МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени К.И. Сатпаева

Институт Информационных и телекоммуникационных технологий

Кафедра Электроники, телекоммуникации и космических технологии

Анализ оптимального декодера для линейных кодов

ДИПЛОМНАЯ РАБОТА

специальность 5В071900 – Радиотехника, электроника и телекоммуникация

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени К.И. Сатпаева

Институт Информационных и телекоммуникационных технологий

Кафедра Электроники, телекоммуникации и космических технологии

ДОПУЩЕН К ЗАЩИТЕ
Завсдующий кафедрой ЭТиКТ
канд.техн. наук
_____ Е.Таштай
«15» 05 2019 г.

ДИПЛОМНАЯ РАБОТА

На тему: Анализ оптимального декодера для линейных кодов

по специальности 5В071900 – Радиотехника, электроника и телекоммуникация

Выполнил (а) Н.Е.Толеген

Рецензент
доцент каф. «ТКС» АУЭС,
канд.техн.наук
_____ А.О.Касимов
«15» 05 2019 г.

Ассоциированный профессор
ЭТиКТ , канд.техн.наук
_____ Д.Б.Илипбаева
«___» _____ 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени К.И. Сатпаева
Институт Информационных и телекоммуникационных технологий

Кафедра Электроники, телекоммуникации и космических технологии

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой ЭТиКТ
канд.техн. наук
_____Е.Таштай
«10» 02 2018 г.

ЗАДАНИЕ

на выполнение дипломной работы

Обучающемуся *Толеген Нурисламу Ергалиевичу*
Тема *Анализ оптимального декодера для линейных кодеров*
Утверждена *приказом ректора университета* № 1162-б от " 16 "__10__2018г.
Срок сдачи законченной работы "___"_____2019г.
Исходные данные к дипломной работе: *(1,3) RM код в матрице кодирования* R (r, m)
Краткое содержание дипломной работы:
а) *Аналитический обзор оптимального декодирования для линейных кодов*
б) *Классификация линейных кодов. Методика построения RM кодов*
в) *Расчет кодирования и декодирования RM кода*
Перечень графического материала (с точным указанием обязательных чертежей):
представлены _____ слайдов презентации работы
Рекомендуемая основная литература: *из ___ наименований*

- Shu Lin , Daniel J ,Costello , Jr - Error Control Coding «PEARSON»
- Б.Д.Кудряшов - Основы теории кодирования «БХВ-Петербург»
- А. М. Голиков - Модуляция , кодирование и моделирование в телекоммуникационных системах «ЭБС ЛАНЬ»
- В. А. Варгаузин , И. А. Цикин - Методы повышения энергетической и спектральной эффективности цифровой радиосвязи «БХВ-Петербург»

ГРАФИК

подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
Анализ организации беспроводной сети	10.01.2019 г. -17.02.2019 г.	выполнена
Характеристика систем широкополосного беспроводного доступа	17.03.2019 г. -24.03.2019 г.	выполнена
Приемопередающая всенаправленная антенна Omni-5.3-10	24.03.2019 г. -28.03.2019 г.	выполнена

Подписи

консультантов и нормоконтролера на законченную дипломную работу (проект) с указанием относящихся к ним разделов работы (проекта)

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормоконтролер	Смайлов Н.К сениор-лектор каф.ЭТиКТ	16.05.19	

Научный руководитель _____ Л.Б.Илипбаева
(подпись)

Задание принял к исполнению обучающийся _____ Н.Е.Толеген
(подпись)

Дата "14" 05 2019 г.

АННОТАЦИЯ

Целью данного дипломного проекта является анализ оптимального декодера для линейных кодеров. В проекте рассмотрены усреднение шума для уменьшения ошибок, а также кодовая скорость и компромисс между надежностью и скоростью передачи данных.

В расчётной части проекта приведены алгоритмы кодирования основанные на надежности линейных блочных кодов с мягким решением. Дан расчёт кодирования и декодирования RM кода.

АНДАТПА

Дипломдық жобаның мақсаты желілік кодерлер ушін оңтайлы декодерді талдау болып табылады. Жоба шуды темендетуге, сондай-ак коэффициент жылдамдығын жэне сенімділік пен деректерді беру жылдамдығының арақатынасын темендетуді қарастырады.

Жобаның есептеу бөлігінде жұмсақ шешіммен сызықтық блоктық кодтардың сенімділігіне негізделген алгоритмдерді кодтау царастырылған. RM кодын кодтау және декодтау есебі жасалды.

ANNOTATION

The purpose of this graduation project is to analyze the optimal de-coder for linear coders. The project considers noise averaging to reduce errors, as well as the code rate and the trade-off between reliability and data transfer rate.

In the computational part of the project, coding algorithms based on the reliability of linear block codes with a soft solution are given. Given the calculation of encoding and decoding RM code.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Практически невозможно представить современные системы связи без использования корректирующих кодов, а также у большого количества систем связи есть проблемы ограничениями на длину кода коррекции ошибок пользователя, которая определяется требованиями минимальной задержки передачи информации.

Декодеры и кодеры применяются для передачи данных. Для любой операции с информацией (даже такой простой, как сохранение) она должна быть каким-то образом представлена (записана, исправлена). Этот процесс имеет специальное название – кодирование информации. Как правило, процесс кодирования преобразует информацию из формы, удобной для непосредственного использования, в форму, удобную для передачи, хранения или автоматической переработки. В более узком смысле кодирование информации называется представлением информации в виде кода.

Целью данной дипломной работы является полный анализ кодирования и декодирования линейных кодов и методы мягкого декодирования.

В начале работы проводиться краткий аналитический обзор помехоустойчивых кодов, применение и его классификация.

Далее определяются алгоритмы кодирования основанных на линейных кодах с мягким решением. Описывается более подробно декодирование с мягким решением, учитываются достоинства и недостатки декодирования с мягким решением. Сравнивается Проводятся расчеты кодирования кода и декодирования кода Рида Миллера.

1 Обзор кодирования блочных линейных помехоустойчивых кодов

Кодирование управления ошибками (Error Control Coding - ECC) - дисциплина теории информации, введенная Клодом Элвудом Шенноном в 1948 году [6]. В своей эпохальной работе Шеннон показал, что шум канала ограничивает скорость передачи, а не вероятность ошибки. Следовательно, возможно разработать безошибочную систему связи с использованием кодирования управления ошибками, где существует максимальная скорость, с которой данные могут передаваться по шумному каналу связи заданной полосы пропускания без ошибок. ECC направлена на разработку методов кодирования для достижения обнаружения ошибок и построения исходных безошибочных данных. На рисунке 1 показана блок-схема системы передачи данных, в которой блоки кодера и декодера отвечают за кодирование и декодирование передаваемой и принимаемой последовательности данных соответственно. Вклад Шеннона состоял в том, чтобы доказать существование таких кодов, и поэтому он был отправной точкой исследования кодирования контроля ошибок. С тех пор много исследований было посвящено оптимизации методов кодирования и декодирования для контроля ошибок в шумных средах [5].

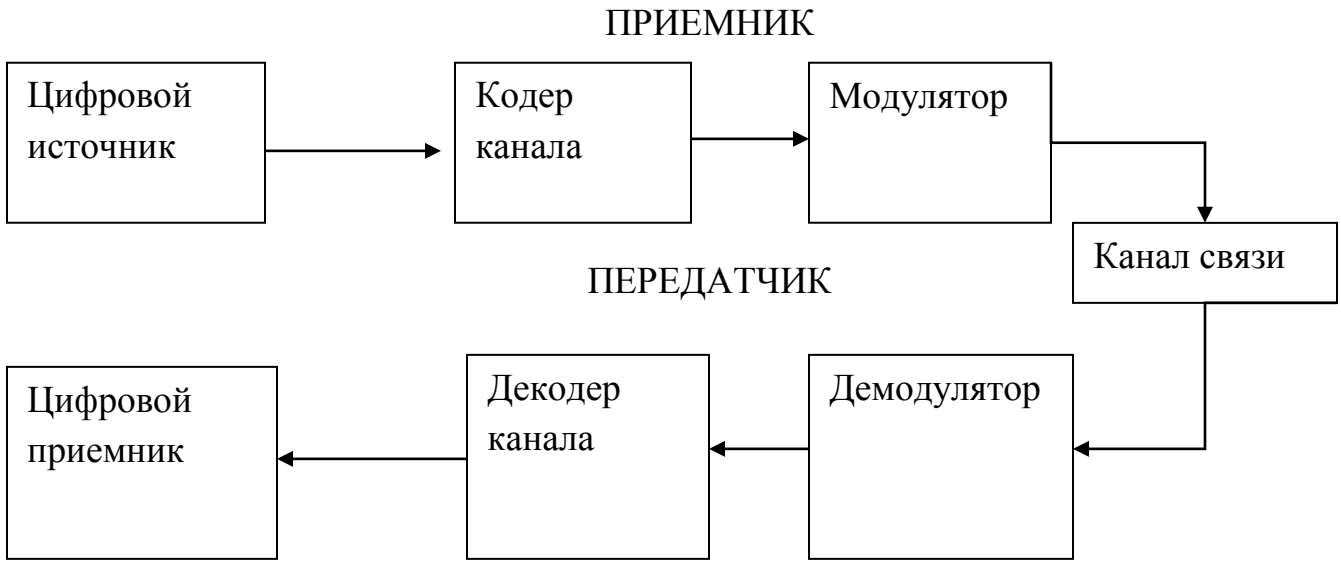

Рисунок 1.1 - Структурная схема системы передачи данных

Одним из наиболее широко используемых методов контроля ошибок является прямая коррекция ошибок. Как и большинство методов управления ошибками, основная идея прямой коррекции ошибок – добавить некоторую избыточность к исходному сообщению, которое приемники могут использовать для проверки согласованности доставленного сообщения и восстановления

