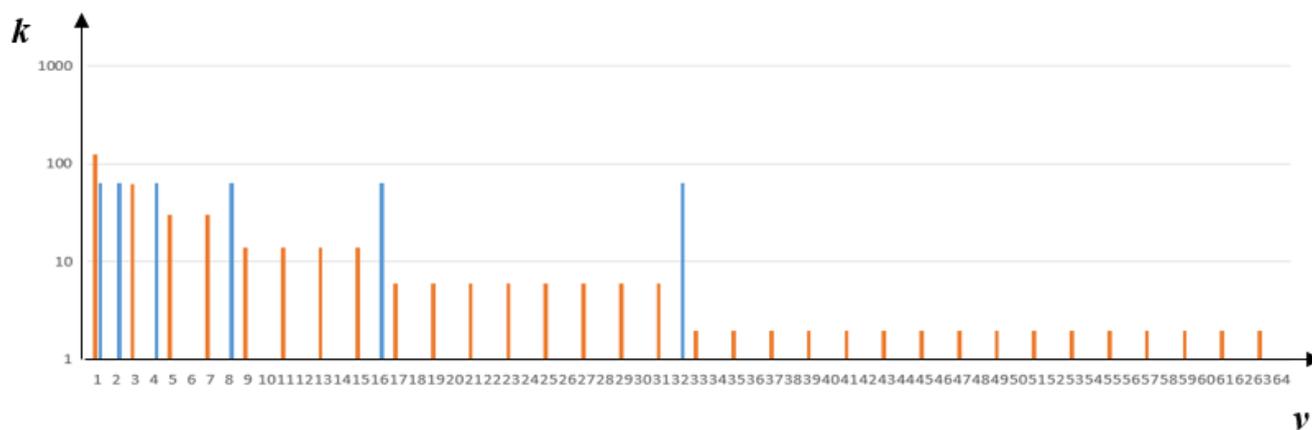
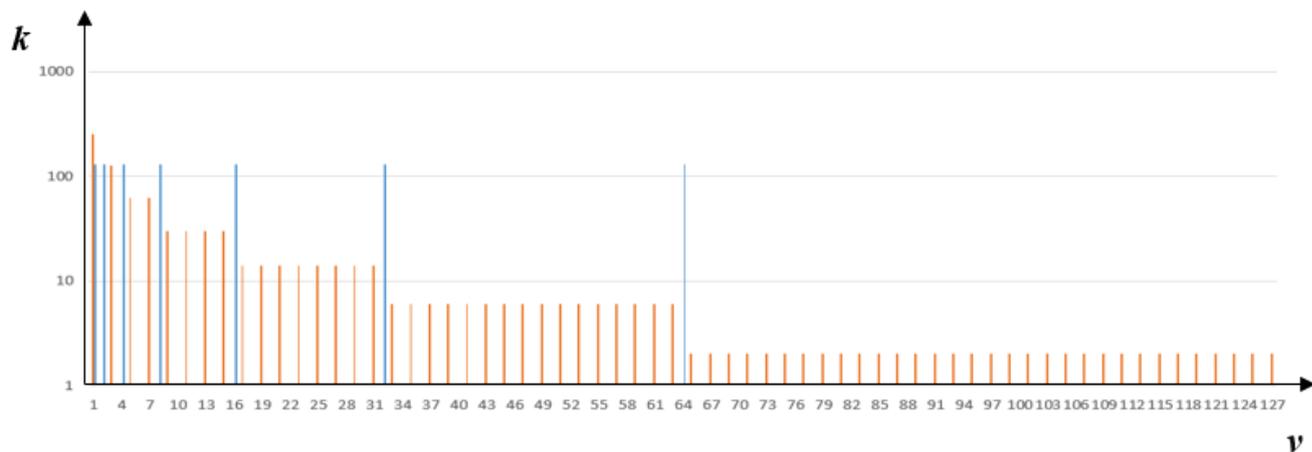


Рисунок 3.4б – при $n=4$

Рисунок 3.4в – при $n=5$ Рисунок 3.4г – при $n=6$ Рисунок 3.4д – при $n=7$

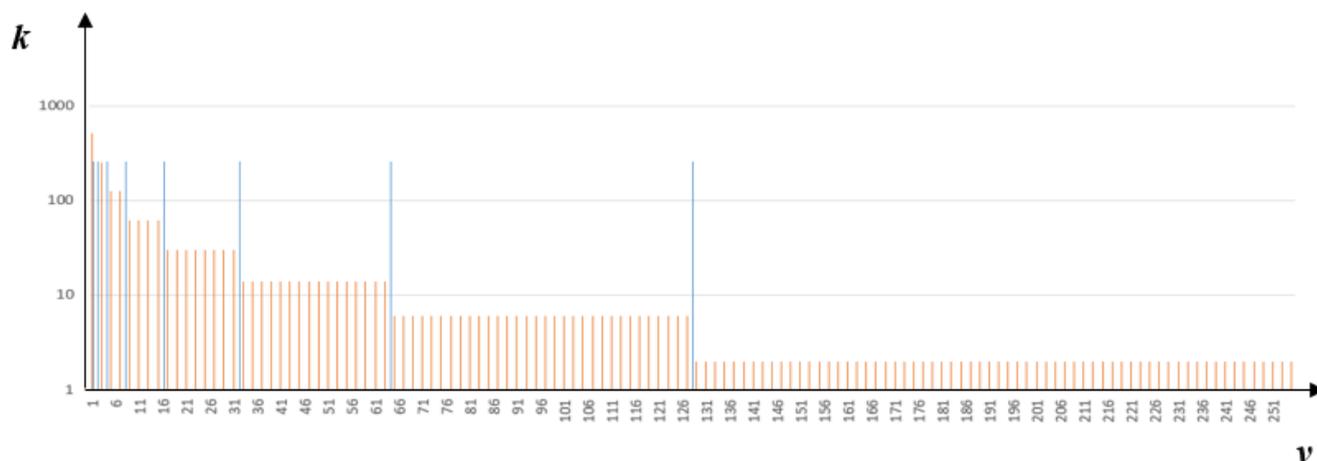


Рисунок 3.4е – при $n=8$

Исходя из изложенного, можно осуществлять соответствующий выбор типа биекции пространств A и B , с учетом требований по показателям качества трансформации элементов $a_i \in A$ в элементы $b_i \in B$.

3.4 Выводы по разделу

Разработанный метод анализа биекции множества A и множества B , соответствующий преобразованию элементов исходного сигнала $a_i \in A$ в цифровые элементы $b_i \in B$ пространства Хэмминга, позволяет провести количественные расчеты с целью выбора наилучшего метода биекции, обеспечивающего минимальную величину дополнительных искажений, возникающих в элементах $a_i \in A$ из-за ошибок в элементах $b_i \in B$. Предложенные «хорошие» и «плохие» методы биекции дают возможность получить оценки диапазона выигрыша по минимизации дополнительных искажений, возникающих в элементах $a_i \in A$ из-за ошибок в элементах $b_i \in B$, за счет «хорошего» выбора с учетом возможностей взвешенного кодирования и кода Грэя. Показано, что при однократных ошибках в элементах $b_i \in B$ наибольшее количество минимальных искажений в элементах $a_i \in A$ достигается при коде Грэя. Однако при этом имеются и максимально возможные искажения величиной $(2^n - 1)$, тогда как для взвешенного кода максимальная

величина искажений не превосходит 2^{n-1} . Важно также отметить, что среднее значение искажений для регулярной структуры элементов $a_i \in A$ совпадает.

Разработанный метод сравнения типов биекции множества A и множества B в условиях возникновения ошибок в элементах $b_i \in B$ позволяет получить спектр возникающих искажений, что дает возможность осуществить выбор типа биекции, исходя из различных условий выбора. Например, по принципу минимизации среднего значения возникающих дополнительных искажений, по принципу минимизации максимальной величины возникающих искажений и т.д.

4. РАЗРАБОТКА МЕТОДА ИССЛЕДОВАНИЯ ВОЗНИКНОВЕНИЯ ОШИБОК В ЦИФРОВОМ КАНАЛЕ ПРИ НАЛИЧИИ ИСКАЖЕНИЙ И ПРЕОБРАЗОВАНИИ РАЗЛИЧНЫХ МЕТРИЧЕСКИХ ПРОСТРАНСТВ

4.1. Постановка задачи и определения

В Разделе 3 исследовались вопросы, связанные с преобразование элементов $a_i \in A$ в элементы $b_i \in B$.

Далее сформированный цифровой сигнал подвергается помехоустойчивому кодированию, после чего согласно рисунку 3.1, формируется новый цифровой сигнал $c_i \in C$, отображающие элементы $b_i \in B$.

Преобразование пространства B в пространство C можно назвать биекцией, поскольку оно отвечает условиям инъективности и сюръективности. При этом подчеркнем, что под пространством C понимается пространство разрешенных кодовых комбинаций помехоустойчивого кода. Отметим также, что биекция пространств отвечает и условию изометричности, поскольку биекция между этими пространствами сохраняет расстояние, определяемой метрикой пространства Хэмминга, к которому относится как пространство B , так и пространство C .

Помехоустойчивому кодированию посвящено огромное количество работ [48,49,50,51], в которых подробно исследуются различные стороны этого важного преобразования в общей телекоммуникационной системе. Поэтому в данной работе ограничимся констатацией факта соответствующей биекции пространств B и C , а также ссылками на известные работы в этой области.

Следуя рисунку 3.2, после помехоустойчивого кодирования в современных системах связи часто используют операцию скремблирования, что можно рассматривать также как биекцию пространств, в данном случае пространства C и пространства G , в результате чего элементы $c_i \in C$ преобразуются в элементы $g_i \in G$.

Отметим, что данной проблеме также посвящено много работ, особенно если учесть, что при этой операции осуществляются еще и функции шифрования [42,43]. Поэтому и здесь ограничимся лишь указанием на биекцию пространств C и G .

Завершающее преобразование пространств, согласно рисунку 3.2 на передающей стороне происходит, когда в модуляторе элементы $g_i \in G$ трансформируются в элементы $q_i \in Q$. С учетом соответствующих обратных преобразований на приемной стороне это преобразование множеств G и Q отвечает определению биекции, но не является изометричным.

Исследуем это преобразование более подробно с учетом того, что в отличие от проблем, которые были рассмотрены в Разделе 3, здесь происходят противоположные преобразования, когда на передающей стороне пространство Хэмминга преобразуется в пространство Евклида.

Помехи, действующие в канале связи, вносят искажения в элементы $q_i \in Q$, в результате которых на приемной стороне элементы $g_i \in G$ и далее в силу биекции элементы $c_i \in C$ и $b_i \in B$ могут содержать ошибки. При этом, как будет показано ниже, интенсивность этого процесса зависит от типа биекции пространств G и Q .

4.2. Биекция пространств G и Q

Пусть сообщениям от некоторого источника соответствуют элементы A, B, C, D , расположенные геометрически в углах квадрата и пусть расстояние между этими точками определяется геометрическими размерами квадрата со стороной равной l .

Очевидно, что матрица, отражающая попарные расстояния между точками A, B, C, D , заданных в пространстве Евклида, будет иметь вид матрицы L_4 :

$$L_4 = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{pmatrix} 0 & l & l\sqrt{2} & l \\ l & 0 & l & l\sqrt{2} \\ l\sqrt{2} & l & 0 & l \\ l & l\sqrt{2} & l & 0 \end{pmatrix} & \begin{matrix} A \\ B \\ C \\ D \end{matrix} \end{matrix} \quad (4.1)$$

Ее элементы для данной модели определяются элементами первой строки или первого столбца, что эквивалентно. В общем случае эти условия могут не выполняться, но для данной модели - это справедливо.

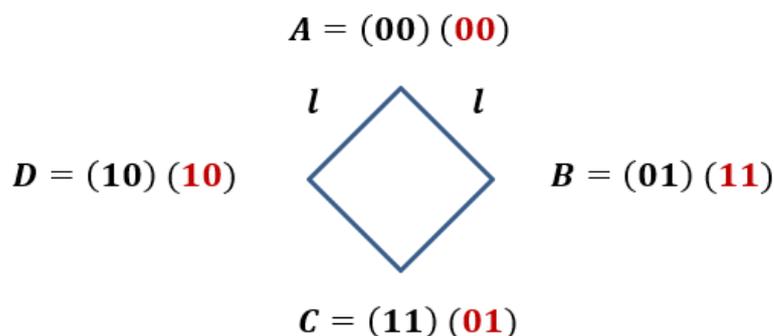


Рисунок 4.1 - Модель расположения кодовых комбинаций

Мы имеем матрицу L_4 , определяющую расстояния между сообщениями согласно введенной метрике. Положим, что эти расстояния будут эквивалентны потерям. Тогда, сопоставляя эту матрицу с матрицей кодовых расстояний $D_2(d)$, определенную в Разделе 3, можно увидеть отличия между вариантами сопоставления.

В первом случае матрице L_4 ставится в соответствие матрица $D_2(d) = D_2^1(d)$

$$D_2^1(d) = \begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix} \quad (4.2)$$

а во втором случае матрице L_4 будет поставлена в соответствие иная матрица $D_2(d) = D_2^2(d)$, а именно:

$$D_2^2(d) = \begin{pmatrix} 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix} \quad (4.3)$$

Сопоставляя элементы матрицы L_4 и $D_2^1(d)$, а также L_4 и $D_2^2(d)$, видим, что в первом случае наибольшее расстояние в матрице L_4 , а именно, $l\sqrt{2}$ совпадает с наибольшим расстоянием в матрице $D_2^2(d)$, а именно, 2, а наименьшее расстояние, а именно, l совпадает с наименьшим расстоянием равным 1. Во втором случае это не так. Если предположить, что в результате помех в канале связи, приводящих к искажениям в $s(t)$ и затем к ошибкам в цифровом сигнале $s_i(t)$, произойдет трансформация одной кодовой комбинации в другую, то вполне резонно полагать, что такое переход будет более вероятен, когда расстояние между соответствующими элементами матриц $D_2^i(d)$ наименьшее. В этом случае, очевидно, что первый вариант будет иметь определенные преимущества.

В формуле (1.4) указано число различных вариантов кодирования, которое соответствует числу типов биекции пространств G и Q .

Вводя переобозначение, запишем эту формулу для определения числа биекций, если длина кодовой комбинации элементов $g_i \in G$ равна n .

$$N_{Bi} = (2^n)! \quad (4.4)$$

Не сложно заметить, что с ростом длины кодовой комбинации n число типов биекций очень быстро нарастает.

При $n = 2$ имеем, согласно (4.4), число биекций равно $N_{Bi} = 24$, а при $n = 3$ уже $N_{Bi} = 40320$.

Конечно, обычный перебор «вручную» различных вариантов биекции невозможен. Однако можно показать эквивалентность определенного числа биекций, что позволит заметно сократить их разнообразие.

Рассмотрим это на примере, когда $n = 3$.

Положим для определенности, что элементы множества $g_i \in G$ соответствуют двоичным кодовым комбинациям взвешенного кода, т.е. $g_0 = 000, g_1 = 001, g_2 = 010, g_3 = 011, g_4 = 100, g_5 = 101, g_6 = 110, g_7 = 111$.

Спектр кодовых расстояний указанных комбинаций согласно (3.10) соответствует матрице кодовых расстояний

$$D_3^{B_{3e}}(d) = \begin{pmatrix} 0 & 1 & 1 & 2 & 1 & 2 & 2 & 3 \\ 1 & 0 & 2 & 1 & 2 & 1 & 3 & 2 \\ 1 & 2 & 0 & 1 & 2 & 3 & 1 & 2 \\ 2 & 1 & 1 & 0 & 3 & 2 & 2 & 1 \\ 1 & 2 & 2 & 3 & 0 & 1 & 1 & 2 \\ 2 & 1 & 3 & 2 & 1 & 0 & 2 & 1 \\ 2 & 3 & 1 & 2 & 1 & 2 & 0 & 1 \\ 3 & 2 & 2 & 1 & 2 & 1 & 1 & 0 \end{pmatrix} \quad (4.5)$$

а спектр весов имеет вид первой строки данной матрицы, а именно: $w = 0; 1; 1; 2; 1; 2; 2; 3$.

При биекции множеств G и Q и при использовании взвешенного кода устанавливается следующее соответствие:

$$\begin{aligned} q_0 \rightarrow g_0 = 000, q_1 \rightarrow g_1 = 001, q_2 \rightarrow g_2 = 010, q_3 \rightarrow g_3 = 011, q_4 \rightarrow g_4 = 100, q_5 \rightarrow g_5 = 101, \\ q_6 \rightarrow g_6 = 110, q_7 \rightarrow g_7 = 111 \end{aligned} \quad (4.6)$$

Эквивалентный тип биекции означал бы, что новый тип биекции обладал бы тем же спектром кодовых расстояний, как и первоначальный, т.е. для нашего примера, как биекция при взвешенном коде.

Обратимся к определению кодового расстояния в пространстве Хэмминга. Это расстояние равно числу разрядов, в которых одна комбинация отличается от другой. Следовательно, эта величина не зависит от места расположения данного отличия в кодовых комбинациях, состоящих из n двоичных символов.

Из этого следует, что если во всех комбинациях внутри них осуществить одинаковый обмен двоичными символами, например, второй символ на третий и т.д., то от этого спектр кодовых расстояний не измениться.

При длине кодовой комбинации в n двоичных символов число эквивалентных биекций будет равно числу перестановок внутри кодовой комбинации, т.е.

$$P_{Bi} = (n)! \quad (4.7)$$

Для нашего примера при $n = 3$ число эквивалентных биекций равно $P_{Bi} = 6$.

Укажем их для определенности в таблице 4.1

Таблица 4.1- Эквивалентные кодовые комбинации

	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7
Взвешенный код (3,2,1)	000	001	010	011	100	101	110	111
Эквивалент 1 (3,1,2)	000	010	001	011	100	110	101	111
Эквивалент 2 (2,3,1)	000	001	100	101	010	011	110	111
Эквивалент 3 (2,1,3)	000	010	100	110	001	011	101	111
Эквивалент 4 (1,2,3)	000	100	010	110	001	101	011	111
Эквивалент 5 (1,3,2)	000	100	001	101	010	110	011	111

Несложный анализ эквивалентных вариантов показывает, что для каждого из них спектр кодовых расстояний оказывается таким же, как и для взвешенного кода.

Помимо рассмотренных случаев эквивалентных биекций существует еще множество матриц с таким же распределением кодовых расстояний.

Покажем это.

Действительно. Поскольку множество G отвечает условию существования метрического пространства, то согласно принятой аксиоматике, если ко всем кодовым комбинациям прибавить одно и то же число, то расстояние между ними не изменится. А значит, последовательно прибавляя к кодовым комбинациям, показанным, например, в таблице 4.1 все возможные кодовые комбинации можно получить все множество матриц с одинаковым распределением кодовых расстояний.

Поскольку общее число различных биекций определяется данным кодом, то можно получить таблицы кодовых расстояний с таким же распределением весов, но в иной последовательности кодовых расстояний.

Приведем пример.

Прибавим ко всем кодовым комбинациям из таблицы 4.1 комбинацию (011).

