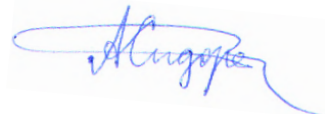


**Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Владимирский государственный университет имени Александра  
Григорьевича и Николая Григорьевича Столетовых»**

На правах рукописи



**СИДОРЕНКО Александр Анатольевич**

**ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОМБИНИРОВАННЫХ  
ПОМЕХОУСТОЙЧИВЫХ КОДОВ**

Специальность: 2.2.15  
«Системы, сети и устройства телекоммуникаций»

диссертация на соискание ученой степени  
доктора технических наук

Научный консультант: доктор технических наук, профессор  
Александр Георгиевич Самойлов

Владимир – 2024

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
Глава 1. СОВРЕМЕННЫЕ ПОДХОДЫ К ПОВЫШЕНИЮ ЭФФЕКТИВНОСТИ ПОМЕХОУСТОЙЧИВЫХ КОДОВ .....	21
1.1. Причины ухудшения качества передачи сигнала.....	21
1.2. Основные направления повышение помехоустойчивости .....	24
1.2.1. Борьба с частотно-селективными замираниями .....	24
1.2.2. Борьба с быстрыми замираниями.....	25
1.2.3. Борьба с преднамеренными помехами .....	26
1.2.4. Кодирование с исправлением ошибок .....	27
1.3. Основные характеристики блочных и сверточных помехоустойчивых кодов .....	28
1.4. Противоречие между основными характеристиками систем цифровой связи.....	31
1.5. Теоретические основы оценки эффективности помехоустойчивого кодирования .....	33
1.6. Современные подходы к повышению эффективности помехоустойчивых кодов .....	35
1.6.1. Комбинированные помехоустойчивые коды .....	36
1.6.2. Декодирование мягкого решения детектора .....	37
1.6.3. Перемежение .....	38
1.6.4. Адаптация параметров кода.....	39
1.6.5. Модификация порождающих и проверочных матриц блочного кода ..	41
1.6.6. Перфорация кодовой последовательности .....	42
Выводы по главе 1.....	43
Глава 2. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПОСЛЕДОВАТЕЛЬНЫХ КАСКАДНЫХ КОДОВ, ПОСТРОЕННЫХ НА ОСНОВЕ ВНЕШНЕГО СВЕРТЧНОГО КОДА.....	45
2.1. Корректирующая способность сверточных кодов .....	45

2.2. Разработка программных генераторов массива информации и выхода детектора .....	53
2.3. Сравнение корректирующей способности сверточного кода при декодировании мягкого и жесткого решения детектора .....	64
2.4. Исследование механизма декодирования перфорированных сверточных кодов .....	71
2.5. Исследование корректирующей способности перфорированного сверточного кода .....	77
2.6. Повышение адаптивных возможностей последовательных каскадных кодов, созданных на основе внешнего сверточного кода.....	85
2.7. Повышение корректирующей способности ГПКК при декодировании по максимуму правдоподобия путем применения к значениям декодируемых бит коэффициентов .....	91
Выводы по главе 2.....	97
<b>Глава 3. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПОСЛЕДОВАТЕЛЬНЫХ КАСКАДНЫХ КОДОВ, ПОСТРОЕННЫХ НА ОСНОВЕ ВНЕШНЕГО БЛОЧНОГО КОДА .....</b>	<b>100</b>
3.1. Обзор часто применяемых на практике блочных кодов .....	100
3.1.1. Коды Хемминга .....	101
3.1.2. Коды Голея.....	102
3.1.3. Коды Боуза-Чоудхури-Хоквенгема.....	102
3.2. Сравнение корректирующей способности блочных кодов .....	104
3.2.1. Расчет корректирующей способности кода Хемминга.....	104
3.2.2. Расчет корректирующей способности кода Голея .....	108
3.2.3. Расчет корректирующей способности кода БЧХ.....	108
3.2.4. Расчет корректирующей способности кода Рида-Соломона.....	109
3.3. Реализация перфорации в блочных кодах .....	113
3.4. Последовательное каскадное кодирование на основе внешнего кода Рида-Соломона.....	124
3.5. Мягкое декодирование кода Рида-Соломона.....	128