поврежденных данных. Таким образом, хотя, часть эффективной скорости передачи данных ограничена. Таким образом, ключевой метрикой для этих кодов является "кодовая скорость" R, которая выражает отношение битрейта без прямой коррекции ошибок к битовой скорости с прямой коррекцией ошибок (R=k/n, для каждого к бита информации будут отправлены n битов данных, из которых n−k являются избыточными) [2]

В вычислительной технике, телекоммуникациях, теории информации и теории кодирования помехоусточивого кода, иногда помехоусточивый код, используется для контроля ошибок данных которые передавались по ненадежным или шумным каналам связи. Центральная идея заключается в том, что отправитель кодирует сообщение с избыточным минимумом в виде помехоусточивого кода. Американский математик Ричард Хэмминг был одним из первых ученых в этой области в 1940-х годах и изобрел первый код для исправления ошибок в 1950 году: код Хэмминга (7,4). Избыточность позволяет получателю обнаруживать ограниченное количество ошибок, которые могут возникнуть в любом месте сообщения, и часто исправлять эти ошибки без повторной передачи. Помехоусточивый код дает приемнику возможность исправлять ошибки без необходимости обратного канала для запроса повторной передачи данных, но за счет фиксированной, более высокой пропускной способности Прямого канала. Поэтому помехоусточивый код применяется в ситуациях, когда повторная передача является дорогостоящей или невозможной, например в случае односторонних линий связи, и при передаче на несколько приемников в многоадресной передаче. Например, в случае спутника, вращающегося вокруг Урана, повторная передача из-за ошибок декодирования может создать задержку в 5 часов. Информация помехоусточивого кода обычно добавляется в запоминающие устройства для восстановления поврежденных данных, широко используется в модемах и используется в системах, где основной памятью является запоминающее устройство с исправлением ошибок. [3]

Обработка помехоусточивого кода в приемнике может применяться к цифровому битовому потоку или при демодуляции цифровой модулированной несущей. Для последнего помехоусточивого кода является неотъемлемой частью начального аналого-цифрового преобразования в приемнике. Декодер Витерби реализует алгоритм мягкого решения для демодуляции цифровых данных из аналогового сигнала, поврежденного шумом. Многие кодеры/декодеры помехоусточивого кода также могут генерировать сигнал частоты битовых ошибок, который может использоваться в качестве обратной связи для тонкой настройки аналоговой приемной электроники. Максимальные доли ошибок или недостающих битов, которые могут быть исправлены, определяются конструкцией кода ECC, поэтому различные коды исправления ошибок подходят для разных условий. В целом, более сильный код вызывает больше избыточности, которая должна передаваться с использованием доступной полосы пропускания, что снижает

эффективную скорость передачи битов при одновременном улучшении полученного эффективного отношения сигнал / шум. Теорема кодирования шумного канала Клода Шеннона отвечает на вопрос о том, сколько пропускной способности остается для передачи данных при использовании наиболее эффективного кода, который превращает вероятность ошибки декодирования в ноль. Это устанавливает границы теоретической максимальной скорости передачи информации канала с некоторым заданным базовым уровнем шума. Однако доказательство не является конструктивным и, следовательно, не дает представления о том, как построить код для достижения потенциала. После лет исследования, некоторые предварительные системы ECC в наше время приходят очень близко к теоретическому максимуму. [3]

ECC осуществляется путем добавления избыточности к передаваемой информации с помощью алгоритма. Избыточный бит может быть сложной функцией многих исходных информационных битов. Исходная информация может появляться или не появляться буквально в закодированном выходе; коды, которые включают немодифицированный вход в выход, являются систематическими, а те, которые не включают такие входы и выходы являются несистематическими. Упрощенным примером ECC является передача каждого бита данных 3 раза, который известен как (3,1) код повторения. Через шумный канал, приемник может видеть 8 версий выпуска, см. таблицу ниже.

Таблица 1.1 8-Серии выпуска которые может видеть приемник

Принятый код	Расшифровывается как
000	0 (без ошибок)
001	0
010	0
100	0
111	1 (без ошибок)
110	1
101	1
011	1

Это позволяет исправить ошибку в любой из трех выборок путем "голосования большинством" или "демократического голосования".

Корректирующая способность этого запоминающего устройства с исправлением ошибок. До 1 бита триплета по ошибке, или опущено до 2 бит триплета (случаи не показаны в таблице).

Он прост в реализации и широко используется, это тройное модульное резервирование является относительно неэффективным и так далее. Лучшие коды ECC обычно исследуют последние несколько десятков или даже последние

несколько сотен ранее полученных битов, чтобы определить, как декодировать текущую небольшую горстку битов (обычно в группах от 2 до 8 бит) [9].

1.1 Усреднение шума для уменьшения ошибок

Можно сказать, что ECC работает путем "усреднения шума"; поскольку каждый бит данных влияет на многие передаваемые символы, повреждение некоторых символов шумом обычно позволяет извлекать исходные пользовательские данные из других, неповрежденных полученных символов, которые также зависят от тех же пользовательских данных.

Из-за этого эффекта" объединения рисков " цифровые системы связи, использующие ECC, как правило, работают значительно выше определенного минимального отношения сигнал / шум, а не ниже него.

Эта тенденция "все или ничего" – эффект обрыва – становится более выраженной по мере того, как используются более сильные коды, приближающиеся к теоретическому пределу Шеннона.

Чередование кодированных данных ECC может уменьшить все или ничего свойства передаваемых кодов ECC, когда ошибки канала, как правило, происходят в пакетах. Однако этот метод имеет ограничения; его лучше всего использовать для узкополосных данных.

Большинство телекоммуникационных систем используют фиксированный код канала, предназначенный для того, чтобы выдерживать ожидаемую частоту битовых ошибок в худшем случае, а затем вообще не работать, если частота битовых ошибок еще хуже. Однако некоторые системы адаптируются к заданным условиям ошибок канала: некоторые экземпляры гибридного автоматического повторного запроса используют фиксированный метод ECC, пока ECC может обрабатывать частоту ошибок, а затем переключаться на ARQ, когда частота ошибок становится слишком высокой; адаптивная модуляция и кодирование использует различные скорости ECC, добавляя больше битов коррекции ошибок на пакет, когда есть более высокие частоты ошибок в канале, или принимая их, когда они не нужны[14].

1.2 Классификация помехоустойчивых кодов

Двумя основными категориями кодов ECC являются блочные коды и сверточные коды.

Блочные коды работают на блоках фиксированного размера (пакетах) битов

или символов заданного размера. Практические блочные коды обычно могут быть жестко декодированы за полиномиальное время до их длины блока.

Сверточные коды работают на битовых или символьных потоках произвольной длины. Они чаще всего мягко декодируются с помощью алгоритма Витерби, хотя иногда используются другие алгоритмы. Декодирование Витерби позволяет асимптотически оптимизировать эффективность декодирования с увеличением длины ограничения сверточного кода, но за счет экспоненциально возрастающей сложности. Сверточный код, который завершается, также является "блочным кодом", поскольку он кодирует блок входных данных, но размер блока сверточного кода обычно произволен, в то время как блочные коды имеют фиксированный размер, продиктованный их алгебраическими характеристиками. Типы прекращения для сверточных кодов включают "хвост-кусать"и" бит-топить".

Существует много типов блочных кодов, но среди классических наиболее заметным является кодирование Рида-Соломона из-за его широкого использования в компакт-дисках, DVD и жестких дисках. Другие примеры классических блочных кодов включают коды Голея, BCH, многомерной четности и Хэмминга.

Хэмминг ECC обычно используется для исправления ошибок флэш-памяти NAND.[3] это обеспечивает одноразрядное исправление ошибок и 2-разрядное обнаружение ошибок. Коды Хэмминга подходят только для более надежных одноуровневых ячеек (SLC) NAND. Более плотная многоуровневая ячейка (MLC) NAND требует более сильной многоразрядной коррекции ECC, такой как BCH или Reed-Solomon. [сомнительно – обсуждение] ни Flash, как правило, не использует исправления ошибок [4].

Классические блочные коды обычно декодируются с использованием алгоритмов жесткого решения[6], что означает, что для каждого входного и выходного сигнала принимается жесткое решение, соответствует ли он одному или нулевому биту. Напротив, сверточные коды обычно декодируются с использованием алгоритмов мягких решений, таких как алгоритмы Viterbi, MAP или BCJR, которые обрабатывают (дискретизированные) аналоговые сигналы и которые позволяют значительно повысить производительность коррекции ошибок, чем декодирование жестких решений.

Почти все классические блочные коды применяют алгебраические свойства конечных полей. Поэтому классические блочные коды часто называют алгебраическими кодами.

В отличие от классических блочных кодов, которые часто определяют способность обнаруживать или исправлять ошибки, многие современные блочные коды, такие как LDPC-коды, не имеют таких гарантий. Вместо этого современные коды оцениваются с точки зрения их частоты битовых ошибок.

Большинство кодов исправления ошибок вперед исправляют только битовые сальто, но не битовые вставки или битовые удаления. В этом параметре расстояние Хэмминга является подходящим способом измерения частоты битовых ошибок.

Несколько кодов прямой коррекции ошибок предназначены для исправления битовых вставок и битовых удалений, таких как коды маркеров и водяных знаков. Расстояние Левенштейна является более подходящим способом измерения частоты битовых ошибок при использовании таких кодов[13].

В теории информации турбо-коды (первоначально во французских Турбокодах)-это класс высокопроизводительных кодов прямой коррекции ошибок (FEC), разработанных около 1990-91 годов (но впервые опубликованных в 1993 году), которые были первыми практическими кодами, близко приближающимися к пропускной способности канала, теоретическому максимуму для скорости кода, при которой надежная связь все еще возможна при определенном уровне шума. Турбо-коды используются в мобильной связи 3G/4G (например, в UMTS и LTE) и в спутниковой связи (в глубоком космосе), а также в других приложениях, где разработчики стремятся обеспечить надежную передачу информации по каналам связи с ограниченной пропускной способностью или задержкой при наличии искажающих данные помех. Турбо-коды конкурируют с LDPC-кодами, которые обеспечивают аналогичную производительность.