Тогда таблица 4.1 примет вид:

Таблица 4.2- Эквивалентные кодовые комбинации

	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7
Взвешенный код (3,2,1)	011	010	001	000	111	110	101	100
Эквивалент 1 (3,1,2)	011	001	010	000	111	101	110	100
Эквивалент 2 (2,3,1)	011	010	111	110	001	000	101	100
Эквивалент 3 (2,1,3)	011	001	111	101	010	000	110	100
Эквивалент 4 (1,2,3)	011	111	001	101	010	110	000	100
Эквивалент 5 (1,3,2)	011	111	010	110	001	101	000	100

Сравнение кодовых комбинаций из таблицы 4.1 и таблицы 4.2 показывает идентичность спектров кодовых расстояний и даже сохранения последовательности. Однако спектр весов, сохраняя те же численные значения, меняет их последовательность, а именно, теперь для таблицы 4.2 спектр весов с сохранением последовательности чисел имеет вид: $w = 2; 1; 1; 0; 3; 2; 2; 1$.

Данное обстоятельство может оказаться важным при биекции множеств G и Q , поскольку в отличие от (4.6), будет получена следующая последовательность соответствия элементов $g_i \in G$ и $q_i \in Q$:

$$\begin{aligned} q_0 \rightarrow g_3 = 011, q_1 \rightarrow g_2 = 001, q_2 \rightarrow g_2 = 001, q_3 \rightarrow g_0 = 000, q_4 \rightarrow g_7 = 111, q_5 \rightarrow g_6 = 110, \\ q_6 \rightarrow g_5 = 101, q_7 \rightarrow g_4 = 100 \end{aligned} \quad (4.8)$$

При определенных условиях отличия, отраженные в (4.6) и (4.8) могут не иметь значения, если искажения элементов $q_i \in Q$ одинаковы. Но если это допущение не соответствует действительности, то биекции, соответствующие (4.6) и (4.8) нельзя признать эквивалентными. Поэтому случаи, примером которых является (4.8), будем называть квазиэквивалентными биекциями.

Суммируя вышеизложенное, отмечаем, что число всех вариантов биекций может быть сокращено в $\Pi_{Bi} = (n)!$ раз, как указано в (4.7), а также дополнительно за счет квазиэквивалентности еще в 2^n раз.

В конечном итоге это означает, что общее число вариантов различных биекций можно рассчитать по формуле:

$$N_{1-Bi} = \frac{(2^n)!}{(2^n) \cdot n!} \quad (4.9)$$

Для рассматриваемого примера, когда $n = 3$ число вариантов, рассчитанное по формуле (4.4), равнялось 40 320. С учетом формулы (4.9) получаем 840, что заметно меньше, а это существенно.

Для $n = 2$ соответственно имеем $N_{Bi} = 24$ и $N_{1-Bi} = 3$.

Из известных, чаще всего применяемых кодов, – это взвешенный код и код Грэя. Возвращаясь к примеру, когда $n = 2$, можно отметить, что фактически существуют только три группы квазиэквивалентных кодов, названные взвешенный код, код Грэя и некоторый третий тип кода.

Исследуем это более подробно и построим таблицы этих кодов.

Для взвешенного кода имеем:

Таблица 4.3- Эквивалентные кодовые комбинации

	q_0	q_1	q_2	q_3
Взвешенный код (2,1)	00	01	10	11
Эквивалентный код (1,2)	00	10	01	11
Квазиэквивалентный код 1 (2,1)	01	00	11	10
Квазиэквивалентный код 2 (1,2)	10	00	11	01
Квазиэквивалентный код 3 (2,1)	10	11	00	01
Квазиэквивалентный код 4 (1,2)	01	11	00	10
Квазиэквивалентный код 5 (2,1)	11	10	01	00
Квазиэквивалентный код 6 (1,2)	11	01	10	00

У всех этих кодов эквивалентная матрица кодовых расстояний, в чем несложно убедиться и одинаковая последовательность кодовых расстояний,

характеризующих отличия в кодовых комбинациях, а именно (0, 1, 1, 2) (см. первую строчку матрицы).

$$D_2^{B3e} = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} \quad (4.10)$$

Для кода Грэя получаем следующую таблицу эквивалентных и квазиэквивалентных биекций:

Таблица 4.4- Эквивалентные кодовые комбинации

	q_0	q_1	q_2	q_3
Код Грэя (2,1)	00	01	11	10
Эквивалентный код (1,2)	00	10	11	01
Квазиэквивалентный код 1 (2,1)	01	00	10	11
Квазиэквивалентный код 2 (1,2)	10	00	01	11
Квазиэквивалентный код 3 (2,1)	10	11	01	00
Квазиэквивалентный код 4 (1,2)	01	11	10	00
Квазиэквивалентный код 5 (2,1)	11	10	00	01
Квазиэквивалентный код 6 (1,2)	11	01	00	10

В этом случае получается иная последовательность кодовых отличий (0, 1, 2, 1) (см. первую строчку матрицы), что несложно заметить, сравнив матрицу (4.10) и приведенную ниже (4.11)

$$D_2^{Грей} = \begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix} \quad (4.11)$$

Оставшийся третий тип, представлен в таблице 4.5.

$$D_2^{Грей} = \begin{pmatrix} 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix} \quad (4.12)$$

Таблица 4.5 - Эквивалентные кодовые комбинации

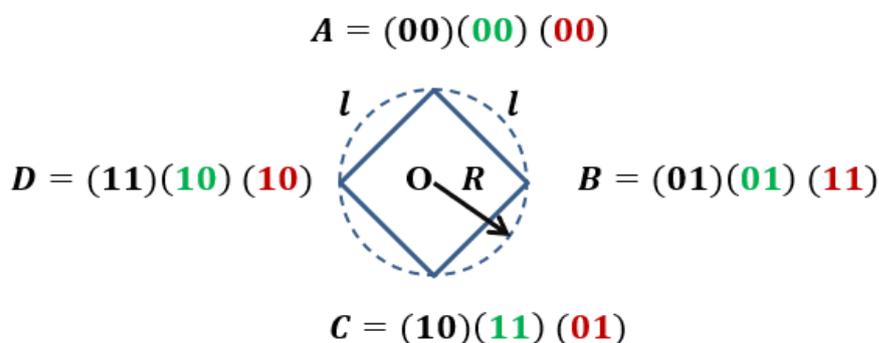
	q_0	q_1	q_2	q_3
Третий тип кодирования (2,1)	00	11	01	10
Эквивалентный код (1,2)	00	11	10	01
Квазиэквивалентный код 1 (2,1)	01	10	00	11
Квазиэквивалентный код 2 (1,2)	01	01	00	11
Квазиэквивалентный код 3 (2,1)	10	01	11	00
Квазиэквивалентный код 4 (1,2)	10	10	11	00
Квазиэквивалентный код 5 (2,1)	11	00	10	01
Квазиэквивалентный код 6 (1,2)	11	00	01	10

Согласно (4.12) последовательность кодовых отличий в данном случае имеет вид (0, 2, 1, 1).

Суммируя полученные результаты, укажем для сравнения последовательности отличий, полученные в (4.10), (4.11) и (4.12) Соответственно имеем: (0, 1, 1, 2), (0, 1, 1, 2) и (0, 2, 1, 1).

На первый взгляд может показаться, что имеющиеся отличия не имеют существенного значения. Однако, как только их сопоставишь с элементами множества Q , эти отличия демонстрируют заметное влияние на помехоустойчивость передаваемых сигналов, особенно в условиях действия помех, приводящих к ошибкам разной кратности, а также структуры пространства Q .

Покажем это на примере, изображенном на рисунке 4.2, который дополняет рисунок 4.1.

Рисунок 4.2 - Различные типы биекции пространств G и Q

На рисунке 4.2 изображены три типа биекции, представленные в таблицах (4.3), (4.4) и (4.5) для некоторой «замкнутой» структуры в форме круга с радиусом R . В общем виде это может быть и иная фигура, однако особенностью их является «замкнутость» конструкции, что, как будет показано ниже, широко распространено в сигнально-кодowych конструкциях при многократных методах дискретной модуляции.

Для взвешенного кода расстояние между точками A и B , которое обозначим через d_{AB}^{B3g} , равно $d_{AB}^{B3g} = 1$. Далее, используя аналогичные обозначения, имеем: $d_{BC}^{B3g} = 2$, $d_C^{B3g} = 1$ и $d_{DA}^{B3g} = 2$.

Для кода Грэя такая же последовательность кодовых расстояний имеет вид: $d_{AB}^{Грей} = 1$, $d_{BC}^{Грей} = 1$, $d_C^{Грей} = 1$ и $d_{DA}^{Грей} = 1$.

Для третьего типа биекции последовательность кодовых расстояний имеет вид: $d_{AB}^{3-ий} = 2$, $d_{BC}^{3-ий} = 1$, $d_C^{3-ий} = 2$ и $d_{DA}^{3-ий} = 1$, что фактически эквивалентно взвешенному коду с очевидным смещением «по кругу», что представлено на рисунке 4.2.

Предположив, что указанные расстояния являются мерой искажений или мерой числа неверно принятых элементов $g_i \in G$, восстанавливаемых из $q_i \in Q$ на приеме, можно сравнить выигрыш от «хорошего» выбора биекции между G и Q по сравнению с «плохим».

В данном примере «хороший» выбор соответствует коду Грэя и означает возможное суммарное количество неверно принятых элементов $g_i \in G$ из-за помех в канале связи и соответствующих искажениях элементов $q_i \in Q$. Это количество равно 4.

Для взвешенного кода и «3-его кода» аналогичная величина равна 6, т.е. на 50% больше, чем для кода Грэя.

Данный пример показывает, важность проведения аналогичных исследований для существующих и перспективных сигнально-кодowych конструкций при различных многократных методах дискретной модуляции.

2.3. Искажения при четырехкратных методах дискретной модуляции

Исследуем особенности эффективного сопоставления кодовых комбинаций на примере двумерных сигналов с фазовой и амплитудно-фазовой манипуляциями.

Согласно [17] иллюстрацией двумерной фазовой модуляцией можно рассматривать рисунок 4.2.

Мысленно дополним рисунок 4.2 точкой O , расположенной в центре квадрата и соответственно окружности, которую можно описать вокруг квадрата.

Из-за различных влияний в канале связи фазовый вектор OA может с вероятностью $P_{90^\circ}^{OA}$ трансформироваться в вектор OB , с вероятностью $P_{180^\circ}^{OA}$ в вектор OC и с вероятностью $P_{270^\circ}^{OA}$ в вектор OD . Продолжая аналогичные обозначения, можно записать матрицу, состоящую из вероятностей трансформаций векторов OA , OB , OC и OD , в том числе для случаев, когда трансформации не происходит.

Предположим, что вероятности трансформации на 90° , на 180° и на 270° для всех векторов OA , OB , OC и OD одинаковы, пусть также отсутствие трансформаций происходит с одинаковыми вероятностями. Тогда матрицу вероятностей трансформаций векторов запишем в следующем виде:

$$P_T = \begin{pmatrix} P_{0^\circ} & P_{90^\circ} & P_{180^\circ} & P_{270^\circ} \\ P_{270^\circ} & P_{0^\circ} & P_{90^\circ} & P_{180^\circ} \\ P_{180^\circ} & P_{270^\circ} & P_{0^\circ} & P_{90^\circ} \\ P_{90^\circ} & P_{180^\circ} & P_{270^\circ} & P_{0^\circ} \end{pmatrix} \quad (4.13)$$

Обозначим через P_A, P_B, P_C, P_D соответственно передачу векторов OA , OB , OC и OD . Тогда для первого и второго вариантов кодирования и погрешностей, задаваемых матрицей (4.1) имеем следующее значение средней величины погрешности:

$$\begin{aligned} P_{ср} = & P_A \cdot (p_{90^\circ} \cdot l + p_{180^\circ} \cdot l \cdot \sqrt{2} + p_{270^\circ} \cdot l) + P_B \cdot (p_{270^\circ} \cdot l + p_{90^\circ} \cdot l + p_{180^\circ} \cdot l \cdot \sqrt{2}) + \\ & + P_C \cdot (p_{180^\circ} \cdot l \cdot \sqrt{2} + p_{270^\circ} \cdot l + p_{90^\circ} \cdot l) + P_D \cdot (p_{90^\circ} \cdot l + p_{180^\circ} \cdot l \cdot \sqrt{2} + p_{270^\circ} \cdot l) \end{aligned} \quad (4.14)$$

При этом для первого варианта кодирования в среднем будет искажено следующее количество двоичных символов:

$$\begin{aligned} K_{cp1} &= p_A \cdot (p_{90^\circ} \cdot 1 + p_{180^\circ} \cdot 1 \cdot \sqrt{2} + p_{270^\circ} \cdot 1) + p_B \cdot (p_{270^\circ} \cdot 1 + p_{90^\circ} \cdot 1 + p_{180^\circ} \cdot 1 \cdot \sqrt{2}) + \\ &+ p_C \cdot (p_{180^\circ} \cdot 1 \cdot \sqrt{2} + p_{270^\circ} \cdot 1 + p_{90^\circ} \cdot 1) + p_D \cdot (p_{90^\circ} \cdot 1 + p_{180^\circ} \cdot 1 \cdot \sqrt{2} + p_{270^\circ} \cdot 1) = \\ &= (p_A + p_B + p_C + p_D) \cdot (p_{90^\circ} + p_{180^\circ} + p_{270^\circ}) + p_{180^\circ} \cdot (p_A + p_B + p_C + p_D) \end{aligned} \quad (4.15)$$

а для второго варианта кодирования соответственно:

$$\begin{aligned} K_{cp2} &= p_A \cdot (p_{90^\circ} \cdot 2 + p_{180^\circ} \cdot 1 \cdot \sqrt{2} + p_{270^\circ} \cdot 1) + p_B \cdot (p_{270^\circ} \cdot 2 + p_{90^\circ} \cdot 1 + p_{180^\circ} \cdot 1 \cdot \sqrt{2}) + \\ &+ p_C \cdot (p_{180^\circ} \cdot 1 \cdot \sqrt{2} + p_{270^\circ} \cdot 1 + p_{90^\circ} \cdot 2) + p_D \cdot (p_{90^\circ} \cdot 1 + p_{180^\circ} \cdot 1 \cdot \sqrt{2} + p_{270^\circ} \cdot 2) = \\ &= (p_A + p_B + p_C + p_D) \cdot (p_{90^\circ} + p_{180^\circ} + p_{270^\circ}) + p_{90^\circ} \cdot (p_A + p_C) + p_{270^\circ} \cdot (p_B + p_D) \end{aligned} \quad (4.16)$$

Сравнивая выражения (4.15) и (4.16) несложно видеть, что абсолютное отличие в среднем числе неверно восстановленных двоичных символов равно

$$\begin{aligned} K_{cp2} - K_{cp1} &= p_{90^\circ} \cdot (p_A + p_C) + p_{270^\circ} \cdot (p_B + p_D) - p_{180^\circ} \cdot (p_A + p_B + p_C + p_D) = \\ &= (p_{90^\circ} - p_{180^\circ}) \cdot (p_A + p_C) + (p_{270^\circ} - p_{180^\circ}) \cdot (p_B + p_D) \end{aligned} \quad (4.17)$$

Выражение (4.17) показывает, что в зависимости от значений вероятностей p_{90° , p_{180° и p_{270° , а также p_A, p_B, p_C и p_D предпочтение может быть отдано либо одному способу сопоставления, либо другому.

Пусть из-за влияния помех плотность вероятности возникновения фазового сдвига относительно номинального положения для всех исходных векторов OA , OB , OC и OD одинакова. Обозначим ее через $w(x)$, где величина x меняется от -360°

до $+360^\circ$. Тогда

$$p_{90^\circ} = \int_{45^\circ}^{135^\circ} W(x)dx + \int_{-225^\circ}^{-315^\circ} W(x)dx, \quad p_{180^\circ} = \int_{135^\circ}^{225^\circ} W(x)dx + \int_{-135^\circ}^{-225^\circ} W(x)dx,$$

$$p_{1270^\circ} = \int_{225^\circ}^{315^\circ} W(x)dx + \int_{-45^\circ}^{-135^\circ} W(x)dx.$$

Предположим, что плотность $w(x)$ распределена по нормальному центрированному закону, т.е.

$$W(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{x^2}{2\sigma^2}} \quad (4.18)$$

Выражение (4.17) после подстановки (4.18) и предположения равенства вероятностей P_A, P_B, P_C, P_D примет вид:

$$K_{cp2} - K_{cp1} = 2 \cdot \left(\int_{45^\circ}^{135^\circ} W(x) dx + \int_{225^\circ}^{315^\circ} W(x) dx - 2 \cdot \int_{135^\circ}^{225^\circ} W(x) dx \right) \quad (4.19)$$

Если усть, что выражение (4.18) можно рассматривать в пределах $(-\infty, +\infty)$, выражение (4.19) примет вид:

$$K_{cp2} - K_{cp1} = 2 \cdot \sum_{i=0}^{\infty} \left(\int_{45^\circ+i \cdot 360^\circ}^{135^\circ+i \cdot 360^\circ} W(x) dx + \int_{225^\circ+i \cdot 360^\circ}^{315^\circ+i \cdot 360^\circ} W(x) dx - 2 \cdot \int_{135^\circ+i \cdot 360^\circ}^{225^\circ+i \cdot 360^\circ} W(x) dx \right) \quad (4.20)$$

Определим также относительное отличие в среднем числе неверно восстановленных двоичных символах, как:

$$\delta_{cp} = \frac{K_{cp2} - K_{cp1}}{K_{cp2} + K_{cp1}} = \frac{\sum_{i=0}^{\infty} \left(\int_{45^\circ+i \cdot 360^\circ}^{135^\circ+i \cdot 360^\circ} W(x) dx + \int_{225^\circ+i \cdot 360^\circ}^{315^\circ+i \cdot 360^\circ} W(x) dx - 2 \cdot \int_{135^\circ+i \cdot 360^\circ}^{225^\circ+i \cdot 360^\circ} W(x) dx \right)}{\sum_{i=0}^{\infty} \left(\int_{45^\circ+i \cdot 360^\circ}^{135^\circ+i \cdot 360^\circ} W(x) dx + \int_{225^\circ+i \cdot 360^\circ}^{315^\circ+i \cdot 360^\circ} W(x) dx + 2 \cdot \int_{135^\circ+i \cdot 360^\circ}^{225^\circ+i \cdot 360^\circ} W(x) dx \right)} \quad (4.21)$$

На рисунке 7 показана зависимость получаемого выигрыша $(K_{cp2} - K_{cp1})$ для различных значений среднеквадратичного отклонения.

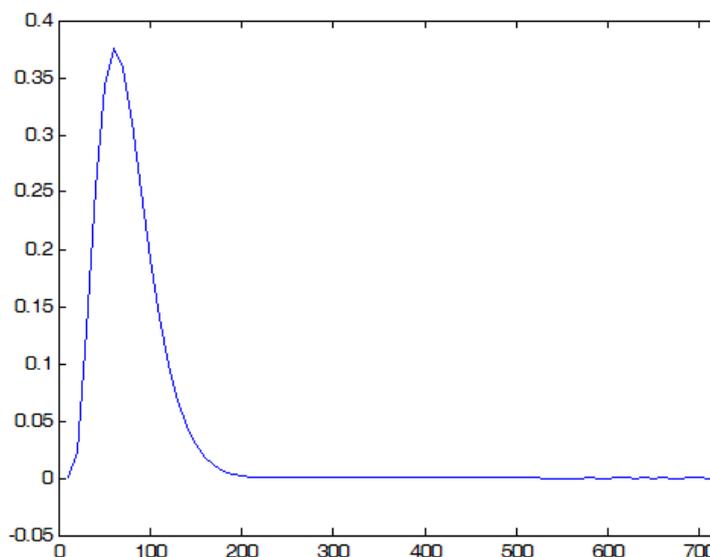


Рисунок 4.3 - Зависимость выигрыша $(K_{cp2} - K_{cp1})$ от среднеквадратичного отклонения.