3.6. Мягкое декодирование блочных кодов с использованием синдромной решетки .....	133
3.7. Декодирование мягкого решения перфорированного кода Хемминга ..	139
3.8. Повышение адаптивных возможностей последовательного каскадного кода на основе внешнего блочного кода .....	142
3.9. Повышение корректирующей способности ГПКК при декодировании по максимуму правдоподобия с применением коэффициентов к значениям декодируемых бит .....	145
Выводы по главе 3.....	149
<b>Глава 4. ДЕКОДИРОВАНИЕ БЛОЧНЫХ И СВЕРТОЧНЫХ КОДОВ С ПРИМЕНЕНИЕМ ДЕКОДЕРА ТУРБОКОДА.....</b>	<b>152</b>
4.1. Вычисление надежности информационных символов в перфорированных сверточных кодах в соответствии с алгоритмом SOVA .....	152
4.2. Оценка корректирующей способности перфорированного турбокода при применении алгоритма декодирования MAP.....	161
4.2.1. Оценка целесообразности введения перфорации в разные части кодового слова турбокода .....	161
4.2.2. Оценка корректирующей способности турбокода при введении перфорации .....	185
4.3. Исследование возможности декодирования блочного и сверточного кода с использованием декодера турбокода .....	190
4.4. Повышение корректирующей способности КПИБ путем снижения чувствительности декодера к достоверности значений проверочных бит кодового слова.....	198
Выводы по главе 4.....	204
<b>Глава 5. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ТУРБОКОДОВ.....</b>	<b>206</b>
5.1. Построение ТСБК при декодировании в соответствии с критерием максимума апостериорной вероятности .....	206



5.2. Сравнение корректирующей способности турбокодов, созданных на основе сверточных и блочных кодов при декодировании в соответствии с критерием максимума апостериорной вероятности.....	217
5.2.1. Сравнение результатов декодирования кодовых слов турбокода, созданного на основе сверточного и блочного кода .....	217
5.2.2. Сравнение корректирующей способности турбокодов, созданных на основе сверточного и блочного кода .....	222
5.3. Повышение корректирующей способности турбокодов путем снижения чувствительности декодера к достоверности значений информационных бит кодового слова.....	229
5.4. Построение комбинированного кода на основе турбокода с дополнительным кодированием информационных бит.....	237
5.5. Повышение корректирующей способности комбинированного кода на основе турбокода с дополнительным кодированием информационных бит	245
Выводы по главе 5.....	252
ЗАКЛЮЧЕНИЕ .....	255
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ .....	261
СПИСОК ЛИТЕРАТУРЫ.....	262
Приложение 1. Акты использования результатов диссертационного исследования .....	281

## ВВЕДЕНИЕ

Информация играет все более возрастающую роль во всех видах человеческой деятельности. За последнее время резко повысились требования, предъявляемые к системам передачи информации. Необходимо вести передачу все больших объемов информации на все большие расстояния со все большей скоростью. При этом энергетические ресурсы передатчика, как правило, ограничены. Растут требования и к достоверности передачи данных.

С каждым днем XXI век все сильнее оправдывает свое название «Информационного века». Один из основателей современной теории информации, Норберт Виннер, считал, что информация является новым элементом к материи и энергии. Наш век – век достижений теории информации – науки, которая изучает количественные закономерности, связанные с получением, передачей, обработкой и хранением информации [1, 2, 3]. Сама же теория информации зародилась в XX веке. С тех пор теория информации прошла большой путь, обогатилась огромным числом интересных научных открытий и доказала свою практическую важность. Прародителями теории информации считаются, прежде всего, Г. Найквист и Р. Хартли [4]. Ральф Вinton Лайон Хартли ввел понятие «информации» (энтропии) как случайной переменной и был первым, кто попытался определить «меру информации». Ранние работы Гарри Теодора Найквиста по определению ширины частотного диапазона, требуемого для передачи информации, опубликованные в статье «Certain factors affecting telegraph speed», заложили основы для последующих успехов в разработке теории информации.