Существует множество различных экземпляров турбо-кодов, использующих различные кодеры компонентов, коэффициенты ввода/вывода, перемежители и шаблоны проколов. В этом примере реализации энкодера описывается классический турбо-энкодер и демонстрируется общая конструкция параллельных турбо-кодов.

Эта реализация кодера отправляет три подблока битов. Первый подблок - это m-битный блок данных полезной нагрузки. Второй подблок - это N / 2 бита четности для данных полезной нагрузки, вычисленных с использованием рекурсивного систематического сверточного кода (RSC-кода). Третий подблок - это N / 2 бита четности для известной перестановки данных полезной нагрузки, снова вычисленных с использованием кода RSC. Таким образом, с полезной нагрузкой отправляются два избыточных, но разных подблока битов четности. Полный блок имеет m + n битов данных со скоростью кода m / (m + n). Перестановка данных полезной нагрузки выполняется устройством, называемым перемежителем. Аппаратно этот кодер турбо-кода состоит из двух идентичных RSC-кодеров, C1 и C2, как показано на рисунке, которые соединены друг с другом с помощью схемы конкатенации, называемой параллельной конкатенацией:

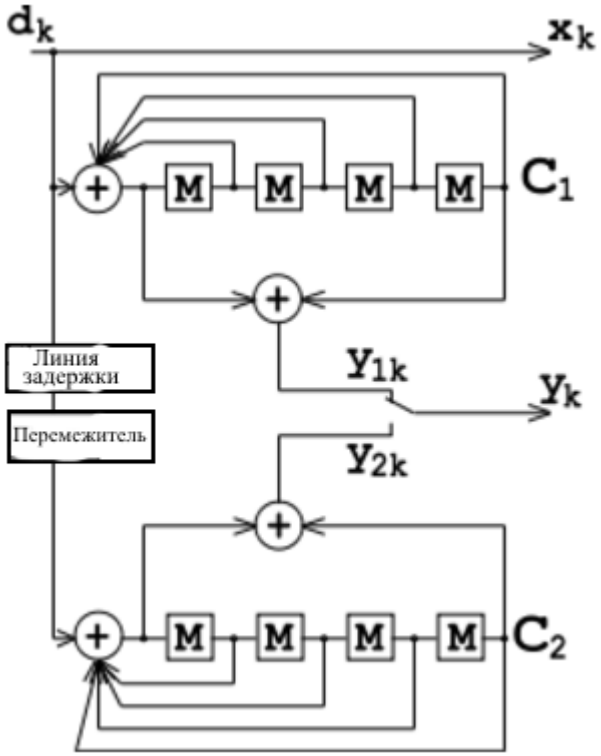

Рисунок 1.2 – Схема кодера турбо кода

На рисунке М-это регистр памяти. Линия задержки и сила перемежителя вводят биты d_k, чтобы появиться в разных последовательностях. На первой итерации входная последовательность d_k появляется на обоих выходах кодировщика, x_k и y_1 или y_2 из-за систематического характера кодировщика. Если кодеры C_1 и C_2 используются в N_1 и N_2 итерациях, то их скорости соответственно равны[7].

1.3 Кодовая скорость и компромисс между надежностью и скоростью передачи данных

Основной принцип ECC заключается в добавлении избыточных битов, чтобы помочь декодеру узнать истинное сообщение, которое было закодировано передатчиком. Кодовая скорость данной системы ECC определяется как скорость между количеством информационных битов и общим количеством битов (т. е. информация плюс избыточные биты) в данном пакете связи. Следовательно, кодовая скорость-это действительное число. Низкая скорость кода, близкая к нулю, подразумевает сильный код, который использует много избыточных битов для достижения хорошей производительности, в то время как большая скорость кода, близкая к 1, подразумевает слабый код.

Избыточные биты, защищающие информацию, должны передаваться с использованием тех же коммуникационных ресурсов, которые они пытаются защитить. Это приводит к фундаментальному компромиссу между надежностью и скоростью передачи данных[8]. в одном крайнем случае сильный код (с низкой скоростью кода) может вызвать значительное увеличение SNR приемника, уменьшающее частоту битовых ошибок, за счет снижения эффективной скорости передачи данных. С другой стороны, не используя какой-либо ECC (т. е. кодовую скорость, равную 1), использует полный канал для целей передачи информации за счет оставления битов без какой-либо дополнительной защиты[12].

Один интересный вопрос заключается в следующем: насколько эффективным с точки зрения передачи информации может быть ECC, который имеет незначительную частоту ошибок декодирования? На этот вопрос ответил Клод Шеннон со своей второй теоремой, которая говорит, что емкость канала является максимальной скоростью передачи битов, достижимой любым ECC, частота ошибок которого стремится к нулю:[9] его доказательство опирается на гауссовское случайное кодирование, которое не подходит для реальных приложений. Эта верхняя граница, данная работой Шеннона, создала долгий путь в разработке ECC, который может приблизиться к конечной границе производительности. Различные коды сегодня могут достигать почти предела Шеннона. Однако создание кодов , как правило, чрезвычайно сложно осуществить.

Самые популярные ECC имеют компромисс между производительностью и вычислительной сложностью. Обычно их параметры дают диапазон возможных кодовых скоростей, которые могут быть оптимизированы в зависимости от сценария. Обычно эта оптимизация выполняется для достижения низкой вероятности ошибки декодирования при минимизации влияния на скорость передачи данных. Другим критерием оптимизации скорости передачи кода является сбалансированность низкой частоты ошибок и числа повторных передач с учетом стоимости энергии связи.[10]

1.4 Объединенные помехоустойчивые коды для повышения производительности

Классические (алгебраические) блочные коды и сверточные коды часто объединяются в конкатенированные схемы кодирования, в которых большую часть работы выполняет короткий сверточный код с ограничением длины Витерби-декодированный код, а блочный код (обычно Рид-Соломон) с большим размером символа и длиной блока "зачистка" любые ошибки, сделанные сверточным декодером. Однопроходное декодирование с этим семейством кодов коррекции ошибок может дать очень низкие частоты ошибок, но для условий

передачи на большие расстояния (например, в глубоком космосе) рекомендуется итеративное декодирование.

1.5 Обзор кодов методов кодирования блочных линейных кодов

Коды Bose-Chaudhuri-Hocquenghem (БЧХ) предлагают гибкость в выборе длины блока и скорости кода, и могут быть конструированы для коррекции любого, заданного числа ошибок. Быстрый алгоритм декодирования может быть использован для жесткого декодирования кодов BCH. Для любого целого числа

$$m \geq 3 \text{ и } 0 < t < 2m-1 \tag{1.1}$$

существует t-исправляющий ошибку BCH (n, k) код с

$$n = 2m-1 \text{ и } n-k \leq mt, \tag{1.2}$$

минимальное расстояние dmin которого ограничено следующим образом:

$$2t+1 \leq dmin \leq 2t+2. \tag{1.3}$$

Коды BCH можно определить в двоичном поле, таком как коды Хэмминга, и в недвоичном (символьном) поле, таком как коды Рида-Соломона.

Для одиночных кодов с исправлением ошибок, если общее число битов в переданном кодовом слове равно n, то

$$m = n-k \tag{1.4}$$

контрольных битов должны быть способны указывать по крайней мере n+1 различных состояний. Из них одно состояние означает отсутствие ошибки, а n состояний указывают местоположение ошибки в каждой из n позиций, где также возможно наличие ошибки в самих битах избыточности. Таким образом, n+1 состояний должны быть обнаружены N−K битами, а n−k бит могут указывать на 2n−k различных состояний. Поэтому мы должны иметь

$$2n-k \geq n+1 \tag{1.5}$$

или эквивалентно

$$2m-1 \geq n \text{ для кода } (n, k) \tag{1.6}$$

с единственной возможностью исправления ошибок.

Коды Хэмминга имеют

$$d_{min}=3, \quad (1.7)$$

и, следовательно, t=1, т. е. одна ошибка может быть исправлена независимо от количества битов проверки на четность. Код Хэмминга (n, k) имеет m=n−K битов проверки на четность, где

$$n=2m-1 \text{ и } k=2m-1-m, \text{ при } m\geq 3. \quad (1.8)$$

Матрица проверки на четность H кода Хэмминга имеет m строк и n столбцов, и последние n−K столбцов должны быть выбраны таким образом, чтобы она формировала матрицу идентификаторов. Ни один столбец не состоит из всех нулей; каждый столбец уникален и имеет m элементов. В виду этого, синдром всех одиночных ошибок будет различен и одиночные ошибки можно обнаружить. При увеличении числа битов проверки на четность K возможность исправления ошибок остается неизменной (т. е. t=1), но скорость кода kn улучшается, конечно, за счет дополнительной сложности кодирования и декодирования[18].

Найдите матрицу проверки на четность, матрицу генератора и все 16 кодовых слов для кода (7, 4) Хэмминга. Определить синдром, если полученное кодовое слово а) 0001111 и Б) 0111111.