Для данного примера с условиями, рассмотренными в [2], полагая $P_{90^\circ} = P_{270^\circ}$ и $P_{90^\circ} > P_{180^\circ}$, очевидно, что первый способ сопоставления (код Грэя) приводит в

среднем к меньшему числу неверно восстановленных двоичных символов, т.к. выражение (44) всегда больше нуля. При этом максимум достигается при $\sigma = 61,01847538$. Однако в общем случае это утверждение не верно для иных распределений $W(x)$.

Исследуем также двумерный вариант амплитудно-фазовой модуляции, показанной на рисунке 4.4.

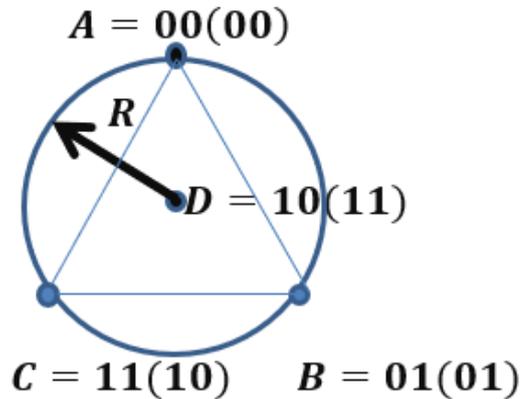


Рисунок 4.4 - Диаграмма амплитудно-фазовой модуляции

Согласно первому варианту, сопоставление осуществляется по коду Грэя. Второй вариант соответствует взвешенному коду.

Полагая геометрические расстояния между точками A, B, C, D эквивалентом погрешностей, возникающих в случае ошибки, можно построить матрицу L_4 , аналогичную (4.1).

$$L_4 = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{pmatrix} 0 & R\sqrt{3} & R\sqrt{3} & R \\ R\sqrt{3} & 0 & R\sqrt{3} & R \\ R\sqrt{3} & R\sqrt{3} & 0 & R \\ R & R & R & 0 \end{pmatrix} & \end{matrix} \quad (4.18)$$

Определим аналогично (4.13) матрицу вероятностей трансформаций:

$$P_T = \begin{pmatrix} P_{A-A} & P_{A-B} & P_{A-C} & P_{A-D} \\ P_{B-A} & P_{B-B} & P_{B-C} & P_{B-D} \\ P_{C-A} & P_{C-B} & P_{C-C} & P_{C-D} \\ P_{D-A} & P_{D-B} & P_{D-C} & P_{D-D} \end{pmatrix} \quad (4.19)$$

И запишем среднее число неверно принятых двоичных символов при демодуляции. Для первого варианта (код Грэя) имеем:

$$K_{cp1} = p_A \cdot (p_{A-B} \cdot 1 + p_{A-C} \cdot 2 + p_{A-D} \cdot 1) + p_B \cdot (p_{B-A} \cdot 1 + p_{B-C} \cdot 1 + p_{B-D} \cdot 2) + \\ + p_C \cdot (p_{C-A} \cdot 2 + p_{C-B} \cdot 1 + p_{C-D} \cdot 1) + p_D \cdot (p_{D-A} \cdot 1 + p_{D-B} \cdot 2 + p_{D-C} \cdot 1) \quad (4.20)$$

Для второго варианта (взвешенный код) соответственно:

$$K_{cp2} = p_A \cdot (p_{A-B} \cdot 1 + p_{A-C} \cdot 1 + p_{A-D} \cdot 2) + p_B \cdot (p_{B-A} \cdot 1 + p_{B-C} \cdot 2 + p_{B-D} \cdot 1) + \\ + p_C \cdot (p_{C-A} \cdot 1 + p_{C-B} \cdot 2 + p_{C-D} \cdot 1) + p_D \cdot (p_{D-A} \cdot 2 + p_{D-B} \cdot 1 + p_{D-C} \cdot 1) \quad (4.21)$$

Определим абсолютное значение отличий

$$K_{cp2} - K_{cp1} = p_A \cdot (p_{A-D} - p_{A-C}) + p_B \cdot (p_{B-C} - p_{B-D}) + \\ + p_C \cdot (p_{C-B} - p_{C-A}) + p_D \cdot (p_{D-A} - p_{D-B}) \quad (4.22)$$

Можно предположить, что

$$p_A = p_B = p_C = p_D,$$

$$p_{A-D} = p_{B-D} = p_{C-D} = p_{D-A} = p_{D-B} = p_{D-C} \text{ и}$$

$$p_{A-B} = p_{A-C} = p_{B-A} = p_{B-C} = p_{C-A} = p_{C-B},$$

тогда выражение (4.22) можно упростить

$$K_{cp2} - K_{cp1} = 0 \quad (4.23)$$

Результат означает, что рассматриваемые способы сопоставления эквивалентны и код Грэя, по сравнению с рассмотренным выше случаем не имеет преимуществ.

Если указанные выше условия равенства вероятностей не выполняются, то данный вывод требует уточнений.

2.4. Искажения при многократных методах дискретной модуляции

Изложенный выше метод анализа биекции пространств, при сопоставлении элементов $g_i \in G$, принадлежащих пространству Хэмминга, с элементами $q_i \in Q$ на передающей стороне, и обратными преобразованиями на приеме, можно применить для проведения аналогичного анализа для различных методов многократной дискретной модуляции большего размера.

Основная проблема в данном случае – это быстро нарастающая размерность решаемой задачи, согласно формулам (4.7) и (4.9), что не позволяет «вручную» методом последовательного перебора рассмотреть все возможные варианты. Кроме того важно учитывать распределение вероятностей возникновения ошибок различной кратности, что имеет существенное значение при сравнении типов биекции пространств G и Q .

В этой связи исследуем наиболее часто встречающиеся многократные методы дискретной модуляции с замкнутой концентрической структурой примеры которой можно видеть на рисунке 1.7а и на рисунке 1.8а.

Поскольку именно распределение вероятностей ошибочной трансформации на приемной стороне элементов $q_i \in Q$ в элементы $g_i \in G$ из-за помех, действующих в канале связи и вносящих искажения в элементы $q_i \in Q$, является определяющим в плане оптимизации типа биекции, то проведем аналитические исследования для случая возникновения однократных ошибок, вероятность которых много больше вероятности ошибок большей кратности.

Представим на рисунке 4.5 диаграмму, иллюстрирующую замкнутую концентрическую структуру, состоящую из N элементов $q_0, q_1, \dots, q_{N-2}, q_{N-1}$.

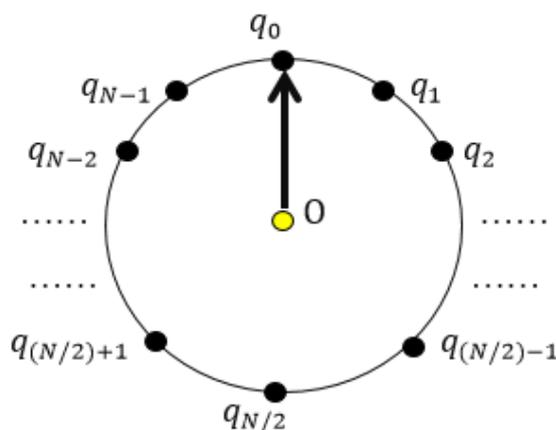


Рисунок 4.5 - Замкнутая концентрическая структура многократной дискретной модуляции

Выберем в качестве исходной точки q_0 . Относительно нее можно записать матрицу расстояний, положив расстояние между любой парой элементов равной

единице, отождествив это с соответствующим углом поворота вектора из точки O к $q_0, q_1, \dots, q_{N-2}, q_{N-1}$.

При этих условиях по аналогии с (4.1) имеем:

$$L_N = \begin{pmatrix} 0 & 1 & 2 & \dots & (N/2) - 1 & (N/2) & (N/2) - 1 & \dots & 3 & 2 & 1 \\ 1 & 0 & 1 & 2 & \dots & (N/2) - 1 & (N/2) & (N/2) - 1 & \dots & 3 & 2 \\ 2 & 1 & 0 & 1 & 2 & \dots & (N/2) - 1 & (N/2) & (N/2) - 1 & \dots & 3 \\ & & & & & \dots & \dots & \dots & \dots & \dots & \\ 1 & 2 & 3 & \dots & (N/2) - 1 & (N/2) & (N/2) - 1 & \dots & 2 & 1 & 0 \end{pmatrix} \quad (4.24)$$

Далее установив вероятность возникновения каждого элемента в матрице (4.24), можно установить наилучший тип биекции между множествами G и Q .

Введем еще дополнительные ограничения, полагая, что ошибочное преобразование на один элемент, когда $q_i = q_{i+1}$ или $q_i = q_{i-1}$, намного более вероятно, чем какое-либо иное ошибочное преобразование, и, следовательно, ими можно пренебречь.

В этих условиях, полагая $g_0 = (000\dots 00)$, можно построить код Грэя, $g_0 = (000\dots 00)$, $g_1 = (000\dots 01)$, $g_2 = (000\dots 11)$, ..., $g_{N-1} = (100\dots 00)$. Важной характеристикой этого кода является то, что между любой парой кодовых комбинаций кодовое расстояние равно 1. Следовательно, поворот вектора на рисунке 4.4 из положения O_{q_i} в положение $O_{q_{i+1}}$ или $O_{q_{i-1}}$ будет соответствовать кодовому расстоянию между комбинациями g_i и g_{i+1} или между g_i и g_{i-1} величиной равной 1.

Поскольку на окружности на рисунке 4.4 расположено ровно N точек, то, рассматривая равновероятные шаги от g_i к g_{i+1} и, соответственно, от g_i к g_{i-1} , получаем суммарное количество битовых ошибок из-за неверно принятых элементов равное $2N = 2^{n+1}$.

В отличие от кода Грэя, когда между соседними кодовыми комбинациями всегда кодовое расстояние равно единице, можно построить код, который назовем код АнтиГрэя, при котором расстояние между соседними кодовыми комбинациями будет принимать максимально возможное значение.

Для реализации кода АнтиГрэй следует в коде Грэя после первой комбинации $g_0 = (000\dots00)$ в качестве следующей кодовой комбинации взять ее инверсию, т.е. $\overline{g_0} = (111\dots11)$. В результате кодовое расстояние между g_0 и $\overline{g_0}$ составит величину равную n .

Далее между комбинацией $\overline{g_0} = (111\dots11)$ и следующей комбинацией кода Грэя $g_1 = (000\dots01)$ кодовое расстояние будет равно $(n-1)$.

Продолжая дальше, аналогично строя комбинацию для кода АнтиГрэя $\overline{g_1} = (111\dots10)$, получим кодовое расстояние между g_1 и $\overline{g_1}$ равное n . И далее между $\overline{g_1}$ и комбинацией кода Грэя g_2 , получаем кодовое расстояние равное $(n-1)$.

Продолжая дальнейшие построения, получаем, что для кода АнтиГрэя, суммарное отличие из-за однократных ошибок побитно составит величину равную $N(2n-1) = (2n-1)2^n$.

Сравнивая полученное значение с тем, что было при коде Грэя, получаем оценку диапазона, в котором будут находиться те или иные типы биекции при преобразовании элементов $g_i \in G$ в $q_i \in Q$.

В данном случае можно констатировать, что код АнтиГрэя в рассматриваемой ситуации может внести по сравнению с кодом Грэя в $\frac{(2n-1)}{2} \approx n$ раз больше битовых ошибок.

Для еще одного сравнения с часто используемым кодом рассмотрим взвешенный код. В этом случае:

$$g_0 = (000\dots01) \quad g_1 = (000\dots01), \quad g_2 = (000\dots01), \quad g_3 = (000\dots10), \dots, \quad g_{N-1} = (111\dots11).$$

Учитывая алгоритм построения взвешенного кода можно для замкнутой концентрической конструкции указать сумму битовых ошибок $4 \cdot n + \sum_{i=1}^{n-1} 2^{i+1}$.

Для более наглядного сравнения представим значения числа возможных битовых ошибок для исследованных выше кодов.

Данные таблицы 4.6 свидетельствуют о том, что по сравнению с кодом Грэя число битовых ошибок в коде АнтиГрэя практически линейно нарастают с увеличением длины кодовой комбинации.

Таблица 4.6 Число битовых ошибок при коде Грэя, АнтиГрэя и взвешенном коде

n	1	2	3	4	5	6	7	8	9
Код Грэя	4	8	16	32	64	128	256	512	1024
Код АнтиГрэя	4	12	40	112	288	704	1664	3840	8704
Взвешенный код	4	12	28	60	124	252	508	1020	2044

При взвешенном коде число битовых ошибок с ростом n стремятся к удвоенному числу битовых ошибок по сравнению с кодом Грэя.

2.5. Выводы по разделу

Структура сигнально кодовой конструкции определяет тип биекции множеств G и Q при которой можно получить минимальное число битовых ошибок для $q_i \in Q$ заданного распределения вероятности трансформации элементов $g_i \in G$ в элементы.

Распределение вероятностей трансформации элементов $g_i \in G$ в элементы $q_i \in Q$ является определяющим при выборе оптимального типа биекции, минимизирующем битовую ошибку.

Замкнутая концентрическая структура многократной дискретной модуляции влияет на выбор типа биекции по сравнению с разомкнутой структурой.

При превалирующем значении вероятности однократной ошибки для значительного числа типов биекции можно рекомендовать код Грэя и его аналоги, число которых равно $\frac{(2^n)!}{(2^n) \cdot n!}$. В тоже время для ряда сигнально кодовых

конструкций код Грэя либо не дает преимущества по сравнению, например, со взвешенным кодом, либо может дать худшие результаты.

Если распределение вероятности возникновения ошибок при трансформации элементов $g_i \in G$ в элементы $q_i \in Q$, не позволяет пренебрегать вероятностью возникновения двукратных, трехкратных и т.д. ошибок, то следует провести специальные исследования, согласно предложенному методу, для выбора наилучшего типа биекции.

5. РАЗРАБОТКА ПРОГРАММЫ РАСЧЕТА ДОПОЛНИТЕЛЬНЫХ ИСКАЖЕНИЙ ДЛЯ РАЗЛИЧНЫХ МОДЕЛЕЙ ОШИБОК.

5.1. Постановка задачи

Выше было показано, что при передаче сигналов от источника к получателю происходят их многократные преобразования. При этом различные типы биекции пространств, в которых отображаются сигналы, позволяют минимизировать дополнительные искажения, возникающие из-за ошибок в дискретном канале связи, либо уменьшить вероятность ошибки на бит, при возникновении искажений в непрерывном канале связи. Данная оптимизация выбора типа биекции во многом определяется моделью ошибок или соответственно моделью действующих помех, что не всегда можно отобразить в виде аналитического выражения, с помощью которого в дальнейшем можно провести необходимые преобразования и расчеты.

В этой связи целесообразно разработать программную модель, в которой можно задавать различные виды ошибок, для которых далее проводить расчеты оптимальных типов биекции пространств, минимизирующих вероятность битовых ошибок, либо соответственно величину дополнительных искажений.

5.2. Описание программного алгоритма

Данный алгоритм удалось реализовать через программные модули MS Excell.

На начальном этапе программа считывает данные, которые пользователь загрузил в файл .xls.

Если условие равенства нарушается, то алгоритм производит нормировку матрицы, тем самым приравнивая сумму вероятностей в строке к определенному коэффициенту.

Затем алгоритм считает максимальное количество знаков после «запятой» в дробных элементах матрицы и домножает их на число с таким количеством нулей,

чтобы числа оказались целыми. Это сделано для упрощения подсчетов пользователем и минимизации затрат ресурсов вычислительного устройства.

Далее программное приложение начинает проводить перестановки различных вариантов кодирования, начиная с перестановок столбцов и строк «хвоста».

Таким образом, с помощью данной простейшей программной реализации пользователь может найти «наилучший» вариант кодирования по принципу наименьшей дисперсии и минимального коэффициента искажений.

Пользователь имеет возможность анализа способов кодирований с количеством символов на элемент сообщения от 2 до произвольного значения, учитывая мощность вычислительной машины.

Программное обеспечение для выполнения расчётов представляет собой специализированный модуль для Microsoft Excel. Диаграмма классов модуля приведена ниже на рисунке 5.1.

Список классов разработанного ПО представлен ниже:

1. SciExcel.ExcelMatrix (Матрица, размещённая на листе Microsoft Excel)
2. SciExcel.Mathematics.Matrix (Matrix Library .Net v2.0 By Anas Abidi, 2004)
3. SciExcel.Mathematics.MatrixDeterminentZero (Исключение - нулевой определитель матрицы)
4. SciExcel.Mathematics.MatrixDimensionException (Исключение - некорректная размерность матрицы)
5. SciExcel.Mathematics.MatrixLibraryExceptions (Исключение - ошибка библиотеки)
6. SciExcel.Mathematics.MatrixNotSquare (Исключение - матрица не квадратная)
7. SciExcel.Mathematics.MatrixNullException (Исключение - матрица не задана)
8. SciExcel.Mathematics.MatrixSingularException (Исключение - матрица не может быть вектором)
9. SciExcel.MyRibbon (Лента)
10. SciExcel.Mathematics.Pair (Значение и его индекс в последовательности)

11. SciExcel.Mathematics.Permutation (Построение перестановок)
12. SciExcel.Mathematics.Permutator (Перестановки из n элементов)
13. SciExcel.ProcessForm (Экранная форма)
14. SciExcel.Mathematics.SciMatrix (Матрица с прикладными расчетами)
15. SciExcel.ThisAddIn (Модуль для Microsoft Excel)
16. SciExcel.Mathematics.VectorDimensionException (Исключение - матрица должна быть вектором из трех элементов)

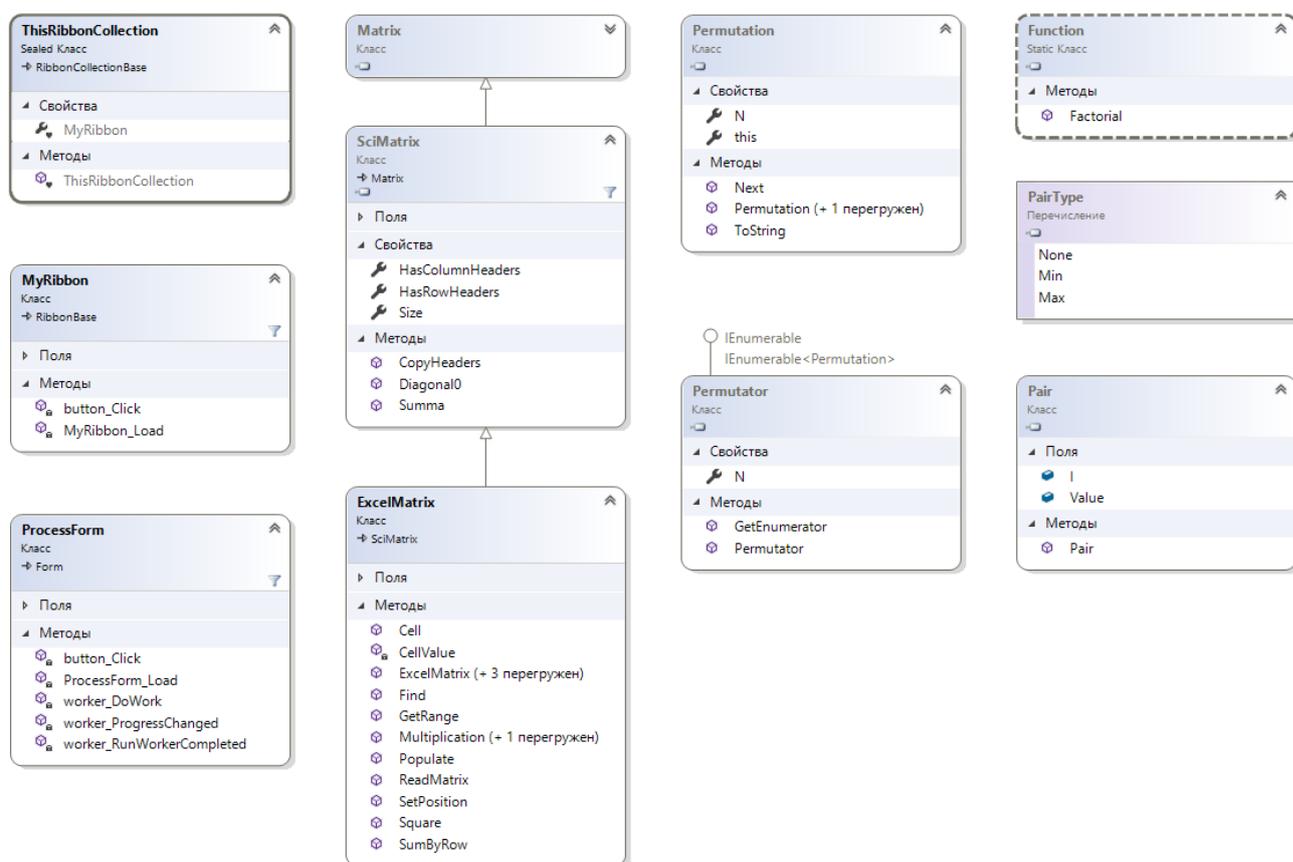


Рисунок 5.1 - Диаграмма классов программного обеспечения

Графы наследования класса SciExcel.ExcelMatrix представлены на рисунке 5.2.