Отцом-основателем теории информации общепризнан Клод Элвуд Шеннон, а годом ее рождения считается 1948 год – когда вышла в свет публикация «Математическая теория связи» [5]. Клод Шеннон ввел основные понятия этой науки: меру количества информации, единицу информации, пропускную способность канала связи.

Первоначальные результаты К. Шеннона были расширены многими учеными,

и, в первую очередь, нашими соотечественниками В.А. Котельниковым, А.Я. Хинчиным, А.Н. Колмогоровым.

К. Шеннон доказал теорему о пропускной способности каналов связи в присутствии помех. Названная теоремой Шеннона-Хартли, данная теорема доказывает, что пропускная способность канала с аддитивным белым гауссовым шумом является функцией средней мощности принятого сигнала, средней мощности шума и ширины полосы пропускания. Теорема Шеннона-Хартли связана с компромиссом между мощностью и полосой пропускания, а также определяет другое важное ограничение – предел Шеннона. Предел Шеннона, равный  $-1,6$  дБ, – это минимальное теоретически возможное значение  $E_b/N_0$ , которое (совместно с канальным кодированием) необходимо для получения сколь угодно низкой вероятности возникновения ошибки в канале с аддитивным белым гауссовым шумом.

Получение, передача, обработка и хранение информации являются основой любой информационной и/или телекоммуникационной системы. Любая информация для того, чтобы быть переданной, должна быть соответствующим образом «закодирована», т.е. переведена на язык специальных символов и сигналов. Одной из задач теории информации является отыскание наиболее экономных методов кодирования, позволяющих передать заданную информацию с помощью минимального количества символов. Эта задача решается как при отсутствии, так и при наличии помех в канале передачи данных. Другая задача теории информации – определение необходимой пропускной способности линии связи для передачи всей поступающей информации. Теория информации представляет собой математическую теорию, посвященную измерению количества информации, преобразованию информации, передаче информации по линиям связи, изучению методов построения различных кодов и пр. [2].

Клод Шеннон не только связал достоверность и скорость передачи данных с пропускной способностью канала связи, но и указал путь надежной передачи сообщений за счет введения избыточности при их кодировании, то есть, до передачи в канал или записи на носитель информации. С тех пор зародилась и стала

быстро развиваться теория помехоустойчивого кодирования, обеспечившая развитие надежных систем передачи и обработки информации. В настоящее время помехоустойчивое кодирование находит применение не только в телекоммуникационных системах, но и в хранении информационных данных, в устройствах цифровой звукозаписи, в цифровых блоках радиовещания и телевидении, в системах сотовой связи и других областях [6].

Известно, что, во-первых, из Шенноновской теории информации следовал тот важный вывод, что построение слишком хороших каналов является расточительством; экономически выгоднее использовать кодирование. Во-вторых, из-за того, что основная теорема кодирования К. Шеннона не конструктивна, то есть, она лишь доказывает существование оптимального помехоустойчивого (корректирующего) кода, обеспечивающего предельное согласование сигнала с каналом, и таким образом только обосновывает принципиальную возможность построения помехоустойчивых кодов, обеспечивающих идеальную передачу, но не указывает способ их построения. В итоге теория Шеннона мобилизовала усилия ученых на разработку конкретных кодов [6]. В теории современных телекоммуникационных систем большое внимание уделяется изучению методов кодирования информации. Семейство помехоустойчивых кодов достаточно разнообразно. Коды различаются длиной сообщения, степенью вводимой избыточности, методами кодирования-декодирования, способностью исправлять одиночные и групповые ошибки [7]. Результаты теории кодирования широко используются в технике электрической связи.

В конце 40-х – начале 50-х гг. XX века В.А. Котельников разработал теоретические основы помехоустойчивости радиосвязи, которые изложил в своей фундаментальной работе «Теория потенциальной помехоустойчивости» [8]. В данном труде определены предельные ограничения на чувствительность радиоприемников при наличии шума в канале. Применение помехоустойчивых кодов способствует повышению помехоустойчивости, то есть, способности системы связи противостоять воздействию помех, действующих в канале передачи. В современных условиях проблема помехоустойчивости остается важнейшей

проблемой радиосвязи, в том числе – военной связи.