Матрица H с проверкой на четность состоит из всех двоичных столбцов, кроме нулевой последовательности, поэтому мы имеем ее в следующем виде:

H=110110110111/100/010/001

и соответствующая генераторная матрица G выглядит следующим образом:

G=1000010000100001||||110110110111

Полученные кодовые слова перечислены в следующей таблице 1.2:

Таблица 1.2 Значения кодовых слов

Сообщение (M)	Кодовое слово (C)
0000	0000000
0001	0001111
0010	0010011
0011	0011100
0100	0100101

0101	0101010
0110	0110110
0111	0111001
1000	1000110
1001	1001001
1010	1010101
1011	1011010
1100	1100011
1101	1101100
1110	1110000
1111	1111111

$$S=RHT=000.$$

Поскольку синдром является нулевым вектором, ошибок в кодовом слове нет.

$$S=RHT=110.$$

Поскольку синдром соответствует первой строке столбца H, Первый БИТ полученного кодового слова находится в ошибке (т. Е. переданное кодовое слово было 1111111).

Коды Рида-Соломона (RS) являются недвоичными циклическими кодами с символами, каждый из которых состоит из M-битов, где $M \geq 1$. Код Рида-Соломона (n, k) используется для кодирования K символов в блоки

$$N=2m-1 \quad (1.9)$$

символов путем добавления $N-K$ символов четности, где каждый символ состоит из M битов. Код Рида-Соломона с возможностью исправления ошибок T имеет

$$N-k=2T \quad (2)$$

символов проверки на четность и минимальное расстояние

$$dmin=2t+1. \quad (2.1)$$

Коды Рида-Соломона достигают максимально возможного минимального расстояния для любого линейного блочного кода с одинаковой длиной входного и выходного блоков кодера, поскольку они могут высокоэффективно использовать

избыточность. Длины блоков и размеры символов могут быть легко скорректированы для удовлетворения различных размеров входных сообщений. Уникальной и ценной заслугой кодов Рида-Соломона является их способность исправлять ошибки пакетов. С помощью эффективных методов декодирования кодов Рида-Соломона имеют широкое применение. Фактически, они имеют приложения в области дальней космической связи, беспроводной связи, такой как WiMAX, запоминающих устройств, таких как компакт-диски, DVD-диски, Blu-ray диски и цифровые системы видеовещания, в виде объединенных кодов.

Определите параметры 8-разрядного RS-кода, исправляющего ошибки 16 символов.

Дано m=8, как число битов в символе, мы, таким образом, имеем

$$n=28-1=255$$

символов в кодовом слове. При n=255 и t=16 имеем

$$K=n-2t=255-32=223$$

символа в кодовом слове. Поэтому мы имеем кодовую скорость

$$R=kn=223255\cong 0,875.$$

Общее количество бит в кодовом слове составляет

$$255\times 8=2040 \text{ бит.}$$

Поскольку этот код может исправить 16 символов, он может, таким образом, исправить последовательных битов.

$$16\times 8=128$$

Таким образом, этот 8-битный код Рида-Соломона чрезвычайно эффективен для исправления ошибок. Однако, если ошибки случайны, и есть, самое большее, одна ошибка на символ, то этот код может исправить только 16-битные ошибки в 2040 битах, следовательно, не эффективный код для исправления случайных ошибок[20].

Чередование часто используется в цифровых системах связи и хранения для повышения производительности прямого исправления ошибок кодов. Многие каналы связи не являются беспамятными: ошибки обычно возникают в виде пакетов, а не независимо. Если количество ошибок в кодовом слове превышает возможности кода для исправления ошибок, восстановить исходное кодовое слово

не удается. Чередование устраняет эту проблему путем перетасовки исходных символов между несколькими кодовыми словами, создавая тем самым более равномерное распределение ошибок.[12] поэтому перемежение широко используется для исправления ошибок пакетов.

1.6 Проверка четности низкой плотности (LDPC)

Коды проверки четности низкой плотности (LDPC) - это класс высокоэффективных линейных блочных кодов, выполненных из многих кодов проверки четности (SPC). Они могут обеспечить производительность, очень близкую к пропускной способности канала (теоретический максимум), используя итерационный подход декодирования мягких решений, при линейной сложности времени с точки зрения их длины блока. Практические реализации в значительной степени зависят от параллельного декодирования составляющих кодов SPC.

Коды LDPC были впервые представлены Робертом г. Галлагером в его кандидатской диссертации в 1960 году, но из–за вычислительных усилий по внедрению кодера и декодера и внедрению кодов Рида-Соломона они в основном игнорировались до 1990-х годов.

LDPC - коды теперь используются во многих последних высокоскоростных коммуникационных стандартов, таких как DVB-C2 (цифровое видео вещание – спутниковое – второе поколение), технологии WiMAX (стандарт IEEE 802.16 E в стандартной для СВЧ-связь), высокоскоростной беспроводной локальной сети (IEEE 802.11 n) и[11] локальные сети 10GBASE-T локальные сети (802.3 an) и Г. ны/г. 9960 (МСЭ-Т стандарт для сети линии электропередач, телефонные линии и коаксиальный кабель). Другие коды LDPC стандартизированы для стандартов беспроводной связи в пределах 3GPP MBMS .

Анализ современных итерационных кодов, таких как турбо-коды и LDPC-коды, обычно предполагает независимое распределение ошибок. Системы, использующие коды LDPC, поэтому обычно используют дополнительное чередование символов внутри кодового слова.

Для турбо-кодов чередование является неотъемлемым компонентом, и его правильный дизайн имеет решающее значение для хорошей производительности. Итеративный алгоритм декодирования работает лучше всего, когда в графе факторов, представляющем декодер, нет коротких циклов; перемежитель выбирается, чтобы избежать коротких циклов.

Конструкций устройство укладки включают:

- прямоугольные (или равномерные) чередующие элементы (по аналогии с описанным выше методом с использованием коэффициентов пропуска)
- сверточные переплетатели

- случайные перемежители (где перемежитель-известная случайная перестановка)
- S-случайный перемежитель (где перемежитель-известная случайная перестановка с ограничением, что никакие входные символы на расстоянии S не появляются на расстоянии S на выходе).

Другой возможной конструкцией является бесконтактный квадратичный полином перестановки (QPP). Он используется, например, в стандарте мобильной связи 3GPP Long Term Evolution.

В системах связи с несколькими несущими может использоваться чередование несущих для обеспечения частотного разнесения, например для уменьшения частотно-селективных замираний или узкополосных помех[21].

2 Алгоритмы кодирования основанные на надежности линейных блочных кодов с мягким решением

Все алгоритмы разработанные до настоящего времени, основаны на жестких решениях согласованного фильтра в демодуляторе приемника; то есть это означает что выход согласованного фильтра в для каждого интервала сигнализации квантуются на двух уровнях, обозначаемых как 0 и 1, которые представляют собой двоичную принятую последовательность с жестким решением. После декодер обрабатывает результаты в двоичной принятой последовательности на основе конкретного метода декодирования. Этот тип декодирования называется декодированием с жестким решением. Декодирование с жестким решением с использованием алгебраических структур кодов называется алгебраическим декодированием. Расстояние, которая используется в декодировании с жестким решением является расстоянием Хэмминга. Задача состоит в том, чтобы декодировать принятую последовательность жесткого решения в самое близкое кодовое слово на расстояние Хемминга.

Твердое решение о принятом сигнале приводит к потере информации, что приводит к снижению производительности. Предположим, что наша модель связи состоит из кодера четности, канала связи (ослабляет данные случайным образом) и декодера жесткого решения

Биты сообщения "01 “применяются к кодеру четности, и можно получить "011" в качестве выходного кодового слова.

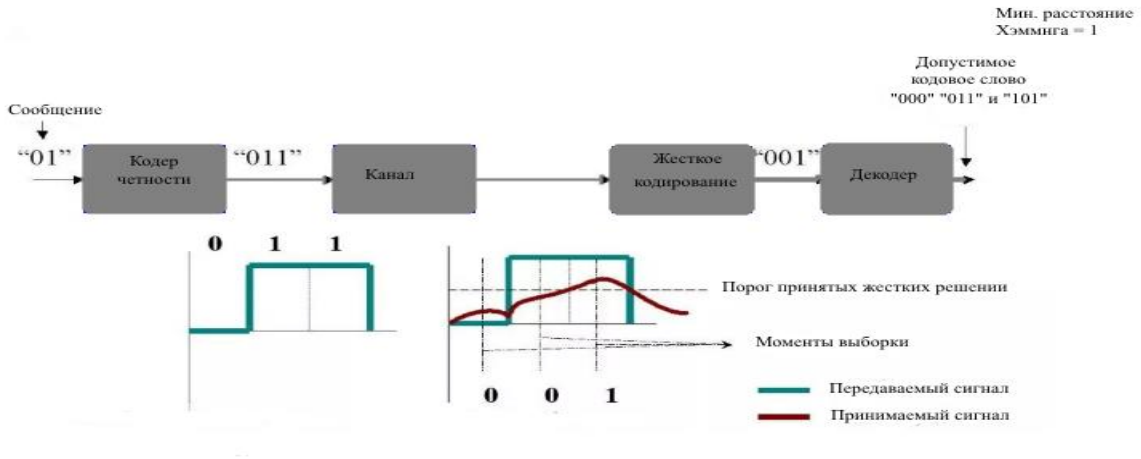

Рисунок 2.1 Структурная схема декодирования с жестким решением

Выходное кодовое слово " 011 " передается по каналу. “0 "передается как" 0 вольт " и "1 “Как "1 Вольт". Канал ослабляет сигнал, который передается, и приемник видит искаженную форму волны (”красный цвет волны"). Декодер

жесткого решения принимает решение на основе порогового напряжения. В нашем случае пороговое напряжение выбирается как 0,5 Вольта (на полпути между “0” и “1” вольтами) . В каждый момент выборки в приемнике (как показано на рисунке выше) детектор жесткого решения определяет состояние бита как “0”, если уровень напряжения падает ниже порога, и “1”, если уровень напряжения выше порога. Таким образом, выход блока жестких решений - “001”. Возможно, этот вывод " 001 " не является допустимым кодовым словом (сравните это со всеми возможными кодовыми словами , приведенными в таблице выше), что означает, что биты сообщения не могут быть восстановлены должным образом. Декодер сравнивает выход блока жестких решений со всеми возможными кодовыми словами и вычисляет минимальное расстояние Хэмминга для каждого случая[18].