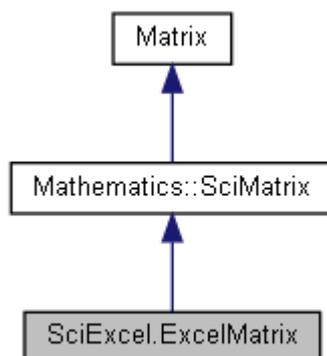


Рисунок 5.2 - Граф наследования класса

В данном классе описана функция `SciExcel.ExcelMatrix.ExcelMatrix`, задающая квадратную матрицу вероятностей размера $N \times N$. Граф вызова функции представлен на рисунке 5.3.

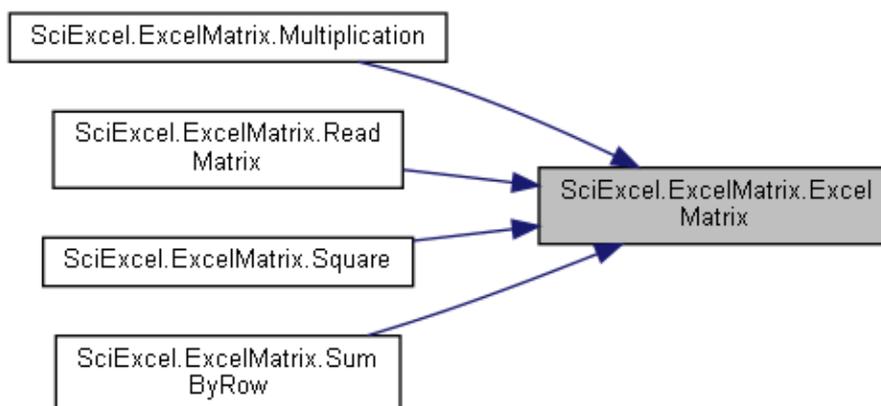


Рисунок 5.3 – Граф вызова функции SciExcel.ExcelMatrix.ExcelMatrix

Функция `SciExcel.ExcelMatrix.Cell` производит всевозможные перестановки в матрице вероятностей. Граф ее вызова представлен на рисунке 5.4.

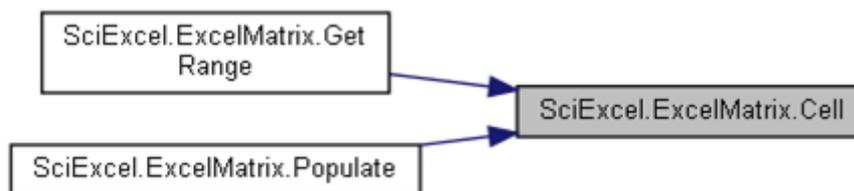


Рисунок 5.4 - Граф вызова функции SciExcel.ExcelMatrix.Cell

Функция `SciExcel.ExcelMatrix.Find` отвечает за точный поиск строки на текущем листе. Граф ее вызова представлен на рисунке 5.5.

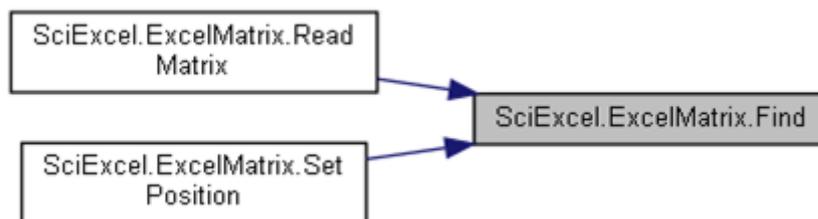


Рисунок 5.5. – Граф вызова функции `SciExcel.ExcelMatrix.Find`

Функция `SciExcel.ExcelMatrix.Multiplication` отвечает за умножение матриц, вектора и построчное суммирование (при поиске показателе эффективности варианта кодирования). Граф вызова функции представлен на рисунке 5.6.

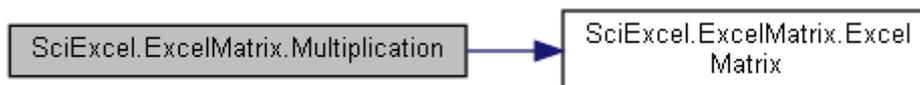


Рисунок 5.6 – Граф вызова функции `SciExcel.ExcelMatrix.Multiplication`

Функция `SciExcel.ExcelMatrix.Populate` производит выгрузку данных на лист Excel. Граф ее вызова представлен на рисунке 5.7.



Рисунок 5.7 – Граф вызова функции `SciExcel.ExcelMatrix.Populate`

Функция `SciExcel.ExcelMatrix.ReadMatrix` производит поиск заголовка и чтение матрицы. Граф ее вызова представлен на рисунке 5.8.

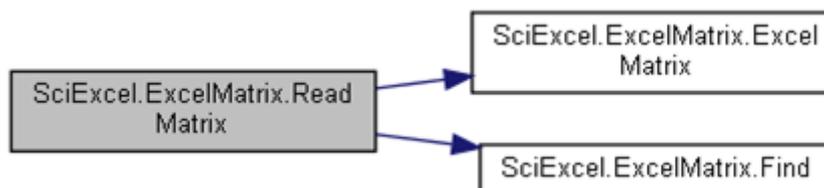


Рисунок 5.8 – Граф вызова функции `SciExcel.ExcelMatrix.ReadMatrix`

Функция `SciExcel.ExcelMatrix.SetPosition` устанавливает место размещения матрицы на листе Excel. Граф ее вызова представлен на рисунке 5.9.



Рисунок 5.9 – Граф вызова функции `SciExcel.ExcelMatrix.SetPosition`

Функция `SciExcel.ExcelMatrix.SumByRow` суммирует матрицу построчно. Граф ее вызова представлен на рисунке 5.10.



Рисунок 5.10 – Граф вызова функции `SciExcel.ExcelMatrix.SumByRow`

Класс `Matrix Library .Net v2.0` By Anas Abidi, 2004.

Класс предназначен для осуществления операций матричной алгебры. Класс предоставляет множество методов, в том числе статических, для выполнения различных операций над матрицами вещественных чисел. Для удобства разработки для работы с матрицами перекрыты операторы '+', '-', '*'.

Класс `SciExcel.Mathematics.MatrixDeterminentZero`

Исключение - нулевой определитель матрицы

Граф наследования класса `SciExcel.Mathematics.Matrix` представлен на рисунке 5.11.

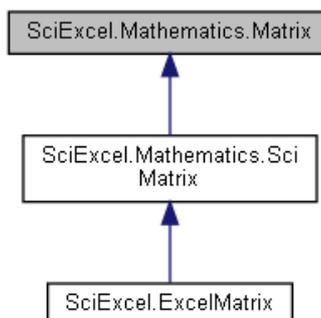


Рисунок 5.11 – Граф наследования класса `SciExcel.Mathematics.Matrix`

5.3. Пользовательский интерфейс, установка ПО

Запустить файл SETUP.EXE из каталога SCIEXCEL\PACKAGE. При выводе сообщения о неизвестном издателе нажать «Установить».

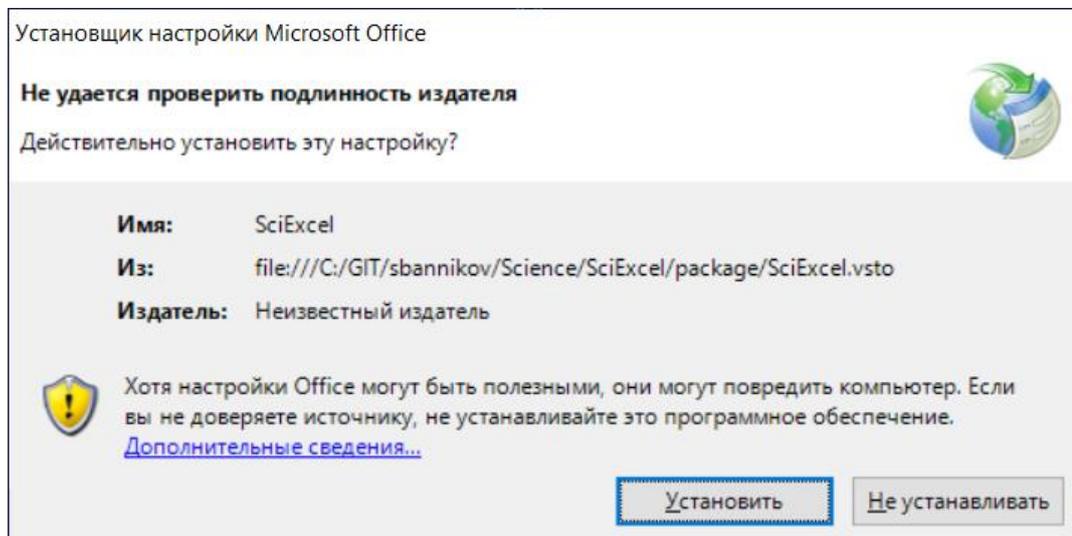


Рисунок 5.12 – Процесс установки ПО

Открыть файл "Расчетные данные.xlsx".

На ленте Excel выбрать пункт «Наука» и нажать кнопку «Расчет». Выполняется поиск исходных данных (как описано в разделе «Порядок размещения данных на листе») и расчет данных. Столбец «Сумма по строке» заполнится автоматически.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Матрица вероятностей				Сумма по строке		Матрица рисков						
2	0,94995	0,012352	0,012352	0,00016062			0	2	3	4			
3	0,012352	0,94995	0,00016062	0,012352			5	0	7	8			
4	0,012352	0,00016062	0,94995	0,012352			9	10	0	12			
5	0,00016062	0,012352	0,012352	0,94995			13	14	15	0			

Рисунок 5.13 – Интерфейс пользователя с расчетами

После успешного расчета выводится сообщение, представленное на рисунке 5.14.

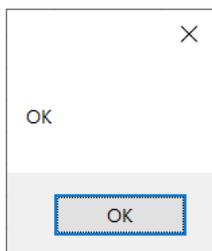


Рисунок 5.14 – Сообщение о верном вводе данных

Если в ячейках будут некорректные данные (не числа), они выделяются красным цветом, как показано на рисунке 5.15.

 A screenshot of an Excel spreadsheet. The active cell is B3, containing the value 0,94995. The spreadsheet contains several tables. The first table, 'Матрица вероятностей', has 5 rows and 4 columns. The second table, 'Сумма по строке', has 5 rows and 1 column. The third table, 'Матрица рисков', has 5 rows and 4 columns. The cell B2 in the first table is highlighted in red, indicating an error.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Матрица вероятностей				Сумма по строке		Матрица рисков						
2	0,94995	0,012352	0,012352	0,00016062	65535		0	2	3	4			
3	0,012352	0,94995	0,00016062	0,012352	0,97481462		5	0	7	8			
4	0,012352	0,00016062	0,94995	0,012352	0,97481462		9	10	0	12			
5	0,00016062	0,012352	0,012352	0,94995	0,97481462		13	14	15	0			

Рисунок 5.15 – Сообщение об ошибке ввода

Порядок размещения данных на листе:

- 1) Матрица вероятностей должна располагаться непосредственно под заголовком «Матрица вероятностей» и может иметь любой размер. Следующая ячейка после конца матрицы должна быть пустой.
- 2) Матрица искажений должна располагаться непосредственно под заголовком «Матрица искажений» и может иметь любой размер. Следующая ячейка после конца матрицы должна быть пустой.
- 3) Сумма по строке матрицы вероятностей выводится в следующий после матрицы столбец, поверх тех данных, которые там есть.
- 4) Для размещения различных расчетов (различных наборов исходных данных) используются различные листы и/или книги Excel.

5.4. Тестовые данные для матрицы 4x4

Для тестирования программного приложения с матрицами из 16-ти элементов была выбрана матрица вероятностей, представленная в таблице 5.1.

В таблице 5.2 представлена выбранная матрица искажений, а в таблице 5.3 представлена нормированная матрица вероятностей.

Таблица 5.1 – Матрица вероятностей 4x4

Матрица вероятностей				Сумма по строке
0,94995	0,012352	0,012352	0,00016062	0,97481462
0,012352	0,94995	0,00016062	0,012352	0,97481462
0,012352	0,00016062	0,94995	0,012352	0,97481462
0,00016062	0,012352	0,012352	0,94995	0,97481462

Таблица 5.2 – Матрица вероятностей 4x4

Матрица искажений			
0	2	3	4
5	0	7	8
9	10	0	12
13	14	15	0

Таблица 5.3 – Нормированная матрица вероятностей

Нормированная матрица вероятностей				Sum	Rk
0,974492976	0,012671127	0,012671127	0,00016477	1	0,25
0,012671127	0,974492976	0,00016477	0,012671127	1	0,25
0,012671127	0,00016477	0,974492976	0,012671127	1	0,25
0,00016477	0,012671127	0,012671127	0,974492976	1	0,25

В таблице 5.3 ячейка Сумм - это сумма элементов каждой строки матрицы, а Rk-значения элементов вектора вероятностей появления сообщения в канале связи.

После расчетов программа выдала 24 варианта перестановки матрицы вероятностей, а также результирующую информацию о показателях

эффективности наилучшего и наихудшего вариантов кодирования. Показатели представлены в таблице 5.4.

Как видно из таблицы 5.4, максимальное значение коэффициента искажений было обнаружено у 5го варианта кодирования, а максимальная дисперсия у 1го варианта перестановки.

Таблица 5.4– Результат работы программы

Максимальное значение коэффициента искажений	0,216809302
Вариант кодирования	5
Минимальное значение коэффициента искажений	0,216809101
Вариант кодирования	1
Максимальное значение дисперсии	2,336252607
Вариант кодирования	1
Минимальное значение дисперсии	2,136150889
Вариант кодирования	5

5.5. Тестовые данные для матрицы 8x8

Для тестирования программного приложения с матрицами из 64-х элементов были выбраны матрица вероятностей, представленные в таблицах 5.5,5.7, 5.9.

Они соответствуют кодовым комбинациям Натурального кода, кода Грея и модифицированного кода Грея со сдвигом не на один символ, а на два.

Матрицы могут быть не нормированы, т.е. сумма элементов построчно не равна единице, поэтому программный алгоритм осуществляет проверку и нормирует матрицы.

Нормированные матрицы представлены в таблицах 5.6, 5.8,5.10.

Таблица 5.5 – Матрица вероятностей Натурального кода

	000	001	010	011	100	101	110	111
000	-	0,000998001	0,000998001	0,000000999	0,000998001	0,000000999	0,000000999	0,000000001
001	0,000998001	-	0,000000999	0,000998001	0,000000999	0,000998001	0,000000001	0,000000999
010	0,000998001	0,000000999	-	0,000998001	0,000000999	0,000000001	0,000998001	0,000000999
011	0,000000999	0,000998001	0,000998001	-	0,000000001	0,000000999	0,000000999	0,000998001
100	0,000998001	0,000000999	0,000000999	0,000000001	-	0,000998001	0,000998001	0,000000999
101	0,000000999	0,000998001	0,000000001	0,000000999	0,000998001	-	0,000000999	0,000998001
110	0,000000999	0,000000001	0,000998001	0,000000999	0,000998001	0,000000999	-	0,000998001
111	0,000000001	0,000000999	0,000000999	0,000998001	0,000000999	0,000998001	0,000998001	-

Таблица 5.6 – Нормированная матрица вероятностей Натурального кода

	000	001	010	011	100	101	110	111
000	-	0,332999889	0,332999889	0,000333333	0,332999889	0,000333333	0,000333333	0,000000334
001	0,332999889	-	0,000333333	0,332999889	0,000333333	0,332999889	0,000000334	0,000333333
010	0,332999889	0,000333333	-	0,332999889	0,000333333	0,000000334	0,332999889	0,000333333
011	0,000333333	0,332999889	0,332999889	-	0,000000334	0,000333333	0,000333333	0,332999889
100	0,332999889	0,000333333	0,000333333	0,000000334	-	0,332999889	0,332999889	0,000333333
101	0,000333333	0,332999889	0,000000334	0,000333333	0,332999889	-	0,000333333	0,332999889
110	0,000333333	0,000000334	0,332999889	0,000333333	0,332999889	0,000333333	-	0,332999889
111	0,000000334	0,000333333	0,000333333	0,332999889	0,000333333	0,332999889	0,332999889	-

Таблица 5.7 – Матрица вероятностей кода Грея

	000	001	011	010	110	111	101	100
000	-	0,000998001	0,000000999	0,000998001	0,000000999	0,000000001	0,000000999	0,000998001
001	0,000998001	-	0,000998001	0,000000999	0,000000001	0,000000999	0,000998001	0,000000999
011	0,000000999	0,000998001	-	0,000998001	0,000000999	0,000998001	0,000000999	0,000000001
010	0,000998001	0,000000999	0,000998001	-	0,000998001	0,000000999	0,000000001	0,000000999
110	0,000000999	0,000000001	0,000000999	0,000998001	-	0,000998001	0,000000999	0,000998001
111	0,000000001	0,000000999	0,000998001	0,000000999	0,000998001	-	0,000998001	0,000000999
101	0,000000999	0,000998001	0,000000999	0,000000001	0,000000999	0,000998001	-	0,000998001
100	0,000998001	0,000000999	0,000000001	0,000000999	0,000998001	0,000000999	0,000998001	-