В последние годы значительное развитие получили методы цифровой обработки и передачи информации в различных телекоммуникационных системах. Одной из наиболее важных функций в работе таких систем является обеспечение надежной защиты данных от помех. Радиоканал является слабым звеном систем передачи данных, поскольку именно в нем передаваемые сигналы подвержены искажениям и затуханию из-за негативного воздействия многочисленных факторов. Помехи и замирания снижают достоверность передачи информации. Повышение достоверности передаваемой по каналу связи информации можно организовать различными способами, например, увеличением мощности передатчика, улучшением чувствительности приемника, увеличением усиления антенн. Реализация приведенных способов обычно требует значительных материальных затрат, а главное, не обеспечивает повышение достоверности передаваемой информации при частотно-селективных замираниях. Повышение помехозащищенности информации достигается различными способами, но многие из них являются неэффективными по тем или иным причинам. Например, повышение мощности передатчика ограничено жесткими требованиями по электромагнитной совместимости источников излучения, а многократное повторение передаваемых блоков приводит к значительному увеличению занятости каналов и соответствующему возрастанию времени обработки информации [6].

В настоящее время задача обеспечения достоверности передачи информации в большинстве случаев решается применением помехоустойчивого кодирования, которое представляет собой класс преобразований сигнала, выполняемых для повышения качества связи. Современные люди окружены устройствами, использующими в своей работе алгоритмы помехоустойчивого кодирования. Технологии помехоустойчивого кодирования стали обязательным элементом систем хранения и передачи данных. Основу современной теории кодирования составляют работы В.А. Котельникова и К.Э. Шеннона. В дальнейшем теория помехоустойчивого кодирования развивалась многими российскими и

зарубежными исследователями, такими, как В.В. Зяблов, Э.Л. Блох, Л.М. Финк, К.Ш. Зигангиров, Л.Е. Назаров, С.И. Егоров, Е.А. Крук, А. Витерби, Дж. Месси, П. Элайс, Р. Галлагер, Д. Форни, А.Э. Нейфах, Дж. Возенкрафт, Е. Берлекэмп, С. Verrou, A. Glavieux, D.J.C. MacKay, Р.М. Фано, Г.Д. Форни, Д.К. Омура, А.Д. Витерби, Э.Р. Берлекэмп, Р.У. Хемминг, Р.Ч. Боуз, Д.К. Рей-Чоудхури, А.М. Хоквингем, М.Д. Голей, И.С. Рид, Г.М. Соломон, К.А. Нордстром, Д.М. Робинсон, Ю.Б. Зубарев, В.В. Золотарев, Г.В. Овечкин, и многими другими. Однако проблема однозначного выбора вида кодирования для конкретного канала передачи информации пока не решена.

Важным результатом теории оптимального кодирования К. Шеннона было то, что принципиально сколь угодно малая вероятность неправильного декодирования может быть достигнута при использовании весьма длинного случайного кода. Однако случайный код большой длины практически не реализуем, и поэтому методы случайного кодирования не являются конструктивными. Действительно, для реализации случайного кодирования, в кодере и декодере необходимо хранить всю совокупность кодовых комбинаций, так как последние никак не связаны друг с другом. При большой длине кодовых комбинаций требуемый объем запоминающего устройства кодера или декодера при использовании случайного кода является показательной функцией длины кода, что исключает возможность практического его использования. В связи с этим, основными задачами теории корректирующих кодов, определившими на много лет вперед пути ее развития, являются изыскания технически реализуемых методов кодирования и декодирования дискретных сообщений, позволяющих строить кодек меньшей сложности, чем при случайном кодировании [9].

Под эффективностью помехоустойчивого кода можно понимать степень приближения системы цифровой связи к пределу К. Шеннона при использовании данного кода. В ряде источников под понятием эффективности помехоустойчивого кода понимают энергетический выигрыш от применения кодирования в децибелах, показывающий величину снижения энергии, необходимой для передачи одного бита цифровых данных при заданной средней вероятности ошибки на бит в случае

использования корректирующего избыточного кода по сравнению с кодом без избыточности [9-12]. При такой трактовке понятия «эффективности помехоустойчивого кодирования» более эффективен тот помехоустойчивый код, который при достижении заданной вероятности битовой ошибки обладает меньшей избыточностью. При повышении эффективности можно уменьшить размеры дорогостоящих антенных устройств, повысить дальность связи, увеличить скорость передачи цифровых данных, снизить необходимую мощность передатчика и улучшить другие характеристики системы передачи информации. Несмотря на широкую распространенность термина, общепринятого и единого подхода к определению понятия «эффективность помехоустойчивого кодирования» нет. Критерии эффективности помехоустойчивого кодирования нуждаются в уточнении.