Расчет минимального расстояния Хэмминга.

Даны все возможные кодовые слова :

1. 000
2. 001
3. 101
4. 110

Им соответствуют значения выхода с жестким решением :

1. 000 – 001
2. 001 – 001
3. 101 – 001
4. 110 – 001

От сюда следует что Хэмминговые расстояния равны :

1. 000 – 1
2. 001 – 1
3. 101 – 1
4. 110 – 3
5.

Задача декодера-выбрать допустимое кодовое слово, которое имеет минимальное расстояние Хэмминга. В нашем случае минимальное расстояние Хэмминга - "1", и есть 3 допустимых кодовых слова с этим расстоянием. Декодер может выбрать любую из трех возможностей, и вероятность получения правильного кодового слова (”001 " – это то, что мы передали) всегда равна 1/3. Поэтому, когда используется декодирование жесткого решения, вероятность восстановления наших данных (в данном конкретном случае) составляет 1/3.

Если выходные сигналы согласованного фильтра не квантуются или квантованы в более двух уровнях, определяется что демодулятор принимает мягкие решения. Последовательность выходов мягкого решения согласованного фильтра называется как последовательность принятого мягкого решения.

Декодирование путем обработки этой принятой последовательности с мягким решением называется декодированием с мягким решением. Поскольку декодер использует дополнительную информацию, содержащуюся в неквантованных (или многоуровневых квантованных) принятых выборках, для восстановления переданного кодового слова, декодирование с мягким решением обеспечивает лучшую производительность по ошибкам, чем декодирование с жестким решением. В общем случае код декодирования с максимальным правдоподобием (MLD) когда с мягким решением имеет выигрыш в кодировании около 3 дБ по сравнению с алгебраическим декодированием кода ; однако декодирование с мягким решением намного сложнее реализовать, чем алгебраическое декодирование, и требует большей вычислительной сложности. Это цена, которую нужно заплатить за лучшую производительность.

Большое количество алгоритмов декодирования мягкого решения были разработаны. Эти алгоритмы декодирования могут быть классифицированы на две основные категории алгоритм декодирования, основанные на надежности (или вероятности) и алгоритмы декодирования на основе структуры кода.

Разница между жестким и мягким декодером решений заключается в следующем

При жестком декодировании решения полученное кодовое слово сравнивается со всеми возможными кодовыми словами и выбирается кодовое слово, которое дает минимальное расстояние Хэмминга

В мягком декодировании решений полученное кодовое слово сравнивается со всеми возможными кодовыми словами и выбирается кодовое слово, которое дает минимальное евклидово расстояние. Таким образом, мягкое декодирование решений улучшает процесс принятия решений, предоставляя дополнительную информацию о надежности (вычисленное евклидово расстояние или вычисленное логарифмическое отношение правдоподобия)

Для того же кодировщика и комбинации каналов позволяет увидеть эффект замены жесткого блока решений на мягкий блок решений[9].

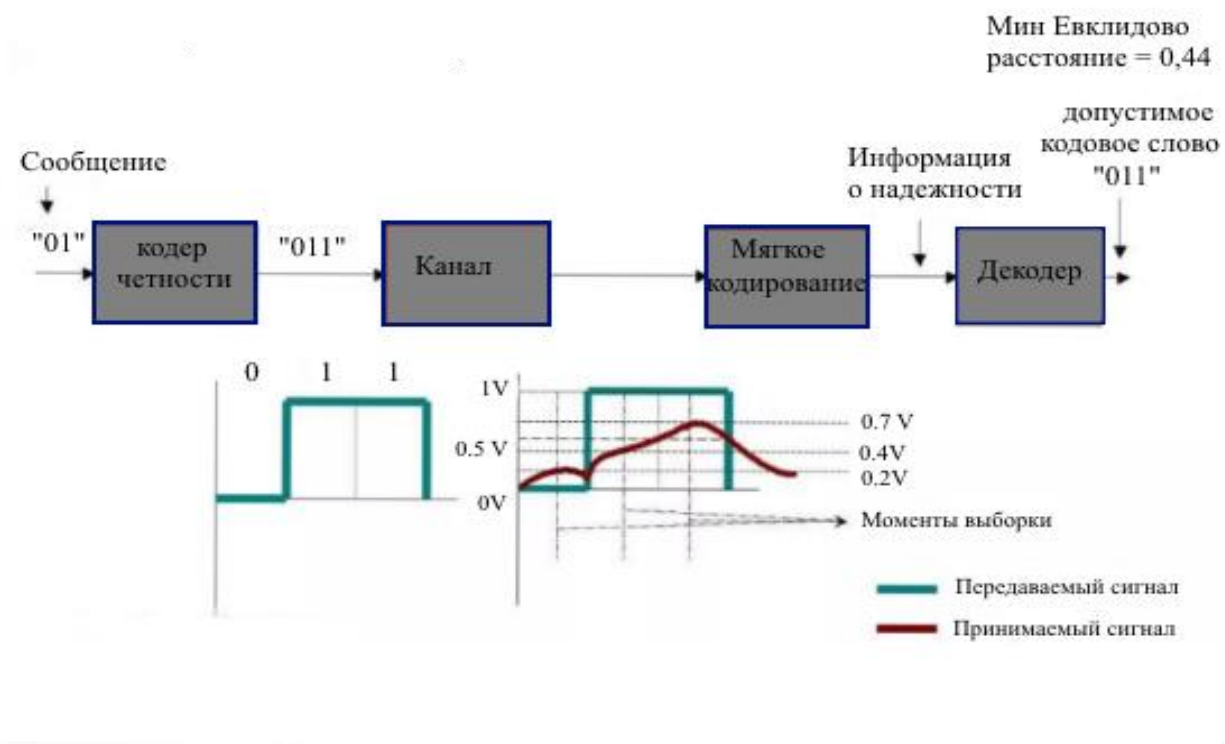

Рисунок 2.1 Структурная схема декодирования с мягким решением

Уровни напряжения принимаемого сигнала в каждый момент дискретизации показаны на рисунке. Блок мягкого решения вычисляет Евклидово расстояние между принимаемым сигналом и всеми возможными кодовыми словами.

Можно сделать расчеты Евклидового расстояния.
Даны допустимые кодовые слова :

1. 0 0 0 (0V 0V 0V)
2. 0 1 1 (0V 1V 1V)
3. 1 0 1 (1V 0V 1V)
4. 1 1 0 (1V 1V 0V)

При этом данным кодовым словам соответствуют уровни напряжения в каждый момент дискретизации принимаемого сигнала :

1. 0 0 0 (0V 0V 0V) - 0.2V 0.4V 0.7V
2. 0 1 1 (0V 1V 1V) - 0.2V 0.4V 0.7V
3. 1 0 1 (1V 0V 1V) - 0.2V 0.4V 0.7V
4. 1 1 0 (1V 1V 0V) - 0.2V 0.4V 0.7V

Необходимо будет найти Евклидово расстояние

1. $(0-0.2)^2+ (0-0.4)^2+ (0-0.7)^2$
2. $(0-0.2)^2+ (1-0.4)^2+ (1-0.7)^2$
3. $(1-0.2)^2+ (0-0.4)^2+ (1-0.7)^2$
4. $(1-0.2)^2+ (1-0.4)^2+ (0-0.7)^2$

В конечном итоге получим значения

1. $(0-0.2)^2+ (0-0.4)^2+ (0-0.7)^2 = 0.69$
2. $(0-0.2)^2+ (1-0.4)^2+ (1-0.7)^2 = 0.49$
3. $(1-0.2)^2+ (0-0.4)^2+ (1-0.7)^2= 0.89$
4. $(1-0.2)^2+ (1-0.4)^2+ (0-0.7)^2= 1.49$

Минимальное евклидово расстояние - "0.49“, соответствующее кодовому слову” 0 1 1" (которое мы передали). Декодер выбирает это кодовое слово в качестве выходного. Даже если кодировщик четности не может исправить ошибки, схема мягкого решения помогла в восстановлении данных в этом случае. Этот факт очерчивает улучшение, которое будет видно, когда эта схема мягкого решения используется в сочетании со схемами прямого исправления ошибок (FEC), такими как сверточные коды, LDPC и т. Д

Из этой иллюстрации мы можем понять, что декодеры мягких решений используют всю информацию (уровни напряжения в этом случае) в процессе принятия решений, тогда как декодеры жестких решений не полностью используют информацию, доступную в полученном сигнале (очевидно, вычисляя расстояние Хэмминга, просто сравнивая уровень сигнала с порогом, при котором пренебрегают фактическими уровнями напряжения).

Примечание: это просто, чтобы проиллюстрировать концепцию мягкого решения и жесткого декодирования решения. Благоразумные души будут достаточно быстры , чтобы обнаружить, что пример кода четности потерпит неудачу для других уровней напряжения (например : 0.2 V, 0.4 V и 0.6 V) . Это происходит потому, что кодеры четности не способны исправлять ошибки, но способны обнаруживать одиночные битовые ошибки. Использование схемы декодирования мягкого решения улучшит производительность приемника примерно на 2 дБ по сравнению со схемой жесткого решения.

Мягкая схема декодирования решений часто реализуется с помощью декодеров Витерби . Такие декодеры используют алгоритм Soft Output Viterbi (SOVA - Алгоритм Витерби Мягкого Выхода), который учитывает априорные вероятности входных символов, производящих мягкий выход, указывающий на надежность решения. [13]

2.1 Основные математические операции кода Рида-Мюллера

Код являются Рида-Мюллера одними из старейших и наиболее известных кодов. Они были открыты и предложены Д.Э.Мюллером и И.С.Ридом в 1954 году. Коды Рида- Мюллера- одни из самых старых кодов для исправления ошибок. Коды исправления ошибок очень полезны при отправке информации на большие расстояния или через каналы, где в сообщении могут возникнуть ошибки. Они стали более распространенными, поскольку в телекоммуникации расширили и развили использование кодов, которые могут само заправляться.