Таблица 5.8 – Нормированная матрица вероятностей кода Грея

	000	001	011	010	110	111	101	100
000	-	0,332999889	0,000333333	0,332999889	0,000333333	0,000000334	0,000333333	0,332999889
001	0,332999889	-	0,332999889	0,000333333	0,000000334	0,000333333	0,332999889	0,000333333
011	0,000333333	0,332999889	-	0,332999889	0,000333333	0,332999889	0,000333333	0,000000334
010	0,332999889	0,000333333	0,332999889	-	0,332999889	0,000333333	0,000000334	0,000333333
110	0,000333333	0,000000334	0,000333333	0,332999889	-	0,332999889	0,000333333	0,332999889
111	0,000000334	0,000333333	0,332999889	0,000333333	0,332999889	-	0,332999889	0,000333333
101	0,000333333	0,332999889	0,000333333	0,000000334	0,000333333	0,332999889	-	0,332999889
100	0,332999889	0,000333333	0,000000334	0,000333333	0,332999889	0,000333333	0,332999889	-

Таблица 5.9 – Матрица вероятностей модернизированного кода Грея

	000	011	110	101	100	111	010	001
000	-	0,000000999	0,000000999	0,000000999	0,000998001	0,000000001	0,000998001	0,000998001
011	0,000000999	-	0,000000999	0,000000999	0,000000001	0,000998001	0,000998001	0,000998001
110	0,000000999	0,000000999	-	0,000000999	0,000998001	0,000998001	0,000998001	0,000000001
101	0,000000999	0,000000999	0,000000999	-	0,000998001	0,000998001	0,000000001	0,000998001
100	0,000998001	0,000000001	0,000998001	0,000998001	-	0,000000999	0,000000999	0,000000999
111	0,000000001	0,000998001	0,000998001	0,000998001	0,000000999	-	0,000000999	0,000000999
010	0,000998001	0,000998001	0,000998001	0,000000001	0,000000999	0,000000999	-	0,000000999
001	0,000998001	0,000998001	0,000000001	0,000998001	0,000000999	0,000000999	0,000000999	-

Таблица 5.10 – Нормированная матрица вероятностей модернизированного кода Грея

	000	011	110	101	100	111	010	001
000	-	0,000333333	0,000333333	0,000333333	0,332999889	0,000000334	0,332999889	0,332999889
011	0,000333333	-	0,000333333	0,000333333	0,000000334	0,332999889	0,332999889	0,332999889
110	0,000333333	0,000333333	-	0,000333333	0,332999889	0,332999889	0,332999889	0,000000334
101	0,000333333	0,000333333	0,000333333	-	0,332999889	0,332999889	0,000000334	0,332999889
100	0,332999889	0,000000334	0,332999889	0,332999889	-	0,000333333	0,000333333	0,000333333
111	0,000000334	0,332999889	0,332999889	0,332999889	0,000333333	-	0,000333333	0,000333333
010	0,332999889	0,332999889	0,332999889	0,000000334	0,000333333	0,000333333	-	0,000333333
001	0,332999889	0,332999889	0,000000334	0,332999889	0,000333333	0,000333333	0,000333333	-

Расчеты проводились с двумя матрицами искажений, всего было проведено 6 расчетов. Матрица искажений представлена в таблице 5.11.

Таблица 5.11 – Матрица искажений №1

Матрица искажений							
0	10	20	30	40	50	60	70
10	0	10	20	30	40	50	60
20	10	0	10	20	30	40	50
30	20	10	0	10	20	30	40
40	30	20	10	0	10	20	30
50	40	30	20	10	0	10	20
60	50	40	30	20	10	0	10
70	60	50	40	30	20	10	0

Алгоритм программы перебирает все возможные варианты перестановок элементов в матрице вероятностей попарно (строка и столбец), начиная с хвоста нумерации.

Если матрица имеет размерность 4x4, то первой итерацией будут переставлены 3я и 2я строка и 3й и 2й столбец (нумерация начинается с 0), затем все тот же 2й столбец и 2я строка, оказавшиеся теперь «в хвосте» будут переставлены вперед «хвоста», позади них окажутся 1й и 3й столбец, 1я и 3я строка, третьим шагом поменяются местами 3й и 1й столбец и 3я и 1я строка. Затем все тот же 3й столбец и 3я строка будут переставлены вперед хвоста, а затем поменяются местами 1е и 2е столбец и строки.

Затем на первую позицию встает уже не 0й номер, а 1я строка и 1й столбец и алгоритм повторяется по кругу.

- 1) 0-1-2-3
- 2) 0-1-3-2
- 3) 0-2-1-3
- 4) 0-2-3-1
- 5) 0-3-1-2
- 6) 0-3-2-1
- 7) 1-0-2-3
- 8) 1-0-3-2
- 9) 1-2-0-3
- 10) 1-2-3-0

- 11) 1-3-0-2
- 12) 1-3-2-0
- 13) 2-0-1-3
- 14) 2-0-3-1
- 15) 2-1-0-3
- 16) 2-1-3-0
- 17) 2-3-0-1
- 18) 2-3-1-0
- 19) 3-0-1-2
- 20) 3-0-2-1
- 21) 3-1-0-2
- 22) 3-1-2-0
- 23) 3-2-0-1
- 24) 3-2-1-0

После проведения расчетов, а именно перебора всех 40320 вариантов перестановки матрицы 8x8 вероятностей для каждого из трех случаев, были получены коэффициенты эффективности, представленные в таблицах 5.12-5.14.

Таблица 5.12 – Результаты работы программы для Натурального кода

Показатель эффективности и вариант кодирования	Значение показателя эффективности и номер варианта кодирования
Максимальное значение коэффициента искажений	39,97666667
Вариант кодирования	1873
Минимальное значение коэффициента искажений	23,34333889
Вариант кодирования	8
Максимальное значение дисперсии	488,605399
Вариант кодирования	3585
Минимальное значение дисперсии	92,0001667
Вариант кодирования	25

Таблица 5.13 – Результаты работы программы для кода Грея

Показатель эффективности и вариант кодирования	Значение показателя эффективности и номер варианта кодирования
Максимальное значение коэффициента искажений	39,97666667
Вариант кодирования	1054
Минимальное значение коэффициента искажений	23,34333889
Вариант кодирования	6
Максимальное значение дисперсии	488,605399
Вариант кодирования	20245
Минимальное значение дисперсии	92,0001667
Вариант кодирования	643

Таблица 5.14 – Результаты работы программы для случайного кода

Показатель эффективности и вариант кодирования	Значение показателя эффективности и номер варианта кодирования
Максимальное значение коэффициента искажений	39,97666667
Вариант кодирования	1
Минимальное значение коэффициента искажений	23,34333889
Вариант кодирования	2357
Максимальное значение дисперсии	488,605399
Вариант кодирования	23238
Минимальное значение дисперсии	92,0001667
Вариант кодирования	2745

Анализируя данные из таблиц 5.12-5.14 можно сделать вывод о том, что для Натурального кода с использованием неравномерной матрицы искажений минимальное значение коэффициента искажений имеет вариант перестановки варианта кодирования №8, выигрыш по сравнению с перестановкой № 1873 составляет порядка 16. Для Кода Грея и случайного кода такой же выигрыш почти в два раза приходится на варианты кодирований №6 и №2357 соответственно. Таким же образом можно отследить выигрыш по дисперсии.

Для удобства анализа программный алгоритм производит свод всех вычисленных коэффициентов по принципу дерева, группируя одинаковые показатели. Пользователь может производить сортировку данных по номеру

расчета, дисперсии, коэффициенту искажений или числу перестановок. Фрагмент сводных данных представлен на рисунке 5.16.

Расчет	Матожидание СВ	Дисперсия	Число перестановок
8.1.1	23.343339	155.788996	48
8.1.1	23.343339	172.438974	48
8.1.1	23.343339	189.072302	192
8.1.1	23.343339	189.088952	96
8.1.1	23.343339	222.322307	48
8.1.1	23.343339	222.372257	192
8.1.1	23.343339	239.038885	96
8.1.1	23.343339	255.688863	96
8.1.1	23.343339	288.922218	96
8.1.1	23.343339	288.972168	192
8.1.1	23.343339	288.988818	48
8.1.1	23.343339	305.638796	48

Рисунок 5.16 – Фрагмент сводных данных по расчетам программы

5.6. Выводы по разделу

Разработана программная модель расчетов, позволяющая провести исследование и соответствующий выбор типа биекции. Для реализации алгоритма использовался высокоуровневый язык C# и программная платформа Microsoft.NET. Для матричных вычислений использовалась специализированная библиотека Matrix Library .Net v2.0 (C) Anas Abidi, 2004. Для реализации интерфейса пользователя использовались стандартные возможности Microsoft Excel, в том числе по созданию встраиваемых (add-in) модулей.

Программное приложение позволяет на основе данных матриц искажений и вероятностей провести поиск наиболее эффективного варианта кодирования, исходя из рассчитанных показателей дисперсии и коэффициента искажений. Программный алгоритм производит свод всех вычисленных коэффициентов по принципу дерева, группируя одинаковые показатели. Пользователь может производить сортировку данных по номеру расчета, дисперсии, коэффициенту искажений или числу перестановок. Приведены описание программного алгоритма, диаграммы классов ПО, описание пользовательского интерфейса разработанной программы, примеры расчетов в разработанном ПО.

ЗАКЛЮЧЕНИЕ

Добиваясь эффективной передачи сигналов от источника к получателю, помимо обеспечения минимальной скорости кодирования источника, выбора соответствующего помехоустойчивого кода, обеспечения требуемого времени передачи, сложности реализации и т.д. необходимо учитывать, что при любых преобразованиях сигналов из одного вида в другой имеется возможность оптимизации данного выбора при сопоставлении элементов исходного сигнала с его отображением.

При этом эффективность преобразования сигналов из одного вида в другой должна определяться не только скоростью кодирования, но и выбором метода сопоставления исходного сигнала и его отображения после преобразования.

Оптимизация метода отображения исходного сигнала в условиях действия помех позволяет получить заметный выигрыш по помехоустойчивости. При этом скорость кодирования остается неизменной, что является существенным фактором, поскольку достигаемый положительный эффект может быть получен без каких-либо дополнительных затрат, уменьшающих пропускную способность канала.

Выполненные в диссертационной работе исследования позволили получить зависимости эффективности преобразования сигналов в цифровой при различных значениях точности отображения в виде введенного критерия качества, оцениваемого по величине искажений элементов исходного сигнала, данные зависимости получены для общего случая (для многоуровневого непрерывно-дискретного сигнала с учетом требований к точности отображения его элементов).

Показано, что предельная эффективность преобразования различных сигналов в цифровой достигается при длине преобразуемого сигнала, стремящегося к бесконечности. А, следовательно, реально полученные значения эффективности при ограниченной длительности преобразуемого сигнала отличается от теоретически достижимой эффективности на 7-10%.

Точное непреодолимое отличие реальной эффективности от теоретически достижимой, в условиях ограниченности длительности исходного сигнала, зависит от точности отображения и вида сигнала.

Разработанный метод анализа биекции множества A и множества B , соответствующий преобразованию элементов исходного сигнала $a_i \in A$ в цифровые элементы $b_i \in B$ пространства Хэмминга, позволил провести количественные расчеты с целью выбора наилучшего метода биекции, обеспечивающего минимальную величину дополнительных искажений, возникающих в элементах $a_i \in A$ из-за ошибок в элементах $b_i \in B$.

Предложенные «хорошие» и «плохие» методы биекции дают возможность получить оценки диапазона выигрыша по минимизации дополнительных искажений, возникающих в элементах $a_i \in A$ из-за ошибок в элементах $b_i \in B$, за счет «хорошего» выбора с учетом возможностей взвешенного кодирования и кода Грэя.

Виды помех, влияющие на появление ошибок в цифровом сигнале, необходимо учитывать, поскольку распределение ошибок оказывается существенным фактором при оптимизации выбора метода отображения исходного сигнала.

Показано, что при однократных ошибках в элементах $b_i \in B$ наибольшее количество минимальных искажений в элементах $a_i \in A$ достигается при коде Грэя. Однако при этом имеются и максимально возможные искажения величиной $(2^n - 1)$, тогда как для взвешенного кода максимальная величина искажений не превосходит 2^{n-1} , т.е. практически в два раза меньше.

Важно также отметить, что среднее значение искажений для регулярной структуры элементов $a_i \in A$ в этом случае совпадает.

Разработанный метод сравнения типов биекции множества A и множества B в условиях возникновения ошибок в элементах $b_i \in B$ позволяет получить более подробное описание, а именно, спектр возникающих искажений, что дает

возможность осуществить выбор типа биекции, исходя из различных условий выбора.

Например, по принципу минимизации среднего значения возникающих дополнительных искажений, по принципу минимизации максимальной величины возникающих искажений и т.д.

В диссертации показано, что структура сигнально кодовой конструкции определяет тип биекции множеств G и Q , при которой можно получить минимальное число битовых ошибок для заданного распределения вероятности трансформации элементов $g_i \in G$ в элементы $q_i \in Q$.

При этом распределение вероятностей трансформации элементов $g_i \in G$ в элементы $q_i \in Q$ является определяющим при выборе оптимального типа биекции, минимизирующем битовую ошибку.

Показано, что замкнутая концентрическая структура многократной дискретной модуляции влияет на выбор типа биекции по сравнению с разомкнутой структурой.

При преобладающем значении вероятности однократной ошибки для значительного числа типов биекции можно рекомендовать код Грэя и его аналоги, число которых равно $\frac{(2^n)!}{(2^n) \cdot n!}$. В тоже время для ряда сигнально кодовых конструкций код Грэя либо не дает преимущества по сравнению, например, со взвешенным кодом, либо может дать худшие результаты.

С целью получения практически важного механизма выбора оптимального типа биекции с учетом различного вида возникающих ошибок в диссертации разработана программная модель расчетов, позволяющая провести данные исследования и соответствующий выбор типа биекции.

Учитывая, что в современных системах связи практически все виды сообщений преобразуются в цифровую форму и в последующем сформированные цифровые сигналы проходят различные этапы преобразований, в том числе связанных с помехоустойчивым кодированием, виды не исправленных и не

обнаруженных ошибок существенно меняются, поскольку ошибки малой кратности, и в первую очередь однократные ошибки, обнаруживаются и исправляются. А это означает, что оптимизация в условиях однократных ошибок становится неверной и требует более глубокого и обстоятельного рассмотрения.

Результаты данной работы будут безусловно полезны в этом при осуществлении оптимизации возникающих искажения с учетом всех влияющих факторов в системе в целом, а не только на отдельных этапах.

Список литературы

1. Huffman, D. A. A Method for the Construction of Minimum Redudancy Codes / D.A. Huffman // Proc. IRE. Vol. 40, September, 1952. - PP. 1098-1101.
2. Hamming, R. W. Error detecting and error correcting codes / R. W. Hamming // Bell Syst. Tech. J. 29:147-60, 1950. PP. 35-51.
3. Hamming, R. W. Coding and Information Theory / R. W. Hamming; Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980. 346 p.
4. Fano, R. Transmission of information; a statistical theory of communications / R. Fano; Cambridge, Massachusetts, M.I.T. Press, 1961.
5. Фано, Р. Передача информации. Статистическая теория связи / Пер. с англ. яз. И. А. Овсеевич, Р. Л. Добрушин; Москва: Мир, 1965. - 440 с.
6. Shannon, C. E. A symbolic analysis of relay and switching circuits / C. E. Shannon; 1938. 236 p.
7. Shannon C. E. An Algebra for Theoretical Genetics / C. E. Shannon; 1940. 465 p.
8. Shannon C. E. Communication Theory of Secrecy Systems // Bell System Technical Journal. 1949. PP. 23-65.
9. Shannon C. E. A Mathematical Theory of Communication // Bell System Technical Journal. 1948. Т. 27. С. 379—423, 623—656.
10. Shannon C. E. Communication in the presence of noise / C. E. Shannon // Proc. Institute of Radio Engineers. Jan, 1949, Vol. 37, № 1. - PP. 10—21.
11. Шеннон, К. Работы по теории информации и кибернетике / К. Шеннон; Москва: Издательство иностранной литературы, 1963. - 830 с.
12. Котельников, В. А. О пропускной способности эфира и проволоки в электросвязи / В.А. Котельников // Материалы к I Всесоюзному съезду по вопросам технической реконструкции дела связи и развития слаботочной промышленности / Всесоюзный энергетический комитет, 1933. - С. 762-770.
13. Котельников, В.А. Теория потенциальной помехоустойчивости / В.А. Котельников; Москва: Госэнергоиздат, 1956. - 152 с.

14. Котельников, В. А. Проблемы помехоустойчивой радиосвязи: радиотехнический сборник / В.А. Котельников; Москва: Госэнергоиздат, 1947. - 245 с.
15. Kotel'nikov, V. A. The Theory of Optimum Noise Immunity / V. A. Kotel'nikov; McGraw-Hill Book Co., 1959. - 140 p.
16. Зюко, А.Г. Теория передачи сигналов / А.Г. Зюко, Д.Д. Кловский, М.В. Назаров, Л.М. Финк; Москва: Радио и связь, 1986. - 304 с.
17. Зюко, А.Г. Помехоустойчивость и эффективность систем передачи информации / А.Г. Зюко; Москва: Радио и связь, 1985. - 271 с.
18. Зюко, А.Г. Эффективность систем передачи сообщений / А.Г. Зюко // Электросвязь. - 1977. - № 6. - С. 17-19.
19. Зюко, А.Г. Эффективность систем связи с корректирующими кодами / А.Г. Зюко; в кн.: Статистическая теория связи и ее практические приложения; Москва: Связь, 1979. - С. 97-101.
20. Зюко, А.Г. Методы повышения эффективности систем передачи дискретных сообщений / В.Л. Банкет, Н.Н. Курутов; в кн.: Проблемы, методы и средства Электросвязи. К.: Техника, 1980. - С. 15-20.
21. Зюко, А.Г. Системы приема фазового и относительно-фазового радиотелеграфирования порядка / А.А. Филиппов, А.И. Фалько // Труды учебных институтов связи. - 1962. - № 9. - С. 37-44.
22. Окунев, Ю.Б. Теория фазоразностной модуляции / Ю.Б. Окунев; Москва: Связь, 1979. - 215 с.
23. Окунев, Ю.Б. Цифровая передача информации фазоманипулированными сигналами / Ю.Б. Окунев; Москва: Радио и связь, 1991. - 317 с.
24. Аджемов, А.С. Размножение ошибок при передаче дискретного сообщения методом скользящего индекса с подтверждением / А.С. Аджемов // Электросвязь. - 1976. - № 2. - С. 9-18.
25. Аджемов, А.С. Выбор способа задания уточняющих символов при методе скользящего индекса с подтверждением / А.С. Аджемов // Электросвязь. - 1978. - № 10. - С. 15-18.