История корректирующих кодов началась с изобретением кодов Хемминга, которое произошло практически одновременно с появлением теории информации. Чуть позднее были разработаны коды Голя [10]. В пятидесятые годы прошлого века значительные усилия были приложены на попытки построения в явном виде классов кодов, позволяющих получить обещанную К. Шенноном сколь угодно малую вероятность ошибки, но результаты оказались достаточно скромными. В дальнейшем в помехоустойчивых кодах четко выделились два класса кодов: блочные коды и бесконечные коды с древовидной структурой, получившие название сверточных кодов. Построение и декодирование блочных кодов носило алгебраический характер. В кодировании и декодировании сверточных кодов преобладал вероятностный подход. В шестидесятых годах прошлого века были разработаны недвоичные коды Рида-Соломона, способные эффективно противостоять группирующимся ошибкам. Со временем, для улучшения корректирующей способности помехоустойчивые коды начали соединять в разнообразные комбинации. Было выяснено, что даже для варианта соединения двух относительно простых кодов в комбинацию, можно получить код с весьма высокой корректирующей способностью [6, 9]. В 1962 году Р. Галлагер ввел класс кодов, которые получили наименования кодов с низкой плотностью проверок.

Коды Галлагера значительно приблизились к пределу Шеннона для случая канала передачи данных с относительно низкой вероятностью появления ошибочного бита. В 1993 году, как дальнейшее развитие методов комбинированного кодирования [10, 11], были представлены турбокоды, имеющие итеративный алгоритм декодирования. Турбокоды показали лучшие результаты корректирующей способности в каналах связи с относительно высокой вероятностью появления ошибки. В 2008 г. Э. Ариканом были открыты полярные коды, значительно приблизившиеся к пределу Шеннона, однако их реализация на практике вызвала значительные затруднения.

За последние десять лет по направлению повышения помехоустойчивости цифровых систем связи, в том числе с применением помехоустойчивого кодирования, завершены диссертационные исследования: Лихобабина Е.А., Трифонова П.В., Акмалходжаева А.И., Синицина Д.В., Карболина В.А., Тамразяна Г.М., Чилихина М.Ю., Демидова Д.С. [13 – 20] и др.

В настоящее время существует значительное количество вариантов построения и методов декодирования помехоустойчивых кодов, способных исправлять как одиночные, так и групповые ошибки. Вместе с тем характеристики практических реализаций помехоустойчивых кодов значительно отстают от теоретических пределов. Возникают существенные сложности в удовлетворении требований к эффективности кодов с целью достижения постоянно растущих требований к системам передачи и хранения данных.

**Объект исследования** – комбинированные помехоустойчивые коды.

**Предмет исследования** – методы разработки комбинированных помехоустойчивых кодов.

**Цель работы** – повышение эффективности комбинированных помехоустойчивых кодов.

Поставленная цель достигается решением следующих задач:

1. Определить основные критерии эффективности помехоустойчивого кодирования.
2. Выполнить анализ современных подходов к повышению эффективности



помехоустойчивых кодов.

3. Повысить адаптивные возможности последовательных каскадных кодов.

4. Выполнить анализ возможности построения турбокода на основе блочного кода при декодировании в соответствии с критерием максимума апостериорной вероятности.

5. Выполнить анализ возможности создания комбинированных кодов путем реализации декодирования блочных и сверточных кодов с применением декодера турбокода.

6. Выполнить анализ возможности создания комбинированного кода на основе турбокода с дополнительным кодированием информационных бит.

7. Повысить эффективность декодера каскадного кода, использующего для декодирования внешнего кода метод максимального правдоподобия.