Коды Рида-Мюллера имеют много интересных свойств , это стоит изучить; они образуют бесконечное семейство кодов, и большие коды Рида-Мюллера могут быть построены из меньших. К сожалению, коды Рида-Мюллера становятся слабее по мере увеличения их длины. Тем не менее, они часто используются в качестве строительных блоков в других кодах. Одним из главных преимуществ кодов Рида - Мюллера является их относительная простота в кодировании сообщений и декодировании принимаемых передач. Коды Рида-Мюллера, как и многие другие коды, имеют тесные связи с теорией проектирования; мы кратко исследуем эту связь между кодами Рида-Мюллера и проектами, вытекающими из аффинных геометрий.

В системе связи, мы представляем информацию в виде последовательности 0 и 1 (двоичная форма). На рисунке 2 показана структурная схема канала связи . Сообщение обычно не передается в нем полностью, а скорее отправляется по кусочкам в блоках фиксированной длины, называемых словами сообщения .

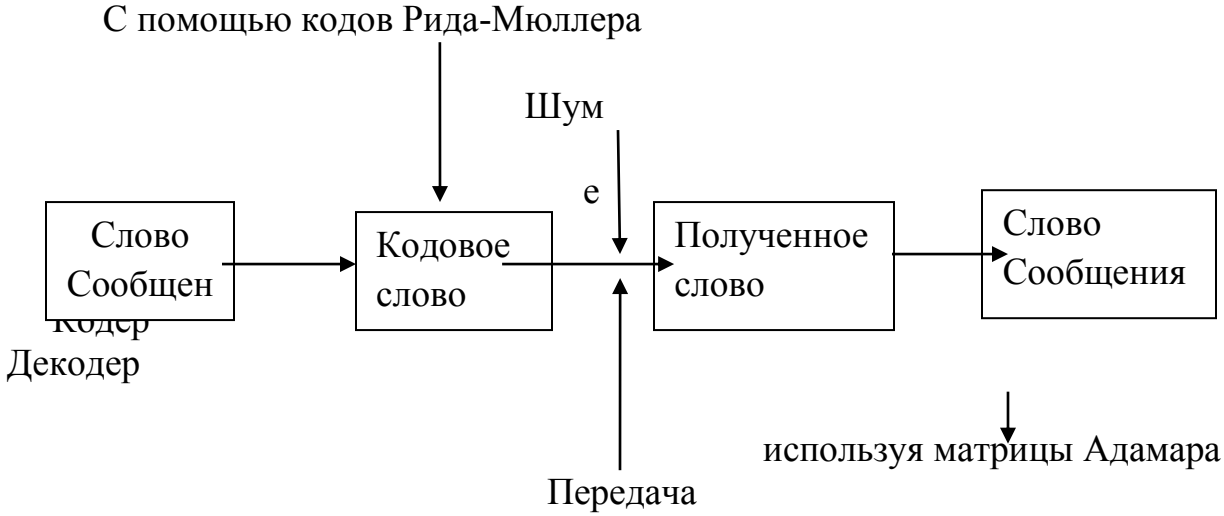

Рисунок 2. 2– Передача кода Рида – Мюллера через канал связи

Каждое слово Сообщения кодируется в кодовое слово с использованием подходящего алгоритма кодирования, а затем отправляется по шумному каналу. Коллекция все кодовые слова называются кодом. При передаче кодовое слово может быть повреждено ошибкой (e) в слово. Затем декодер выполняет несколько заданий. Во-первых, он должен определить, что слово не является кодовым словом, определить ошибку (e), которая произошла, и исправить ее соответствующим образом, чтобы получить исходное кодовое слово. Его можно легко расшифровать для того чтобы получить первоначально слово Сообщения как выход для приемника.[22]

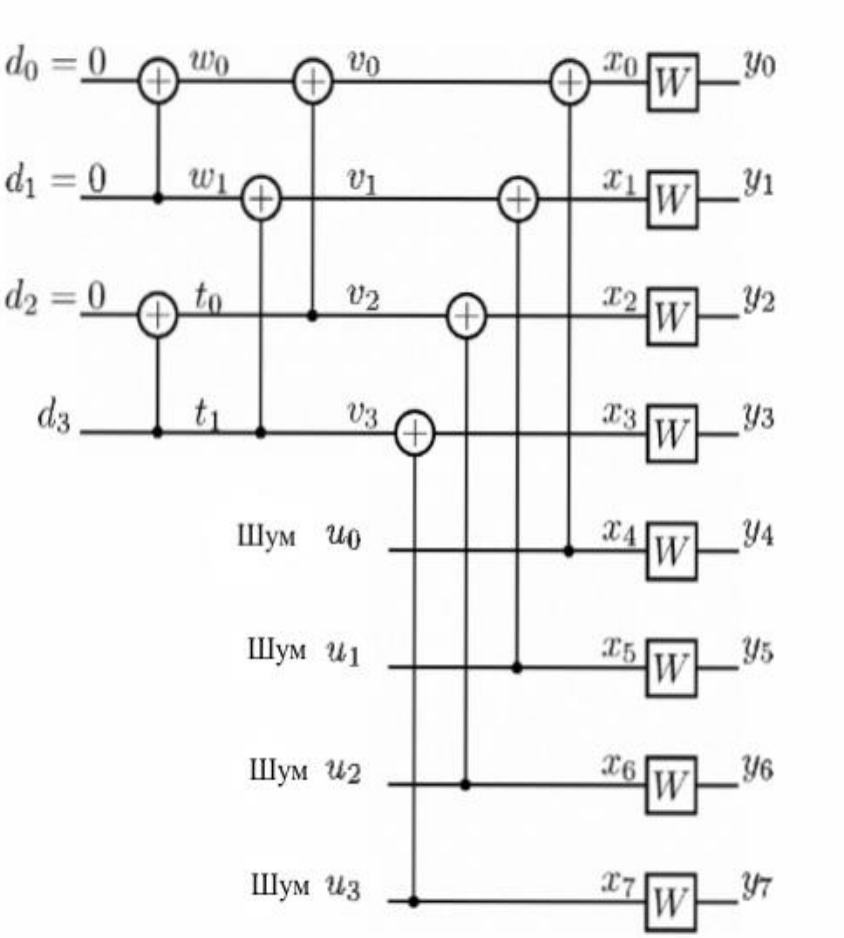

Рисунок 2.3

С битами данных u_o, u_l, u_2 и u_4, равными нулю. Учитывая выход канала y_0- y_7, рекурсивный алгоритм декодирования разбивает задачу декодирования для X E RM(l, 3) на первую задачу декодирования для V E RM(O,2) и вторую для U E RM (1, 2) .

При декодировании для V часть U рассматривается как чистый двоичный шум с независимыми компонентами Бернулли.

Определение терминов и операции. Векторные пространства, используемые в этой работе, состоят из строк длиной 2^m, где m-положительное целое число, чисел в

$F_2 = \{0, 1\}$. Кодовые слова кода Рида Мюллера образуют подпространство такого пространства. Векторы можно манипулировать с помощью трех основных операций: сложения, умножения и скалярного произведения. Для двух векторов

$$x = (x_1, x_2, \ldots, x_n) \text{ и } y = (y_1, y_2, \ldots, y_n),$$

сложение определяется как :

$$x + y = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n), \tag{2.2}$$

где каждый x_i или y_i либо 1 или 0, и

$$1 + 1 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 0 + 0 = 0.$$

Например, если x и y определены как

$$x = (10011110) \text{ и } y = (11100001),$$

то сумма x и y равна

$$x + y = (10011110) + (11100001) = (01111111).$$

Добавление скаляра $a \in F_2$ к вектору x определяется как :

$$a + x = (a + x_1, a + x_2, \ldots, a + x_n). \tag{2.3}$$

Дополнением $\ddot{X}$ вектора x является вектор, равный 1+x. Примером добавления константы к вектору является

$$1 + (000111) = (111000).$$

Умножение определяется по формуле

$$x * y = (x_1 * y_1, x_2 * y_2, \ldots, x_n * y_n), \tag{2.4}$$

где каждый x_i или y_i либо 1 или 0, и

$$1 * 1 = 1, \quad 0 * 1 = 0, \quad 1 * 0 = 0, \quad 0 * 0 = 0.$$

Например, используя те же x и y выше, произведение x и y

$$x * y = (10011110) * (11100001) = (10000000).$$

Умножение константы a ∈ F_2 на вектор x определяется

$$a * x = (a * x_1, a * x_2, \ldots, a * x_n). \quad (2.5)$$

Пример Нижнего $0 * (111001) = (000000)$. Скалярное произведение x и y определяется

$$x * y = x_1 * y_1 + x_2 * y_2 + \cdot * * + x_n * y_n$$

Например, используя x и y сверху,

$$x \cdot y = (10011110) \cdot (11100001) = 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1.$$

Все три эти операции требуют векторов с одинаковым количеством координат. Векторы могут быть связаны с логическими многочленами. Логический многочлен является линейной комбинацией логических одночленов с коэффициентами в F2. Логическое мономиальное p в переменных $x_1, \ldots X_м$, это выражение вида,

$$p = x_1^{r1} \cdot x_2^{r2} \cdot \cdot x_m^{rm} \quad \text{где} \quad r_i \in \{0, 1, 2, \ldots\} \text{ and } 1 \leq i \leq m \quad (2.6)$$

Сокращенная форма p' , p получается в результате применения правил,

$$x_i x_j = x_j x_i \quad \text{и} \quad x_i^2 = x_i , \quad (2.7)$$

Пока факторы не будут различны. Степень p-это обычная степень p', которая является числом переменных в p'. Логический многочлен в приведенный форме, если каждый одночлены в сокращенной форме. Степень Логического многочлена q-обычная степень его приведенной формы q'

Примером Булева многочлена в приведенной форме со степенью три является

$$q = x_1 + x_2 + x_1x_2 + x_2x_3x_4. \quad (2.8)$$

Теперь мы можем описать, как связать логическое одночлены в m переменных в вектор с 2^m записей. Степень-ноль одночлены - это 1, а степень - один одночленов являются $x_1, x_2, \ldots x_м$. Сначала мы определяем векторы, связанные с этими мономами. Вектор связанный с мономиальным 1 является просто вектором длины 2^m, где каждая запись вектора равна 1. Итак, в пространстве размера 2^3 вектор, связанный с 1, равен (11111111).