26. Adzhemov A.S., Sineva I.S. Efficiency of genetic-like coding algorithm for metric space sources / 2-nd IEEE International conference on circuit and systems for communication, Russia, Moscow, 06.2004. 4 p.
27. Аджемов, А.С. Теоретические границы и возможности их достижения в будущих инфокоммуникациях / А.С. Аджемов // Электросвязь. - 2013. - № 11. – С. 40-44.
28. Аджемов, А.С. Предельные значения эффективности асинхронного сопряжения дискретного сигнала с синхронным цифровым трактом / А.С. Аджемов // Электросвязь. - 1985. - № 3. – С. 21-28
29. Аджемов, А.С. Оценка эффективности асинхронной передачи факсимильного сигнала по синхронному цифровому тракту / А.С. Аджемов // Электросвязь. - 1986. - № 8. – С. 25-30.
30. Аджемов, А.С. Метод аналого-цифровых и цифро-аналоговых преобразований на основе сплайн-интерполяции / А.С. Аджемов, И.С. Синева // Электросвязь. - 1998. - № 2. – С. 30-33.
31. Adzhemov, A. S. Noise immunity enhancement by using a genetic-like coding algorithm for metric source without introduction of additional redundancy / A. S. Adzhemov, A. V. Pestryakov, I. S. Sineva and Y. S. Shinakov // International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT&QM&IS). - 2017. - PP. 158-161.
32. Adzhemov, A. S. On Some Features of Binary Code Combinations / A. S. Adzhemov, S. A. Adzhemov // Systems of Signals Generating and Processing in the Field of on Board Communications. – 2019. - PP. 1-7.
33. Adzhemov, A. S. Code Distance Table and its Application / A. S. Adzhemov // Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). – 2018. - PP. 1-5.
34. Колмогоров, А.Н. Основные понятия теории вероятностей / А.Н. Колмогоров; Москва: Наука, 1974. – 234 с.
35. Аджемов, А.С. Общая теория связи: учеб. пособие / А.С. Аджемов, В.Г. Санников; Москва: Горячая Линия-Телеком, 2017. - 624 с.

36. Пат. SU 1111257 A1 СССР. Асинхронное сопряжение цифровых сигналов [Текст] / Ю.В. Ларин, В.М. Суханов (СССР); заявитель и патентообладатель ЛЭИС им. М.А. Бонч-Бруевича; опубл. 29.06.1983.
- 37.Корнилов, И.И. Основы построения телекоммуникационных сетей и систем, Многоканальные телекоммуникационные системы, Цифровые системы передачи / И.И. Корнилов, Л.А. Марыкова; Самара, ИУНЛ ПГУТИ, 2013. - 90 с.
- 38.Сидоренко, И.А. Исследования влияния джиттера на процесс цифрового преобразования / И.А. Сидоренко, И.И. Лубков // Научные ведомости. - 2019. - Том 46. - № 3. - С. 573-577.
- 39.Ваггенер, Б. Методы импульсной кодовой модуляции / Б. Вагнер; Нью-Йорк: Ван Ностранд Рейнхольд. – 1995. - 146 с.
- 40.Ваггенер, Б. Проектирование систем кодовой импульсной модуляции / Б. Вагнер; Бостон, Массачусетс: Artech House. – 1999. – 211 с.
- 41.Величкин, А.И. Передача аналоговых сообщений по цифровым каналам связи / А.И. Величкин; М. Радио и связь. - 1983. - 240 с.
- 42.Pohlman, K.C. Principles of Digital Audio / K.C. Pohlman; Boston: McGraw-Hill. - 2005. - PP. 860.
- 43.Watkinson, J.R. The Art of Digital Audio / J.R. Watkinson; Boston, MA: Focal Press. - 1994. – PP. 650.
- 44.Васильев, Н. Метрические пространства / Н. Васильев // Квант. - 1990. - № 1. - С. 5-8.
- 45.Васильев, Н. Метрические пространства / Н. Васильев // Квант. - 1970. - № 10. - С. 13-20.
- 46.Френкель, А Основания теории множеств /А. Френкель, И. Бар-Хиллел; Москва.: Мир, 1966. - 556 с.
- 47.Конахович, Г. Ф. Защита информации в телекоммуникационных системах / Г. Ф. Конахович, В. П. Климчук, С. М. Паук; Москва: МК-Пресс, 2005. - 288 с.
- 48.Блох, Э.Л. Теория кодирования / Э.Л. Блох; Москва: Мир, 1964, 266 с.
- 49.Блох Э.Л. Исправление ошибок и стираний кодами Боуза–Чоудхури / Э.Л. Блох // Проблемы передачи информации. – 1965. - Том 1. - Выпуск 3. – С. 12-19.

50. Зигангиров К.Ш. О корректирующей способности кодов с малой плотностью проверок на четность / К.Ш. Зигангиров, А.Е. Пусане, Д.К. Зигангиров, Д.Дж. Костелло // Проблемы передачи информ. – 2008. – С.50–62
51. Зигангиров, Д.К. Декодирование низкоплотностных кодов с проверочными матрицами, составленными из перестановочных матриц, при передаче по каналу со стираниями / Д.К. Зигангиров, К.Ш. Зигангиров // Проблемы передачи информации. – 2006. - Том 42. - Выпуск 2. - С. 44–52
52. Золотарев, В.В. Помехоустойчивое кодирование: методы и алгоритмы / В.В. Золотарев, Г.В. Овечкин; Москва: Горячая линия-Телеком, 2004. - 127 с.
53. Золотарёв В.В. Теория кодирования как задача поиска глобального экстремума / В.В. Золотарев; Москва: Горячая линия-Телеком, 2018. - 222 с.
54. Фролов, А. А. Асимптотическая оценка доли ошибок, исправляемых q-ичными МПП-кодами / А.А. Фролов, В.В. Зяблов // Проблемы передачи информации. - 2010. - Т. 46. - № 2. - С. 47–65.
55. Окунев, Ю.Б. О кодировании сигналов с многократной фазоразностной манипуляцией / Ю.Б. Окунев // Электросвязь. - 1964. - № 2. - С. 8-14.
56. Окунев Ю.Б. Прием сигналов с фазоразностной манипуляцией методами моделирования / Ю.Б. Окунев // Труды научно-технической конференции ЛЭИС. -1965. - Вып. 2. - С. 3-10.
57. Окунев Ю.Б. Фазоразностная манипуляция высокого порядка / Ю.Б. Окунев // Труды учебных институтов связи. - 1966. - Вып 28. - С. 12-18.
58. Емельянов, Г.А. Передача дискретной информации и основы телеграфии / Г.А. Емельянов, В.О. Шварцман; Москва: Связь, 1973. - 367 с.
59. Марков А. А. Введение в теорию кодирования / А.А. Марков; Москва: Наука, 1982. - 192 с.
60. Derek, R. Understanding the Effect of Clock Jitter on High Speed ADCs / R. Derek, T. Eric, S. Alison // LINEAR technology. – 2006. - № 1013. - PP. 24-56.
61. Панфилов, И.П. Теория электрической связи / И.П. Панфилов, В.Е. Дырда; Москва: Радио и связь, 1991. - 344 с.

62. Дуленко, Ю.В. О помехоустойчивости относительной фазовой манипуляции / Ю.В. Дуленко // Электросвязь. - 1962. - № 11. - С. 61-62.
63. Jacobs, J. Comparison of M-ary modulation systems / J. Jacobs // Bell. Syst. Tech. J. - 1967. - Vol. 45. - N 5. - PP. 843-864.
64. Аджемов, А.С. Программная реализация для оценки эффективности алгоритма кодирования источника при преобразовании метрических пространств / А.С. Аджемов, А.Ю. Кудряшова // Труды НИИР: сб. ст.; Москва: НИИР. -2020. - № 4. - С. 7-23.
65. Аджемов, А.С. Особенности оценки качества передачи сигналов в различных метрических пространствах / А.С. Аджемов, А.Ю. Кудряшова // Фундаментальные проблемы радиоэлектронного приборостроения. - 2017. - Т. 17. - № 4. - С. 886-888.
66. Аджемов, А.С. Об особенностях оценки качества преобразования сигналов при последовательных преобразованиях в различных метрических пространствах / А.С. Аджемов, А.Ю. Кудряшова // Сборник трудов XII Международной научно-технической конференции «Технологии информационного общества». Москва, Московский технический университет связи и информатики (МТУСИ); Москва: ИД Медиа Паблишер. - 2018. - С. 211-213.
67. Аджемов А.С. Особенности оценки мощности множества вариантов выбора двоичных кодовых комбинаций при цифровизации сигнала / А.С. Аджемов, А.Ю. Кудряшова // Системы синхронизации, формирования и обработки сигналов. - 2018. - Т. 9. - № 1. - С. 5-8.
68. Аджемов, А.С. Особенности оценки мощности множества вариантов выбора двоичных кодовых комбинаций / А.С. Аджемов, А.Ю. Кудряшова // Фундаментальные проблемы радиоэлектронного приборостроения. - 2018. - Т. 18. - № 4. - С.926-929.
69. Кудряшова, А.Ю. Особенности оценки кодирования при различных конфигурациях пространства источника / А.Ю. Кудряшова // DSPA: Вопросы применения цифровой обработки сигналов. - 2018. - Т. 8. - № 3. - С. 228-232

70. Аджемов, А.С. Построение эффективного алгоритма оценки кодирования источника при преобразовании метрических пространств / А.С. Аджемов, А.Ю. Кудряшова, А.В. Долгопятова / Сборник трудов XIII Международной научно-технической конференции «Технологии информационного общества»; Москва: ИД Медиа Паблшер. - 2019. - С.195-198.
71. Кудряшова, А.Ю. A method of efficient coding of color images under the condition of permissible and forbidden values of color gamut / А.Ю. Кудряшова // Т-Comm: Телекоммуникации и транспорт. - 2019. - Т.13. - № 6. - С.65-70.
72. Аджемов, А.С. Модель эффективного кодирования цветного изображения с учетом особенностей систем колориметрии / А.С. Аджемов, А.Ю. Кудряшова // Инфокоммуникационные и радиоэлектронные технологии. - 2019. - Т.2. - № 3. - С. 349-360.
73. Аджемов, А.С. Использование модели зрительной системы человека для оптимизации параметров передачи изображения / А.С. Аджемов, А.Ю. Кудряшова // Инфокоммуникационные и радиоэлектронные технологии. - 2019. - Т.2. - № 4. - С. 489-502.
74. Кудряшова, А.Ю. Минимизация искажений сигнала при наиболее оптимальном выборе метода цифрового преобразования / А.Ю. Кудряшова // Т-Comm: Телекоммуникации и транспорт. - 2020. - Т.14. - № 5. - С.27-34.
75. Kudryashova, A. Y. Building an Algorithm for Estimating the Effective Coding of a Source when Converting Signals in Various Metric Spaces / A.Y. Kudryashova, A. S. Adzhemov // Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). – 2018. - PP. 1-4.
76. Adzhemov, A. S. About Interferable Binary Code Constructions / A. S. Adzhemov, A.Y. Kudryashova // Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO). - 2020. - PP. 1-4.
77. Adzhemov, A. S. Model of Effective Color Image Coding Taking into Account the Peculiarities of Colorimetry System / A. S. Adzhemov, A.Y. Kudryashova // Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO). - 2019. - PP. 1-5.

78. Kudryashova, A. Y. Application of Weber-Fechner Law in Image Transmission in the Field of Onboard Communications / A.Y. Kudryashova, A. S. Adzhemov, Vlasuyk I. V. // Systems of Signals Generating and Processing in the Field of on Board Communications. - 2019. - PP. 1-6.
79. Adzhemov, A. S. Features rate estimation options binary codewords with the digitalization of the signal / A. S. Adzhemov, A.Y. Kudryashova // Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO). – 2018. - PP. 1-5.
80. Adzhemov, A. S. Evaluation Program of an Efficient Source Coding Algorithm Under the Condition of Converting Metric Spaces / A. S. Adzhemov, A.Y. Kudryashova // Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). – 2019. - PP. 1-5.
81. Adzhemov, A. S. About features of evaluation of the quality of generation and signal processing at stage transformations in wiring and optical communication systems / A. S. Adzhemov, A.Y. Kudryashova // Systems of Signals Generating and Processing in the Field of on Board Communications. – 2018. - PP. 1-4.
82. Adzhemov, A. S. Reducing Distortion when Applying the Rational Choice of the Digital Conversion Method / A. S. Adzhemov, A.Y. Kudryashova // Systems of Signals Generating and Processing in the Field of on Board Communications. – 2020. - PP. 1-6.
83. Adzhemov, A. S. Features of Converting Signals to Binary and Minimizing Distortion / A. S. Adzhemov, A.Y. Kudryashova // Systems of Signals Generating and Processing in the Field of on Board Communications. – 2021. - PP. 1-5.
84. Adzhemov, A. S. Distortion with Multiple Discrete Modulation Techniques / A. S. Adzhemov, A.Y. Kudryashova // Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). – 2021. - PP. 1-5, doi: 10.1109/WECONF51603.2021.9470537.
85. Adzhemov, A. S. Features of Bijection of Spaces when Transmitting Messages from Source to Recipient/ A. S. Adzhemov, A.Y. Kudryashova // Systems of Signal

- Synchronization, Generating and Processing in Telecommunications. – 2021. – PP. 1-4.
86. Кудряшова А.Ю. Analysis of the peculiarities of encoding messages from various sources / А. Ю. Кудряшова // Т-Comm: Телекоммуникации и транспорт. - 2021. - Т. 15. - № 9. - С.56-62.
87. Кудряшова А.Ю. Development of a program for calculating additional distortions for various models of errors / А. Ю. Кудряшова // Т-Comm: Телекоммуникации и транспорт. - 2022. - Т. 16. - № 1. - С.51-58.
88. Papoulis, A. Probability, Random Variables, and Stochastic Processes / A. Papoulis; New York: McGraw-Hill Book Company, 1965. – 467 p.
89. Hard, F. J. Windows, Harmonic Analysis, and the Discrete Fourier Transform / J.F. Hard // Proc. IEEE. - Vol. 67, January. - 1979. - PP. 58-63.
90. Martin, G. Gyroscopes May Cease Spinning / G. Martin // IEEE Spectrum. - Vol. 23. - N. 2, February. -1986. - PP. 48-53.
91. Vanderkooy, J. Resolution beyond the Least Significant Bit with Dither / J. Vanderkooy, S. T. Lipshitz // J. Audio Eng. Soc. - N. 3, March. -1984. - PP. 106-112.
92. Blesser, B. A. Digitization of Audio: A Comprehensive Examination of Theory, Implementation, and Current Practice / B. A. Blesser // J. Audio Eng. Soc. - Vol. 26. - N. 10, October. – 1978. - PP.739-771.
93. Sluyter, R. J. Digitization of Speech / R. J. Sluyter // Phillips Tech. Rev. - Vol. 41. - N. 7-8. - 1983-84. - PP. 201-221.
94. Jayant, N. S. Digital Coding of Waveforms / N. S. Jayant, P. Noil; New York: Prentice-Hall, Inc., Englewood Cliffs. - 1984. 415 p.
95. Marcel, J. D. Linear Prediction of Speech / J. D. Marcel, A. H. Jr. Gray; New York: Springer-Verlag, 1976. 347 p.
96. Deller, J. Discrete-Time Processing of Speech Signals / J. Deller, J. Proakis, J. Hansen; New York: Macmillan, 1993. 460 p.
97. Candy J. and Temes G. *Oversampling Delta-Sigma Data Converters*. IEEE Press, 1991. 543 p.

98. Dick, C. FPGA Signal Processing Sigma-Delta Modulation / C. Dick, F. Harris // IEEE Signal Proc. Mag. - Vol. 17. - N. 1, January. - 2000. - PP. 20-35.
99. Cummysky, P. Adaptive Quantization in Differential PCM Coding of Speech / P. Cummysky, N. Jayant, J. Flanagan // Bell Syst. Tec J. - Vol. 52. – 1973. - PP. 115-119.
100. Gersho, A. Asymptotically Optimal Block Quantization / A. Gersho // IEEE Trans. Inf. Theory. - Vol. IT25. - N. 4, July. – 1979. - PP. 373-380.
101. Gersho, A. On the Structure of Vector Quantizers / A. Gersho // IEEE Trans. Inf. Theory. - Vol. IT28. - N. 2, March. – 1982. - PP. 157-166.
102. Lynch, T. J. Data Compression Techniques and Applications / T. J. Lynch; New York: Lifetime Learning Publications, 1985. 516 p.
103. Schafer, R. W. Design of Digital Filter Banks for Speech Analysis / R. W. Schafer, L. R. Rabiner // Bell Syst. Tech. J. - Vol. 50. - N. 10, December. – 1971. - PP. 3097-3115.
104. Hunter, R. International Digital Facsimile Coding Standard / R. Hunter, A. Robinson // Proc. IEEE. - Vol. 68. - N. 7, July. -1980. - PP. 854-867.
105. Cox, R. Three New Speech Coders From the ITU Cover a Range of Applications / R. Cox // IEEE Comm. Mag. - Vol. 35. - N. 9, September. – 1997. - PP. 40-47.
106. Noll, P. Wideband Speech and Audio Coding / P. Noll // IEEE Comm. Mag. - Vol. 31. - N. 11, November. -1993. - PP. 34-44.
107. Solari S. Digital Video and Audio Compression / S. Solari; New York: McGraw-Hill, 1997. 360 p.
108. Rzeszewski, T. Digital Video: Concepts and Applications Across Industries / T. Rzeszewski; IEEE Press, 1995. 280 p.
109. Леоненков, А.В. Решение задач оптимизации в среде MS Excel / А.В. Леоненков; Москва: БХВ-Петербург, 2005. - 704 с.
110. Кашаев, С.П. Программирование в Microsoft Excel на примерах / С.П. Кашаев; Москва: БХВ-Петербург, 2007. - 320 с.
111. Гайдышев, И. Решение научных и инженерных задач средствами Excel, VBA и C/C++ / И. Гайдышев; Москва: БХВ-Петербург, 2004. - 512 с.

112. Петрова, В.А. Программирование и решение сложных задач в Excel / В.А. Петрова; Екатеринбург: Издательство Уральского университета, 2016. - 88 с.
113. Михеев, Р. VBA и программирование в MS Office / Р. Михеев; Санкт-Петербург: БХВ-Петербург, 2006. - 369 с.
114. Гарнаев, А.Ю., Microsoft Office Excel: разработка приложений / А.Ю. Гарнаев, Л.В. Рудикова; Санкт-Петербург: БХВ-Петербург, 2011. 528с.
115. Ларсен, Р. Инженерные расчеты в Excel / Р. Ларсен; Москва: Изд-во «Вильямс», 2004. – 544 с.

Приложение А. Исходный код.**Файл SciExcel\ThisAddIn.cs - Встраиваемый модуль (add-in) для Microsoft Excel. Стартовый класс приложения**

```
using
System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml.Linq;
using Excel = Microsoft.Office.Interop.Excel;

namespace SciExcel
{
    /// <summary>
    /// Модуль (add-in)
    /// </summary>
    public partial class ThisAddIn
    {
        /// <summary>
        /// Приложение Microsoft Excel
        /// </summary>
        public static Excel.Application App;

        /// <summary>
        /// Загрузка модуля
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void ThisAddIn_Startup(object sender, System.EventArgs e)
        {
            // Сохранение ссылки на приложение
            App = Application;
        }

        /// <summary>
        /// Выгрузка модуля
        /// </summary>
        /// <param name="sender"></param>

```

```

    /// <param name="e"></param>
    private void ThisAddIn_Shutdown(object sender, System.EventArgs
e)
    {
    }

    /// <summary>
    /// Текущий рабочий лист Microsoft Excel
    /// </summary>
    /// <returns></returns>
    public static Excel.Worksheet ActiveSheet
    {
        get
        {
            return App.ActiveSheet;
        }
    }

    #region Код, автоматически созданный VSTO

    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
    private void InternalStartup()
    {
        this.Startup += new System.EventHandler(ThisAddIn_Startup);
        this.Shutdown += new
System.EventHandler(ThisAddIn_Shutdown);
    }

    #endregion
    }
}

```

**Файл SciExcel\MyRibbon.cs - Лента (ribbon) Microsoft Excel. Вызов
функции расчета**

```
using
System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Tools.Ribbon;

namespace SciExcel
{
    /// <summary>
    /// Лента
    /// </summary>
    public partial class MyRibbon
    {
        /// <summary>
        /// Загрузка ленты
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void MyRibbon_Load(object sender, RibbonUIEventArgs e)
        {
        }

        /// <summary>
        /// Выполнение расчета
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void button_Click(object sender, RibbonControlEventArgs e)
        {
            try
            {
                var form = new ProcessForm();
                form.ShowDialog();
            }
            catch (Exception ex)
            {
            }
        }
    }
}
```

```

        System.Windows.Forms.MessageBox.Show(ex.Message,
        "Найка", System.Windows.Forms.MessageBoxButtons.OK,
        System.Windows.Forms.MessageBoxIcon.Exclamation);
    }
}
}
}

```

Файл SciExcel\ProcessForm.cs - Главная форма вычисления.

Основная математическая логика.

```

using System;
    using System.Collections.Generic;
    using System.ComponentModel;
    using System.Data;
    using System.Diagnostics;
    using System.Drawing;
    using System.Linq;
    using System.Text;
    using System.Threading.Tasks;
    using System.Windows.Forms;
    using SciExcel.Mathematics;

namespace SciExcel
{
    public partial class ProcessForm : Form
    {
        /// <summary>
        /// Заголовок исходной матрицы вероятностей
        /// </summary>
        private const string probabilityName1 = "Матрица вероятностей";

        /// <summary>
        /// Заголовок нормированной матрицы вероятностей
        /// </summary>
        private const string probabilityName2 = "Нормированная матрица
        вероятностей";

        /// <summary>

```

```

/// Заголовок матрицы искажений
/// </summary>
private const string riskName = "Матрица искажений";

/// <summary>
/// Заголовок вектора вероятностей появления в канале Pк
/// </summary>
private const string riskProbName = "Вектор вероятностей
появления в канале Pк";

/// <summary>
/// Признак завершения расчета
/// </summary>
private bool done = false;

/// <summary>
/// Время выполнения расчета
/// </summary>
private Stopwatch watch = new Stopwatch();

public ProcessForm()
{
    InitializeComponent();
}

/// <summary>
/// Выполнение вычислений в отдельном потоке
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void worker_DoWork(object sender, DoWorkEventArgs e)
{
    bool EnableAnimations = false;

    try
    {
        // Оптимизация визуализации
        EnableAnimations = ThisAddIn.App.EnableAnimations;
        ThisAddIn.App.EnableAnimations = false;
    }
}

```

```

ThisAddIn.App.Calculation =
Microsoft.Office.Interop.Excel.XlCalculation.xlCalculationManual;
// ThisAddIn.App.Visible = false;

// Поиск исходных матриц на листе Excel
// Исходная матрица вероятностей
ExcelMatrix probability1 =
ExcelMatrix.ReadMatrix(probabilityName1, true, true);
// Размерность матрицы
int N = probability1.NoRows;

// Матрица искажений
ExcelMatrix risk1 = ExcelMatrix.ReadMatrix(riskName);

// Вектор сумм строк матрицы
ExcelMatrix mas_sum_strok1 = probability1.SumByRow();
mas_sum_strok1.Populate();

// Создаем нормированную матрицу на основе существующей
ExcelMatrix probability2 = new ExcelMatrix(probability1);
probability2.SetPosition(probabilityName2);

// Нормирование матрицы
for (int row = 0; row < probability1.NoRows; row++)
{
    if (mas_sum_strok1[row, 0] < 1)
        for (int col = 0; col < probability1.NoCols; col++)
        {
            // Нормирование ячейки матрицы
            probability2[row, col] = probability1[row, col] /
mas_sum_strok1[row, 0];
        }
}
probability2.Populate();

// Вектор сумм строк нормированной матрицы
ExcelMatrix mas_sum_strok2 = probability2.SumByRow();
mas_sum_strok2.Populate();

```

```

// Проверка матрицы искажений
if (!risk1.Diagonal0())
{
    throw new Exception("Элементы главной диагонали
матрицы искажений ненулевые");
}

// Вектор вероятностей появления в канале Pk
// Используется равномерное распределение вероятности 1/N,
где N - размерность матриц
ExcelMatrix vekt_prob1 = new
ExcelMatrix(mas_sum_strok2.NoRows, 1);
vekt_prob1.SetPosition(riskProbName);
for (int i = 0; i < mas_sum_strok2.NoRows; i++)
{
    vekt_prob1[i, 0] = mas_sum_strok2[i, 0] / N;
}
vekt_prob1.Populate();

// Построение всех комбинаций
Permutator permutator = new Permutator(N);

double[] dismax = new double[permutator.N];
double[] krismax = new double[permutator.N];

// Возводим элементы матрицы искажений в квадрат
// (инвариант цикла)
ExcelMatrix risk2 = risk1.Square();

// Место размещения статистики расчета
Microsoft.Office.Interop.Excel.Range result =
ExcelMatrix.Find("Статистика");
int rowr = result.Row + 2;
int colr = result.Column;

// Построение всех перестановок матрицы - подготовка
int rowBase = probability2.Row0;
int percent = 0;

```

```

//Parallel.ForEach(permutator, (permutation) =>
foreach (Permutation permutation in permutator)
{
    // Досрочное завершение процесса
    if (worker.CancellationPending)
    {
        return;
    }

    // Базовая строка
    int rown = rowBase + (N + 2 +
(probability1.HasColumnHeaders ? 1 : 0)) * permutation.N - 1;

    // Отображение хода выполнения процесса
    int newpercent = (int)(permutation.N * progress.Maximum /
permutator.N);
    if (newpercent > percent)
    {
        percent = newpercent;
        worker.ReportProgress(percent);
    }

    // Номер и название перестановки
    ExcelMatrix.Cell(rown, probability2.Col0 - 1).Value =
permutation.N;
    ExcelMatrix.Cell(rown, probability2.Col0).Value =
$"Перестановка #{permutation.N}: {permutation}";

    // Формирование очередной перестановки
    ExcelMatrix newArray = new ExcelMatrix(probability2,
permutation);
    newArray.Row0 = rown + 1 +
(probability2.HasColumnHeaders ? 1 : 0);
    newArray.Col0 = probability2.Col0;

    // Копирование формата (долгая синхронная операция)
    /*
probability1.GetRange().Copy();

```

```

newArray.GetRange().PasteSpecial(Microsoft.Office.Interop.Excel.XlPaste
Type.xlPasteFormats);
    */

    // Публикация результата - строго после копирования
формата !
newArray.Populate();

    // Находим выражение по формуле (вектор R) * Pk
ExcelMatrix R = risk1.Multiplication(newArray, vekt_prob1);

    // Находим коэффициент искажений
double koef_r = R.Summa();

    // Находим выражение по формуле (вектор D1)
ExcelMatrix D1 = risk1.Multiplication(newArray, vekt_prob1);

    // Находим выражение по формуле (вектор D2)
ExcelMatrix D2 = risk2.Multiplication(newArray, vekt_prob1);

    // Находим квадрат величины, умножаем на вероятность
double DIS1 = D1.Summa();
double DIS2 = D2.Summa();

    // Находим значение дисперсии
double DIS = DIS2 - DIS1 * DIS1;

    // Публикуем результаты
ExcelMatrix.Cell(rown + 1, probability2.Col0 + N).Value =
"Коэффициент искажений";
ExcelMatrix.Cell(rown + 2, probability2.Col0 + N).Value =
"Матожидание СВ";
ExcelMatrix.Cell(rown + 3, probability2.Col0 + N).Value =
"Матожидание квадрата СВ";
ExcelMatrix.Cell(rown + 4, probability2.Col0 + N).Value =
"Дисперсия";

```

```

// Рядом с матрицей
ExcelMatrix.Cell(rowN + 1, probability2.Col0 + N + 1).Value =
koef_r;
ExcelMatrix.Cell(rowN + 2, probability2.Col0 + N + 1).Value =
DIS1;
ExcelMatrix.Cell(rowN + 3, probability2.Col0 + N + 1).Value =
DIS2;
ExcelMatrix.Cell(rowN + 4, probability2.Col0 + N + 1).Value =
DIS;

// Статистика
ExcelMatrix.Cell(rowr, colr + 0).Value = permutation.N;
ExcelMatrix.Cell(rowr, colr + 1).Value =
permutation.ToString();
ExcelMatrix.Cell(rowr, colr + 2).Value = koef_r;
ExcelMatrix.Cell(rowr, colr + 3).Value = DIS1;
ExcelMatrix.Cell(rowr, colr + 4).Value = DIS2;
ExcelMatrix.Cell(rowr, colr + 5).Value = DIS;
rowr++;

// Сохранение коэффициента и дисперсии
dismax[permutation.N - 1] = DIS;
krismax[permutation.N - 1] = koef_r;
}
//);

int rowN = (int)(rowBase + (N + 2 +
(probability1.HasColumnHeaders ? 1 : 0)) * (permutator.N + 1) - 1);

// Находим минимальное и максимальное значение дисперсии
и коэффициента искажений
Pair kmax = new Pair(krismax, PairType.Max);
Pair kmin = new Pair(krismax, PairType.Min);
Pair dmax = new Pair(dismax, PairType.Max);
Pair dmin = new Pair(dismax, PairType.Min);

ExcelMatrix.Cell(rowN + 0, probability2.Col0).Value =
"Максимальное значение коэффициента искажений";

```

```
ExcelMatrix.Cell(rowN + 1, probability2.Col0).Value = "Вариант кодирования";
```

```
ExcelMatrix.Cell(rowN + 2, probability2.Col0).Value = "Минимальное значение коэффициента искажений";
```

```
ExcelMatrix.Cell(rowN + 3, probability2.Col0).Value = "Вариант кодирования";
```

```
ExcelMatrix.Cell(rowN + 4, probability2.Col0).Value = "Максимальное значение дисперсии";
```

```
ExcelMatrix.Cell(rowN + 5, probability2.Col0).Value = "Вариант кодирования";
```

```
ExcelMatrix.Cell(rowN + 6, probability2.Col0).Value = "Минимальное значение дисперсии";
```

```
ExcelMatrix.Cell(rowN + 7, probability2.Col0).Value = "Вариант кодирования";
```

```
ExcelMatrix.Cell(rowN + 0, probability2.Col0 + 1).Value = kmax.Value;
```

```
ExcelMatrix.Cell(rowN + 1, probability2.Col0 + 1).Value = kmax.I + 1;
```

```
ExcelMatrix.Cell(rowN + 2, probability2.Col0 + 1).Value = kmin.Value;
```

```
ExcelMatrix.Cell(rowN + 3, probability2.Col0 + 1).Value = kmin.I + 1;
```

```
ExcelMatrix.Cell(rowN + 4, probability2.Col0 + 1).Value = dmax.Value;
```

```
ExcelMatrix.Cell(rowN + 5, probability2.Col0 + 1).Value = dmax.I + 1;
```

```
ExcelMatrix.Cell(rowN + 6, probability2.Col0 + 1).Value = dmin.Value;
```

```
ExcelMatrix.Cell(rowN + 7, probability2.Col0 + 1).Value = dmin.I + 1;
```

```
}
```

```
finally
```

```
{
```

```
ThisAddIn.App.Calculation = Microsoft.Office.Interop.Excel.XlCalculation.xlCalculationAutomatic;
```

```
ThisAddIn.App.EnableAnimations = EnableAnimations;
```

```
ThisAddIn.App.Visible = true;
```

```
}
```

```
}
```

```

/// <summary>
/// Загрузка формы
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void ProcessForm_Load(object sender, EventArgs e)
{
    watch.Start();
    worker.RunWorkerAsync();
}

/// <summary>
/// Отображение хода выполнения процесса
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void worker_ProgressChanged(object sender,
ProgressChangedEventArgs e)
{
    // Расчет оставшего времени и прогноз времени завершения
расчета
    progress.Value = e.ProgressPercentage;
    TimeSpan elapsed = watch.Elapsed;
    if (progress.Value > 0)
    {
        long s = (long)(elapsed.TotalSeconds / progress.Value *
(progress.Maximum - progress.Value));
        long m = s / 60; s = s % 60;
        long h = m / 60; m = m % 60;
        long d = h / 24; h = h % 24;
        TimeSpan ts = new TimeSpan((int)d, (int)h, (int)m, (int)s);
        DateTime eta = DateTime.Now + ts;
        labelText.Text = $"Затрачено: {elapsed}, осталось: {ts},
завершение: {eta}";
    }
    else
    {
        labelText.Text = $"Затрачено: {elapsed}";
    }
    labelNumber.Text = $"Рассчитано вариантов:
{e.ProgressPercentage * 0.1}%";
}

```

```

/// <summary>
/// Завершение процесса вычислений
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void worker_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)
{
    watch.Stop();
    button.Text = "Обработка завершена";
    button.Enabled = true;
    done = true;
}

/// <summary>
/// Досрочное завершение процесса или закрытие формы в конце
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button_Click(object sender, EventArgs e)
{
    if (done)
    {
        Close();
    }
    else
    {
        button.Text = "Отмена запрошена";
        button.Enabled = false;
        worker.CancelAsync();
    }
}
}
}
}

```

**Файл SciExcel\ExcelMatrix.cs - Объект "матрица", отображённая на
лист Microsoft Excel**

```

using
System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms.VisualStyles;
using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Tools;

namespace SciExcel
{
    public class ExcelMatrix : Mathematics.SciMatrix
    {
        /// <summary>
        /// Столбец таблицы Excel, в котором начинаются значения
        матрицы
        /// </summary>
        public int Col0;

        /// <summary>
        /// Строка таблицы Excel, в которой начинаются значения
        матрицы
        /// </summary>
        public int Row0;

        #region "Конструкторы"

        /// <summary>
        /// Квадратная матрица размерностью N
        /// </summary>
        /// <param name="N">Размерность квадратной матрицы</param>
        public ExcelMatrix(int N) : base(N) { }

        /// <summary>
        /// Матрица размерностью, равной исходной
        /// </summary>
        /// <param name="source">Исходная матрица</param>
        public ExcelMatrix(ExcelMatrix source) : base(source) { }
    }
}

```

```

/// <summary>
/// Матрица размерностью rows * cols
/// </summary>
/// <param name="rows">Количество строк</param>
/// <param name="cols">Количество столбцов</param>
public ExcelMatrix(int rows, int cols) : base(rows, cols) { }

/// <summary>
/// Построение матрицы путем перестановки исходной матрицы
/// </summary>
/// <param name="source">Исходная матрица</param>
/// <param name="permutation">Перестановка</param>
public ExcelMatrix(ExcelMatrix source, Mathematics.Permutation
permutation) : base(source, permutation) { }

#endregion

#region "Microsoft Excel"

/// <summary>
/// Точный поиск строки на текущем листе Excel
/// </summary>
/// <param name="header">Искомая строка</param>
/// <returns></returns>
public static Range Find(string header)
{
    // Ищем точное совпадение
    Range h = ThisAddIn.ActiveSheet.Cells.Find(What: header,
LookAt: XlLookAt.xlWhole);

    if (h == null)
    {
        throw new Exception($"На текущем листе не найдена строка
'{{header}}");
    }

    return h;
}

```

```

/// <summary>
/// Заданная ячейка таблицы Excel
/// </summary>
/// <param name="row">Номер строки, начиная с 1</param>
/// <param name="col">Номер столбца, начиная с 1</param>
/// <returns></returns>
public static Range Cell(int row, int col)
{
    // Текущий документ
    Worksheet sheet = ThisAddIn.ActiveSheet;
    // Запрашиваемая ячейка
    Range cell = sheet.Cells[row, col];
    // Возврат результата
    return cell;
}

/// <summary>
/// Значение ячейки текущей таблицы Excel
/// </summary>
/// <param name="row">Номер строки, начиная с 1</param>
/// <param name="col">Номер столбца, начиная с 1</param>
/// <param name="type">Ожидаемый тип данных</param>
/// <returns></returns>
private static object CellValue(int row, int col, CellType type =
CellType.None)
{
    // Запрашиваемая ячейка
    Range cell = Cell(row, col);

    // Анализ ожидаемого типа данных
    switch (type)
    {
        case CellType.String:
            return cell.Value != null ? cell.Value : string.Empty;

        case CellType.Number:
            // Сброс цвета ячейки
            cell.Interior.Color = System.Drawing.Color.Transparent;
    }
}

```

```

    if (cell.Value is double) // вещественное число
    {
        return cell.Value;
    }
    if (cell.Value is int) // целое число
    {
        return cell.Value;
    }

    // Выделить некорректную строку
    cell.Interior.Color = System.Drawing.Color.OrangeRed;
    return double.NaN;
}
return cell.Value;
}

/// <summary>
/// Поиск заголовка
/// </summary>
/// <param name="header">Заголовок матрицы</param>
/// <param name="rowHeaders">Матрица включает названия
строк</param>
/// <param name="columnHeaders">матрица включает названия
столбцов</param>
/// <returns></returns>
public static ExcelMatrix ReadMatrix(string header, bool rowHeaders
= false, bool columnHeaders = false)
{
    // Поиск строки заголовка матрицы - ищем точное совпадение
    Range h = Find(header);

    // Строка и столбец ячейки заголовка
    int row = h.Row;
    int col = h.Column;

    // найдена ячейка, в которой начинается матрица - [row0, col]
    int row0 = ++row;

    // Матрица имеет названия столбцов (на это уходит ещё одна
строка) - учтём это при поиске

```

```

if (columnHeaders)
{
    row0++;
}

```

```

// Поиск окончания матрицы по строкам (вниз)
while (CellValue(row, col) != null)
{
    row++;
}
// Теперь row - первая пустая строка

```

```

// Размерность матрицы
int n = row - row0;

```

Excel // Создание матрицы с привязкой к расположению в листе

```

ExcelMatrix m = new ExcelMatrix(n)
{
    Col0 = col,
    Row0 = row0
};

```

```

for (int x = 0; x < n; x++) // столбцы
{
    for (int y = 0; y < n; y++) // строки
    {
        object v = CellValue(row0 + y, col + x, CellType.Number);
        m[y, x] = (double)v;
    }
}

```

```

// Матрица имеет заголовки строк - сохраним их в массиве
if (rowHeaders)
{
    m.RowHeader = new string[n];
    for (int y = 0; y < n; y++) // строки
    {
        object v = CellValue(row0 + y, col - 1, CellType.String);
        m.RowHeader[y] = (string)v;
    }
}

```

```

    }
}

// Матрица имеет заголовки столбцов
if (columnHeaders)
{
    m.ColHeader = new string[n];
    for (int x = 0; x < n; x++) // столбцы
    {
        object v = CellValue(row0 - 1, col + x, CellType.String);
        m.ColHeader[x] = (string)v;
    }
}

return m;
}

```

```

/// <summary>
/// Выгрузка данных на лист Excel
/// </summary>
public void Populate()
{
    // Заголовки строк
    if (RowHeader != null)
    {
        for (int row = 0; row < NoRows; row++)
        {
            Range cell = Cell(Row0 + row, Col0 - 1);
            // Текстовый формат
            cell.Interior.Color = 49407;
            cell.HorizontalAlignment = XlHAlign.xlHAlignCenter;
            cell.NumberFormat = "@";
            // Текст заголовка
            cell.Value = RowHeader[row];
        }
    }
    // Заголовки столбцов
    if (ColHeader != null)
    {
        for (int col = 0; col < NoCols; col++)
        {
            Range cell = Cell(Row0 - 1, Col0 + col);