Вектор, связанный с мономиальным x_1, равен 2^{m-1}, за которым следуют 2^{m-1} нули. Вектор, связанный с мономиальным x_2, равен 2^{m-2}, за ним следуют 2^{m-2} нуля, затем еще 2^{m-2} единицы, а затем еще 2^{m-2} нуля. В общем случае вектор, связанный с мономиальным x_i, представляет собой шаблон из 2 средних, за которым следуют 2^{m-i} нули, повторяемые до тех пор, пока не будут определены значения 2^m. Например , в пространстве размером 2^4 вектор, связанный с x_4, равен (1010101010101010).

Чтобы сформировать вектор для мономиального x_1^{r1} ,· x_2^{r2} ,…, сначала поставят одночлены в сокращенной форме. Затем умножьте векторы, связанные с каждым мономиальным x_i в приведенной форме. Например, в пространстве с m = 3 вектор, связанный с мономиальным $x_1\ x_2\ x_3$, можно найти путем умножения

$$(11110000) * (11001100) * (10101010),$$

что дает (10000000).

Сформировать вектор полинома, просто уменьшить все одночленов на многочлен и найти векторы, связанные с каждым из мономов. Затем, добавить все направления, связанные с каждым из этих мономов вместе, чтобы сформировать вектор связан с полиномом. Это дает нам биекцию между приведенными многочленами и векторами. Отныне мы будем рассматривать уменьшенный многочлен и вектор, связанный с этим многочленом, взаимозаменяемо [4].

3 Расчет кодирования и декодирования RM кода

3.1 Кодирование кода Рида Мюллера

Код Рида Мюллера r-го порядка R (r, m) - множество всех двоичных строк (векторов) длины $n = 2^m$, связанных с логическими многочленами $p(x_1, x_2, \ldots, x_m)$ степени не более r. Код Рида Мюллера нулевого порядка R (0, m) состоит из двоичных строк, связанных с постоянными многочленами 0 и 1; то есть,

$$R(0, m) = \{0,1\} = Rep(2^m) \tag{3.1}$$

Таким образом, R (0, m) - это просто повторение нулей или единиц длины 2^m. С другой стороны, код Рида Мюллера M-го порядка R (m, m) состоит из всех двоичных строк длиной 2^m

Чтобы определить матрицу кодирования R (r, m), пусть первая строка матрицы кодирования равна 1, длина вектора 2^m со всеми записями равна 1. Если r равно 0, то эта строка является единственной в матрице кодирования. Если r равно 1, то добавьте m строк, соответствующих векторам $x_1, x_2, \ldots, x_m$ в матрицу кодирования R (0, m).

Сформировать матрицу кодирования R (r, m), где r больше 1, и (m,r) строк в матрицу кодирования R (r−1, m). Эти добавленные строки состоят из всех возможных уменьшенных мономов степени r, которые могут быть сформированы с помощью строк $x_1, x_2, \ldots x_m$. Например, при m = 3 имеем

$$\begin{bmatrix} 1 & 1\,1\,1\,1\,1\,1\,1\,1 \\ x1 & 1\,1\,1\,1\,0\,0\,0\,0 \\ x2 & 1\,1\,0\,0\,1\,1\,0\,0 \\ x3 & 1\,0\,1\,0\,1\,0\,1\,0 \end{bmatrix} \quad R(1, 3)$$

Строки $x_1 x_2$ = 11000000, $x_1 x_3$ = 10100000 и $x_2 x_3$ = 10001000 добавляются в форму

$$\begin{bmatrix} 1 & 1\,1\,1\,1\,1\,1\,1\,1 \\ x1 & 1\,1\,1\,1\,0\,0\,0\,0 \\ x2 & 1\,1\,0\,0\,1\,1\,0\,0 \\ x3 & 1\,0\,1\,0\,1\,0\,1\,0 \\ x1x2 & 1\,1\,0\,0\,0\,0\,0\,0 \\ x2x3 & 1\,0\,0\,0\,1\,0\,0\,0 \end{bmatrix} \quad R(2, 3)$$

Наконец, строка $x_1 x_2 x_3$ = 10000000 добавляется в форму

$$\begin{bmatrix} 1 & 1\,1\,1\,1\,1\,1\,1\,1 \\ x1 & 1\,1\,1\,1\,0\,0\,0\,0 \\ x2 & 1\,1\,0\,0\,1\,1\,0\,0 \\ x3 & 1\,0\,1\,0\,1\,0\,1\,0 \\ x1x2 & 1\,1\,0\,0\,0\,0\,0\,0 \\ x1x3 & 1\,0\,1\,0\,0\,0\,0\,0 \\ x2x3 & 1\,0\,0\,0\,1\,0\,0\,0 \\ x1x2x3 & 1\,0\,0\,0\,0\,0\,0\,0 \end{bmatrix} R(3, 3)$$

Другим примером матрицы кодирования Рида Мюллера является

$$\begin{bmatrix} 1 & 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ x1 & 1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ x2 & 1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0 \\ x3 & 1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0 \\ x4 & 1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0 \\ x1x2 & 1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ x1x3 & 1\,1\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ x1x4 & 1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ x2x3 & 1\,1\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0 \\ x2x4 & 1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0 \\ x3x4 & 1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0 \end{bmatrix} R(2, 4)$$

Кодирование сообщения с использованием кода Рида Мюллера R (r, m) является простым. Надо взять код, который используется, чтобы быть R (r, m). Ее размерность

$$k = 1 + \left(\frac{m}{1}\right) + \left(\frac{m}{2}\right) + \cdots + \left(\frac{m}{r}\right) \qquad (3.2)$$

Другими словами, матрица кодирования имеет k строк. Мы отправляем сообщения в блоках длиной k. Пусть $m = (m_1, m_2, \ldots m_k)$ будет блокировать, закодированные M_c сообщение

$$M_c = \sum_{i=l}^{k} miRi \qquad (3.3)$$

где R_i-строка матрицы кодирования (r, m).
Например, использование R (1, 3) для кодирования m = (0110) дает

$0 * (11111111) + 1 * (11110000) + 1 * (11001100) + 0 * (10101010) = (00111100).$

Аналогично, использование Ŗ(2, 4) для кодирования

m = (10101110010) дает (0011100100000101).

3.2 Декодирование кода Рида Мюллера

Декодирование зашифрованных сообщений Рида Мюллера сложнее, чем их кодирование. Теория кодирования и декодирования основана на расстоянии между векторами. Расстояние между любыми двумя векторами - это количество мест в двух векторах, которые имеют разные значения. Расстояние между любыми двумя кодовыми словами в коде Ŗ (r, m) равно 2^{m-r} .Основой для кодирования Рида Мюллера является предположение, что самое близкое кодовое слово в Ŗ (r,m) к полученному сообщению – это исходное закодированное сообщение. Таким образом, для исправления ошибок в полученном сообщении расстояние между любыми двумя кодовыми словами в Ŗ (r, m) должно быть больше 2e.

Используемый метод декодирования не очень эффективен, но прост в реализации. Он проверяет каждую строку матрицы кодирования и использует логику большинства, чтобы определить, использовалась ли эта строка при формировании сообщения кодирования. Таким образом, можно определить, что такое кодированное сообщение без ошибок и что такое исходное сообщение.

Этот метод декодирования задается следующим алгоритмом: примените шаги 1 и 2 ниже, к каждой строке матрицы, начиная снизу и работая вверх.

Шаг 1. Выберите строку в матрице кодирования Ŗ (r, m). Найдите характеристические векторы 2^{m-r} (этот процесс описан ниже) для этой строки, а затем возьмите точечное произведение каждой из этих строк с закодированным сообщением.

Шаг 2. Возьмите большинство значений точечных продуктов и назначьте это значение коэффициенту строки.

Шаг 3. После выполнения шагов 1 и 2 для каждой строки, кроме верхней строки снизу матрицы вверх, умножьте каждый коэффициент на соответствующую строку и добавьте результирующие векторы, чтобы сформировать M_y. Добавьте этот результат в полученное закодированное сообщение. Если результирующий вектор имеет больше единиц, чем нулей, то коэффициент верхней строки равна 1, иначе 0. Добавление верхней строки, умноженной на ее коэффициент, к M_y дает исходное закодированное сообщение. Таким образом, мы можем выявить ошибки. Вектор, образованный последовательностью коэффициентов, начиная с верхней строки матрицы кодирования и заканчивая нижней строке исходного сообщения.

Чтобы найти характеристические векторы любой строки матрицы, возьмем мономиальное r, связанное со строкой матрицы кодирования. Затем возьмем E как

множество всех x_i, которые не находятся в мономиальном r, но находятся в матрице кодирования. Характеристика векторов являются векторы, соответствующие Мономах в x_i и $\bar{x}_i$, такие, что только один из x_i или $\bar{x}_i$ в каждом одночлены для всех x_i в E . Например, в последней строке кодировка матрицы Ŗ (2, 4) связана с x_3x_4, поэтому характеристических векторов соответствуют следующие комбинации x_1, x_2, $\bar{x}_1$, и x_2 : x_1x_2, $x1\bar{x}_2$, $\bar{x}_1x_2$, $\bar{x}_1\bar{x}_2$. Эти характеристические векторы имеют свойство, что точечное произведение равно нулю со всеми строками в Ŗ (r, m), кроме строки, которой соответствуют характеристические векторы.

Если исходное сообщение m = (0110) с использованием Ŗ (1, 3), то закодированное сообщение M_c = (00111100). Поскольку расстояние в Ŗ (1, 3) равно $2^{3-1} = 4$, этот код может исправить одну ошибку. Пусть закодированное сообщение после ошибки будет M_e = (10111100). Характерными векторами последней строки x_3 = (10101010) являются x_1 x_2, $x_1\bar{x}_2$, $\bar{x}_1x_2$ и $\bar{x}_1\bar{x}_2$.

Вектор, связанный с x_1, равен (11110000), поэтому $\bar{x}_1$ = (00001111). Вектор, связанный с x_2, равен (11001100), поэтому $\bar{x}_2$ = (00110011). Поэтому мы имеем x_1 x_2 = (11000000), x_1 $\bar{x}_2$ = (00110000), $\bar{x}_1x_2$= (00001100), и $\bar{x}_1\bar{x}_2$ = (00000011). Принимая точечные продукты этих векторов с M_e , мы получаем

(11000000) · (10111100) = 1, (00110000) · (10111100) = 0,
(00001100) · (10111100) = 0, (00000011) · (10111100) = 0.

Можно сделать вывод, что коэффициент x_3 равен 0.

Делая то же самое для предпоследней строки матрицы x2 = (11001100), получаем характеристические векторы x1 x3, $\bar{x}_1$x3, x1$\bar{x}_3$ и $\bar{x}_1\bar{x}_3$. Эти векторы (10100000), (01010000), (00001010), и (00000101), соответственно. Принимая точечные продукты этих векторов с M_e, мы получаем

(10100000) · (10111100) = 0, (01010000) · (10111100) = 1,
(00001010) · (10111100) = 1, (00000101) · (10111100) = 1.

Итак, можно сделать вывод, что коэффициент x_2 равен 1. Делая то же самое для второй строки матрицы x_1 = (11110000), получаем

(10001000) · (10111100) = 0, (00100010) · (10111100) = 1,
(01000100) · (10111100) = 1, (00010001) · (10111100) = 1.

Можно сделать вывод, что коэффициент для x_1 также равен 1.

Если мы добавим 0 = (10101010) и 1 = (1100) и 1 = (11110000), мы получим M_y, который равен (00111100). Тогда мы видим, что сумма M_y и M_e равна

(00111100) + (10111100) = (10000000).

Это сообщение имеет больше нулей, чем единиц, поэтому коэффициент первой строки матрицы кодирования равна нулю. Таким образом, мы можем сложить коэффициенты для четырех строк матрицы, 0,1,1 и 0, и увидеть, что исходное сообщение было (0110). Мы также видим, что ошибка была в первую очередь безошибочным сообщением M_c = (00111100).

ЗАКЛЮЧЕНИЕ

Линейные коды играют большую роль в блоковом коде , который применяется в схемах нахождения и исправления ошибок. С помощью линейных кодов можно реализовывать наиболее эффективные алгоритмы кодирования и декодирования информации.

В первой главе данной работы был воспроизведен обзор о помехоустойчивом коде. Перечислены наиболее известные всем коды.

Проведен расчет нахождения матрицы проверки на четность, матрицы генератора и всех 16 кодовых слов для кода (7, 4) Хэмминга. Определен синдром, при полученном кодовом слове а) 0001111 и б) 0111111.

Далее определены алгоритмы линейного блочного кодирования с мягким решением. Предоставлена информация о декодировании с жестким и мягким решением, приведена структура и расчет Хэммингового расстояния для декодирования с жестким решением и Евклидового расстояния для декодирования с мягким решением.

Обсуждены основные математические операции кода Рида-Мюллера. Также были произведены расчеты кодирования и декодирования кодов Рида – Мюллера. В начале сформирована матрица кодирования R (r, m) . Применен код который использовался R (1, 3) и определена его размерность. Исходя из матрицы кодирования было проведено декодирование пошагово.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Shu Lin , Daniel J ,Costello , Jr - Error Control Coding (1ed) «PEARSON».
2 Б.Д.Кудряшов - Основы теории кодирования «БХВ-Петербург»
3 А.М.Голиков - Модуляция, кодирование и моделирование в телекоммуникационных системах «ЭБС ЛАНЬ»
4 В. А. Варгаузин , И. А. Цикин - Методы повышения энергетической и спектральной эффективности цифровой радиосвязи «БХВ-Петербург»
5 Радиорелейные и спутниковые системы передачи. Под ред. А.С. Немировского.- М.: Радио и связь, 1985- 215 с.
6 https://www.sciencedirect.com/topics/engineering/linear-block-code
7 https://en.wikipedia.org/wiki/Reed%E2%80%93Muller_code#cite_ref-1
8 Горелик Р.А., Голубицкая Е.А. Основы экономики предприятий телекоммуникаций.- М.: Радио и связь, 1997- 324 с.
9 Журавлев В.В. Системы беспроводных технологий // Дискавери».- 1995.- № 8.- с. 39-45.
10 Бакеев Д.Р. Нашествие беспроводных технологий // Информ курьер связь. 2002.- № 11.- с. 15-18.
11 Денисьева О.М., Мирошников Д.Г. Средства связи для последней мили.- М.: Эко-Трендз, 2000- 350 с.
12 Shu Lin; Daniel Costello (2005). Error Control Coding (2 ed.). Pearson. ISBN 978-0-13-017973-9. глава 4.
13 J.H. van Lint (1992). Introduction to Coding Theory. GTM. 86 (2 ed.). Springer-Verlag. ISBN 978-3-540-54894-2. глава 4.5.
14 http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.208.440&rep=rep1&type=pdf
15 Reed, Irving S., A Class of Multiple-Error-Correcting Codes and Decoding Scheme,_ MIT Lincoln Laboratory, 1953.
16 https://www.sciencedirect.com/science/article/pii/S0924650997800128?via%3Dihub
17 Douglas B. West - Discrete mathematics , том 340, выпуск 4, апрель 2017, 722-728 с
18 https://www.researchgate.net/publication/299837333_Decoding_Reed-_Muller_Codes_by_Using_Hadamard_Matrices
19 https://www.gaussianwaves.com/2009/12/hard-and-soft-decision-decoding-2/
20 http://www.ezop.ru/ Электроэнергетика, защита от помех
21 .http://www.mis.ru/ Компания МИС- информаналитика в области телекоммуникаций.

22
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.208.440&rep=rep1&type=pdf
23 Arazi, Benjamin, A Commonsense Approach to the Theory of Error Correcting Codes,_ MIT Press, 1988.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
СӘТБАЕВ УНИВЕРСИТЕТІ

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на дипломную работу

Толеген Нурислама Ергалиулы

5В071900 – Радиотехника, электроника и телекоммуникации

Тема: Анализ оптимального декодера для линейных кодов

В данной дипломной работе рассматриваются линейные коды Рида – Мюллера их кодирование и декодирование с мягким решением.

В начале работы проводиться краткий аналитический обзор помехоустойчивых кодов, применение и его классификация.

Далее определяются алгоритмы кодирования основанных на линейных кодах с мягким решением. Описывается более подробно декодирование с мягким решением, учитываются достоинства и недостатки декодирования с мягким решением.

Проводятся расчеты кодирования кода и декодирования кода Рида Миллера.

Считаю, что дипломная работа выполнена на 70/С/«удовлетворительно», а дипломант, Толеген Нурислам Ергалиулы, заслуживает присвоения академической степени бакалавра техники и технологии по специальности 5В071900-Радиотехника, электроника и телекоммуникации.

Научный руководитель
Ассоциированный профессор ЭТиКТ ,
канд.техн.наук
_____ Л.Б.Илипбаева
"18" мая 2019г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
СӘТБАЕВ УНИВЕРСИТЕТІ

РЕЦЕНЗИЯ

на дипломную работу

Толеген Нурислам Ергалиулы

5В071900 – Радиотехника, электроника и телекоммуникации

На тему: Анализ оптимального декодера для линейных кодов

Выполнено:
а) графическая часть на 15 листах
б) пояснительная записка на 41 страницах

ЗАМЕЧАНИЯ К РАБОТЕ

В данной дипломной работе рассматриваются линейные коды Рида – Мюллера их кодирование и декодирование с мягким решением.

В начале работы проводиться краткий аналитический обзор помехоустойчивых кодов, применение и его классификация.

Далее определяются алгоритмы кодирования основанных на линейных кодах с мягким решением. Описывается более подробно декодирование с мягким решением, учитываются достоинства и недостатки декодирования с мягким решением.

Проводятся расчеты кодирования кода и декодирования кода Рида Миллера.

Оценка работы

Считаю, что дипломная работа выполнена на 70/С/«удовлетворительно», а дипломант, Толеген Нурислам Ергалиулы, заслуживает присвоения академической степени бакалавра техники и технологии по специальности 5В071900-Радиотехника, электроника и телекоммуникации.

Рецензент
канд.техн. наук, доцент АУЭС
_____А.О.Касимов
"15" 05 2019г.

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Төлеген Нұрислам Ергалиұлы

Название: Анализ оптимального декодера для линейных кодов

Координатор: Ляззат Илипбаева

Коэффициент подобия 1:2,7

Коэффициент подобия 2:0,6

Тревога:1

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

.....
.....
.....
.....
.....

11.05.2019 .....

Дата

Подпись заведующего кафедрой /

начальника структурного подразделения