```

```

        // Текстовый формат
        cell.Interior.Color = 49407;
        cell.HorizontalAlignment = XlHAlign.xlHAlignCenter;
        cell.NumberFormat = "@";
        // Текст заголовка
        cell.Value = ColHeader[col];
    }
}
// Собственно матрица
for (int row = 0; row < NoRows; row++)
{
    for (int col = 0; col < NoCols; col++)
    {
        // Выгрузка данных на лист Excel
        Range cell = Cell(Row0 + row, Col0 + col);
        cell.Value = this[row, col];
    }
}
}

/// <summary>
/// Установить место размещения матрицы на листе Excel
/// </summary>
/// <param name="header">Заголовок таблицы</param>
public void SetPosition(string header)
{
    Range h = Find(header);
    Row0 = h.Row + 1 + (HasColumnHeaders ? 1 : 0);
    Col0 = h.Column;
}

public Range GetRange()
{
    Range cell1 = Cell(Row0 + (HasRowHeaders ? -1 : 0), Col0 +
(HasColumnHeaders ? -1 : 0));
    Range cell2 = Cell(Row0 + NoRows - 1, Col0 + NoCols - 1);

    // Текущий документ
    Worksheet sheet = ThisAddIn.ActiveSheet;

    return sheet.Range[cell1, cell2];
}

```

}

#endregion

```

/// <summary>
/// Суммирование матрицы по строкам
/// </summary>
/// <returns></returns>
public ExcelMatrix SumByRow()
{
    // Вектор сумм строк матрицы
    ExcelMatrix vector = new ExcelMatrix(NoRows, 1)
    {
        // Вектор сумм располагается справа от исходной матрицы
        Row0 = this.Row0,
        Col0 = this.Col0 + this.NoCols
    };

    // Считаем сумму в строке
    for (int row = 0; row < NoRows; row++)
    {
        double sum = 0;
        for (int col = 0; col < NoCols; col++)
        {
            sum += this[row, col];
        }
        vector[row, 0] = sum;
    }

    return vector;
}

/// <summary>
/// Квадрат элементов матрицы
/// </summary>
/// <returns></returns>

```

```

public ExcelMatrix Square()
{
    ExcelMatrix square = new ExcelMatrix(this);
    for (int row = 0; row < NoRows; row++)
    {
        for (int col = 0; col < NoCols; col++)
        {
            square[row, col] = this[row, col] * this[row, col];
        }
    }
    return square;
}

```

```

/// <summary>

```

```

/// Умножение матрицы на другую матрицу и суммирование
построчно

```

```

/// </summary>

```

```

/// <param name="m">Матрица</param>

```

```

/// <returns></returns>

```

```

public ExcelMatrix Multiplication(ExcelMatrix m)

```

```

{

```

```

    // Контроль корректности операции

```

```

    if ((NoCols != m.NoCols) || (NoRows != m.NoRows))

```

```

    {

```

```

        throw new Exception($"Размеры матриц не совпадают: {Size}
и {m.Size}");

```

```

    }

```

```

    ExcelMatrix result = new ExcelMatrix(NoRows, 1);

```

```

    for (int row = 0; row < NoRows; row++)

```

```

    {

```

```

        double x = 0;

```

```

        for (int col = 0; col < NoCols; col++)

```

```

        {

```

```

            x += this[row, col] * m[row, col];

```

```

        }

```

```

        result[row, 0] = x;

```

```

    }

```

```

    return result;

```

```

}

```

```

/// <summary>

```

```

    /// Умножение матрицы на другую матрицу и вектор
    ПОЭЛЕМЕНТНО
    /// </summary>
    /// <param name="m"></param>
    /// <param name="vector"></param>
    /// <returns></returns>
    public ExcelMatrix Multiplication(ExcelMatrix m, ExcelMatrix
vector)
    {
        // Контроль корректности операции
        if ((NoCols != m.NoCols) || (NoRows != m.NoRows))
        {
            throw new Exception($"Размеры матриц не совпадают: {Size}
и {m.Size}");
        }
        if ((vector.NoCols != 1) || (NoRows != vector.NoRows))
        {
            throw new Exception($"Размеры матрицы и вектора не
совпадают: {Size} и {vector.Size}");
        }

        ExcelMatrix result = new ExcelMatrix(NoRows, 1);
        for (int row = 0; row < NoRows; row++)
        {
            double x = 0;
            for (int col = 0; col < NoCols; col++)
            {
                x += this[row, col] * m[row, col] * vector[row, 0];
            }
            result[row, 0] = x;
        }
        return result;
    }
}
}
}

```

Файл SciExcel\CellType.cs - Перечислимый тип. Тип ячейки Microsoft

Excel

```

using
System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SciExcel
{
    /// <summary>
    /// Тип ячейки таблицы Excel
    /// </summary>
    public enum CellType
    {
        /// <summary>
        /// Тип не задан
        /// </summary>
        None = 0,
        /// <summary>
        /// Строка
        /// </summary>
        String,
        /// <summary>
        /// Число
        /// </summary>
        Number
    }
}

```

Файл SciLibrary\Mathematics\Function.cs - Функция для вычисления факториала

```

using
System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SciExcel.Mathematics

```

```

{
    /// <summary>
    /// Сборник математических функций
    /// </summary>
    public static class Function
    {
        /// <summary>
        /// Факториал натурального числа
        /// </summary>
        /// <param name="n">Натуральное число</param>
        /// <returns></returns>
        public static long Factorial(int n)
        {
            long res = 1;
            for (int i = 2; i <= n; i++)
            {
                res *= i;
            }
            return res;
        }
    }
}

```

**Файл SciLibrary\Mathematics\SciMatrix.cs - Матрица с заголовками
строк и столбцов**

```

using
System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SciExcel.Mathematics
{
    /// <summary>
    /// Матрица с прикладными расчетами
    /// </summary>
    public class SciMatrix : Matrix
    {
        /// <summary>

```

```

/// Заголовки строк
/// </summary>
protected string[] RowHeader;

/// <summary>
/// Заголовки столбцов
/// </summary>
protected string[] ColHeader;

#region "Конструкторы"

/// <summary>
/// Квадратная матрица размерностью N
/// </summary>
/// <param name="N">Размерность квадратной матрицы</param>
public SciMatrix(int N) : base(N, N)
{
}

/// <summary>
/// Матрица размерностью, равной исходной
/// </summary>
/// <param name="source">Исходная матрица</param>
public SciMatrix(SciMatrix source) : base(source.NoRows,
source.NoCols)
{
    // Проверка на наличие заголовков столбцов и копирование их
при наличии
    if (source.ColHeader != null)
    {
        ColHeader = new string[source.ColHeader.Length];
        Array.Copy(source.ColHeader, ColHeader,
source.ColHeader.Length);
    }
    // Проверка на наличие заголовков строк и копирование их при
наличии
    if (source.RowHeader != null)
    {
        RowHeader = new string[source.RowHeader.Length];

```

```

        Array.Copy(source.RowHeader, RowHeader,
source.RowHeader.Length);
    }
}

```

```

/// <summary>
/// Матрица размерностью rows * cols
/// </summary>
/// <param name="rows">Количество строк</param>
/// <param name="cols">Количество столбцов</param>
public SciMatrix(int rows, int cols) : base(rows, cols)
{
}

```

```

/// <summary>
/// Построение матрицы путем перестановки исходной матрицы
/// </summary>
/// <param name="source">Исходная матрица</param>
/// <param name="permutation">Перестановка</param>
public SciMatrix(SciMatrix source, Permutation permutation) :
this(source)
{
    // Перестановка матрицы
    for (int row = 0; row < NoRows; row++)
    {
        int idx1 = permutation[row];
        for (int col = 0; col < NoCols; col++)
        {
            int idx2 = permutation[col];
            this[row, col] = source[idx1, idx2];
        }
    }
    // Перестановка заголовков строк
    if (RowHeader != null)
    {
        for (int row = 0; row < NoRows; row++)
        {
            int idx1 = permutation[row];
            RowHeader[row] = source.RowHeader[idx1];
        }
    }
    // Перестановка заголовков столбцов
    if (ColHeader != null)

```

```

    {
        for (int col = 0; col < NoCols; col++)
        {
            int idx2 = permutation[col];
            ColHeader[col] = source.ColHeader[idx2];
        }
    }
}

```

```
#endregion
```

```

/// <summary>
/// Есть ли у матрицы заголовки столбцов
/// </summary>
public bool HasColumnHeaders
{
    get
    {
        return ColHeader != null;
    }
}

```

```

/// <summary>
/// Есть ли у матрицы заголовки строк
/// </summary>
public bool HasRowHeaders
{
    get
    {
        return RowHeader != null;
    }
}

```

```

/// <summary>
/// Обозначение размерности матрицы
/// </summary>
public string Size
{
    get

```

```

    {
        return "${NoRows},{NoCols}";
    }
}

```

```

/// <summary>
/// Проверка главной диагонали на нулевые значения
/// </summary>
/// <returns>True - главная диагональ содержит только нулевые
значения

```

```

/// False - главная диагональ содержит как минимум одно
ненулевое значение</returns>

```

```

public bool Diagonal0()
{
    for (int i = 0; i < NoRows; i++)
    {
        if (this[i, i] != 0) { return false; }
    }
    return true;
}

```

```

/// <summary>
/// Сумма всех элементов матрицы
/// </summary>
/// <returns></returns>
public double Summa()
{
    double x = 0;
    for (int row = 0; row < NoRows; row++)
    {
        for (int col = 0; col < NoCols; col++)
        {
            x += this[row, col];
        }
    }
    return x;
}

```

```

/// <summary>
/// Копирование заголовков из другой матрицы
/// </summary>
/// <param name="m"></param>

```

```

public void CopyHeaders(SciMatrix m)
{
    // Контроль корректности операции
    if ((NoCols != m.NoCols) || (NoRows != m.NoRows))
    {
        throw new Exception($"Размеры матриц не совпадают: {Size}
и {m.Size}");
    }

    // Создание массивов
    RowHeader = new string[m.RowHeader.Length];
    ColHeader = new string[m.ColHeader.Length];

    // Копирование массивов
    Array.Copy(m.RowHeader, RowHeader, m.RowHeader.Length);
    Array.Copy(m.ColHeader, ColHeader, m.ColHeader.Length);
}
}
}

```

Файл SciLibrary\Mathematics\Pair.cs - Вспомогательный класс.

Значение и его индекс в последовательности

```

using
System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SciExcel.Mathematics
{
    /// <summary>
    /// Значение и его индекс в последовательности
    /// </summary>
    public class Pair
    {
        public readonly int I;
    }
}

```

```

public readonly double Value;

public Pair(double[] array, PairType type)
{
    I = 0;
    Value = array[0];
    for (int i = 1; i < array.Length; i++)
    {
        switch (type)
        {
            case PairType.Min:
                if (array[i] < Value)
                {
                    I = i;
                    Value = array[i];
                }
                break;

            case PairType.Max:
                if (array[i] > Value)
                {
                    I = i;
                    Value = array[i];
                }
                break;

            default:
                throw new ArgumentException($"Некорректное значение
параметра: {type}", "type");
        }
    }
}

```

Файл SciLibrary\Mathematics\PairType.cs - Перечислимый тип. Тип искомого значения (целевой функции)

```

using
System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SciExcel.Mathematics
{
    /// <summary>
    /// Тип искомого значения (целевой функции)
    /// </summary>
    public enum PairType
    {
        /// <summary>
        /// Целевая функция не задана
        /// </summary>
        None,
        /// <summary>
        /// Минимальное значение
        /// </summary>
        Min,
        /// <summary>
        /// Максимальное значение
        /// </summary>
        Max
    }
}

```

Файл SciLibrary\Mathematics\Permutation.cs - Объект "перестановка"

```

using
System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SciExcel.Mathematics
{
    /// <summary>
    /// Построение перестановок
    /// Основано на коде
    /// https://prog-cpp.ru/permutation/
    /// </summary>

```

```

public class Permutation
{
    /// <summary>
    /// Вектор элементов
    /// </summary>
    private int[] x;

    /// <summary>
    /// Номер перестановки, начиная с 1
    /// </summary>
    public int N { get; private set; }

    /// <summary>
    /// Элемент перестановки
    /// </summary>
    /// <param name="i">Номер элемента, начиная с 0</param>
    /// <returns></returns>
    public int this[int i] { get { return x[i]; } }

    /// <summary>
    /// Конструктор перестановки
    /// </summary>
    /// <param name="count">Длина вектора перестановки</param>
    public Permutation(int count)
    {
        N = 1;
        // Формирование начального массива
        x = new int[count];
        for (int i = 0; i < count; i++)
        {
            x[i] = i;
        }
    }

    /// <summary>
    /// Клонировующий конструктор
    /// </summary>
    /// <param name="permutation">Исходная перестановка</param>
    public Permutation(Permutation permutation)
    {
        N = permutation.N;
    }
}

```

```

// Клонирование массива (посмотреть, нельзя ли лучше)
int size = permutation.x.Length;
x = new int[size];
Array.Copy(permutation.x, x, size);
}

/// <summary>
/// Обмен элементов массива
/// </summary>
/// <param name="i">Индекс первого элемента обмена</param>
/// <param name="j">Индекс второго элемента обмена</param>
private void swap(int i, int j)
{
    int s = x[i];
    x[i] = x[j];
    x[j] = s;
}

/// <summary>
/// Построение следующей перестановки
/// </summary>
/// <returns></returns>
public bool Next()
{
    // Увеличение номера перестановки
    N++;

    // Построение перестановки
    int n = x.Length;
    int j = n - 2;
    while (j != -1 && x[j] >= x[j + 1])
    {
        j--;
    }
    if (j == -1)
    {
        // больше перестановок нет
        return false;
    }
    int k = n - 1;
    while (x[j] >= x[k])
    {

```

```

        k--;
    }
    swap(j, k);
    int l = j + 1;
    int r = n - 1;

    // сортируем оставшуюся часть последовательности;
    while (l < r)
    {
        swap(l++, r--);
    }
    return true;
}

/// <summary>
/// Строковое представление перестановки
/// </summary>
/// <returns></returns>
public override string ToString()
{
    return string.Join(" ", x);
}
}
}

```

**Файл SciLibrary\Mathematics\Permutator.cs - Класс для
формирования последовательности перестановок**

```

using
System;

using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SciExcel.Mathematics
{
    /// <summary>
    /// Перестановки из n элементов

```

```

/// Основано на коде
/// https://prog-cpp.ru/permutation/
/// </summary>
public class Permutator : IEnumerable, IEnumerable<Permutation>
{
#if DEBUG
    /// <summary>
    /// Максимальное число итераций при отладке
    /// </summary>
    private const int max = 50;
    /// <summary>
    /// Текущий счетчик итераций при отладке
    /// </summary>
    private int n = 0;
#endif

    /// <summary>
    /// Текущая перестановка
    /// </summary>
    private Permutation current;

    /// <summary>
    /// Количество элементов
    /// </summary>
    private int count;

    /// <summary>
    /// Конструктор перестановок из n элементов
    /// </summary>
    /// <param name="n">Количество элементов в
перестановке</param>
    public Permutator(int n)
    {
        count = n;
        current = new Permutation(n);
    }

    /// <summary>
    /// Количество перестановок
    /// </summary>
    public long N

```

```

    {
        get
        {
            return Function.Factorial(count);
        }
    }

    /// <summary>
    /// Последовательное построение перестановок
    /// </summary>
    /// <returns></returns>
    public IEnumerator GetEnumerator()
    {
#if DEBUG
        n = 0;
#endif
        yield return new Permutation(current);
        while (current.Next())
        {
            yield return new Permutation(current);
#if DEBUG
            // При отладке делаем ограниченное число итераций
            if (n++ > max) { yield break; }
#endif
        }
        yield break;
    }

    /// <summary>
    /// Последовательное построение перестановок
    /// </summary>
    /// <returns></returns>
    IEnumerator<Permutation>
    IEnumerable<Permutation>.GetEnumerator()
    {
#if DEBUG
        n = 0;
#endif
        yield return new Permutation(current);
        while (current.Next())
        {
            yield return new Permutation(current);
#if DEBUG

```

```
        // При отладке делаем ограниченное число итераций
        if (n++ > max) { yield break; }
    #endif
    }
    yield break;
}
}
```

Приложение Б. Акт о внедрении



УТВЕРЖДАЮ

Проректор по учебной работе МГУСИ,

к.т.н., доцент

Е.В. Титов

«15» октября 2021 г.

АКТ

об использовании результатов диссертационной работы А.Ю. Кудряшовой на тему: «Исследование и разработка эффективных методов кодирования источника при преобразовании метрических пространств в условиях действия помех» в учебном процессе кафедры Общей теории связи МГУСИ

Комиссия в составе:

- секретарь кафедры ОТС, старший преподаватель Потапова Е.А.;
- заведующего кафедрой ОТС д.т.н., профессор Аджемов А.С.

удостоверяет, что в учебном процессе кафедры Общей теории связи при чтении курсов лекций, проведении практических занятий, выполнении курсовой работы по дисциплинам: «Теория информации и кодирования», «Теоретические основы инфокоммуникаций» и «Теория информации» используются результаты диссертационной работы Кудряшовой А.Ю., а именно: разработанные аспирантом методические материалы, используемые в лекционных курсах, комплексы практических упражнений и задач, тестирующие задания для поиска наиболее эффективных методов кодирования источника при заданных вероятностных характеристиках источника и возникающих помех. Результаты диссертационной работы нашли практическое воплощение в электронных мультимедийных материалах, а также и в учебных и учебно-методических пособиях:

1. Аджемов А.С., Кудряшова А.Ю., Поборчая Н.Е., Потапова Е.А., Вероятностно-информационные характеристики случайных сигналов и эффективное кодирование дискретных сообщений в системе передачи информации: для студентов 3 к. ЦЗОПБ. Учебно-методическое пособие и курсовая работа, 2021г., 35с.

2. Аджемов А.С., Кудряшова А.Ю., Поборчая Н.Е., Потапова Е.А., Теория информации учебно-методическое пособие для бакалавров заочной формы обучения, 2021г., 62 с.

3. Аджемов А.С., Кудряшова А.Ю., Поборчая Н.Е., Потапова Е.А., Основы теории информации. Теория кодирования и шифрования. Учебно-методическое пособие и курсовая работа, 2020г., 44 с.

4. Аджемов А.С., Санников В.Г., Общая теория связи. Учебно-методическое пособие и курсовая работа, 2019г., 42 с.

Заведующий кафедрой ОТС, д.т.н., профессор

А.С. Аджемов

Секретарь кафедры ОТС, ст. преподаватель

Е.А. Потапова

**Приложение В. Свидетельство о государственной регистрации
программы для ЭВМ**

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО
о государственной регистрации программы для ЭВМ

№ 2021680147

**Программа для оценки методов кодирования источника
при преобразовании метрических пространств и
действии помех**

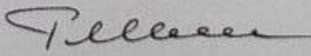
Правообладатель: *Ордена Трудового Красного Знамени
федеральное государственное бюджетное образовательное
учреждение высшего образования "Московский технический
университет связи и информатики" (МТУСИ) (RU)*

Авторы: *Кудряшова Анастасия Юрьевна (RU), Аджемов Артем
Сергеевич (RU)*

Заявка № **2021669538**
Дата поступления **29 ноября 2021 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ **07 декабря 2021 г.**

Руководитель Федеральной службы
по интеллектуальной собственности




Г.П. Излиев