8. Повысить эффективность декодера турбокода, использующего метод декодирования в соответствии с критерием максимума апостериорной вероятности.

9. Разработать программные имитаторы работы кодеков комбинированных кодов с целью проведения количественной оценки повышения эффективности.

**Соответствие паспорту специальности.** Результаты исследования соответствуют следующим пунктам паспорта научной специальности 2.2.15 «Системы, сети и устройства телекоммуникаций».

Пункт 1. Разработка и совершенствование методов исследования, моделирования и проектирования сетей, систем и устройств телекоммуникаций.

Пункт 2. Исследование новых технических, технологических и программных решений, позволяющих повысить эффективность развития цифровых сетей, систем и устройств телекоммуникаций.

Пункт 3. Исследование процессов представления, передачи, хранения и отображения аналоговой, цифровой, видео-, аудио-, голографической и мультимедиа информации; разработка и совершенствование соответствующих алгоритмов и процедур.

Пункт 15. Исследование и разработка новых сигналов, а также

соответствующих модемов, кодеков, мультиплексоров и селекторов, обеспечивающих высокую надежность и качество обмена информацией в условиях воздействия внешних и внутренних помех.

**Методы исследования.** В работе использованы положения теории информации и теории помехоустойчивого кодирования, методы теории графов, теории вероятности и математической статистики.

### **Научная новизна.**

В диссертационном исследовании получены следующие основные результаты, содержащие элементы научной новизны:

1. Решена научная задача, отличающаяся от известных тем, что предложен модифицированный метод последовательного кодирования путем реализации избирательного кодирования кодовых бит внутренним кодом. Модификация заключается в кодировании внутренним кодом определенной части кодовой последовательности на выходе кодера внешнего кода. В результате обеспечивается повышение эффективности кодирования путем реализации возможности гибкой адаптации корректирующей способности кода. Таким образом происходит минимизация избыточности кода для достижения требуемого значения вероятности битовой ошибки в декодированном информационном сообщении.

2. Для решения задачи построения турбокода на основе блочного кода при декодировании в соответствии с критерием максимума апостериорной вероятности, предложен модифицированный метод декодирования, позволяющий производить декодирование с использованием нерегулярной синдромной решетки блочного кода. Осуществлен вывод формулы для расчета вероятности ошибочного бита в декодированном информационном сообщении для такого варианта построения турбокода. Для предложенного варианта построения турбокода без изменения алгоритма декодирования стали возможны: гибкая адаптация параметров и декодирование мягкого решения детектора.

3. Для решения научной задачи повышения эффективности блочных и сверточных кодов предложено добавлять к кодовому слову еще один или несколько блоков информационных бит. При этом становится возможным декодирование такого комбинированного кода модифицированным декодером,

реализующим декодирование по максимуму апостериорной информации. Предложена модификация метода декодирования и структуры декодера. Применение предложенной модификации позволило повысить корректирующую способность без изменения кодера.

4. Предложен новый вариант построения комбинированного кода на основе турбокода с дополнительным кодированием информационных бит. Идея создания комбинированного кода основана на необходимости реализации мер по снижению чувствительности декодера турбокода к ошибкам, возникающим в информационных битах кодового слова турбокода путем дополнительного избирательного кодирования. Кроме повышения корректирующей способности кода положительным эффектом стала возможность гибкой адаптации корректирующей способности кода путем изменения числа дополнительно кодированных информационных бит кодового слова турбокода. Произведен расчет энергетического выигрыша от дополнительного кодирования.

5. Впервые предложена модификация методов декодирования по максимуму правдоподобия и максимизации апостериорной информации путем применения к значениям декодируемых символов коэффициентов, пропорциональных их достоверности. Модификация методов декодирования позволила повысить корректирующую способность без роста избыточности кода и без существенного усложнения алгоритма декодирования.

6. Впервые предложена модификация метода декодирования согласно принципу максимизации апостериорной вероятности, путем применения понижающих коэффициентов к значениям символов, достоверность которых оказывает большее влияние на результат декодирования. Модификация метода декодирования позволила повысить корректирующую способность без роста избыточности кода и без существенного усложнения алгоритма декодирования.

7. Разработаны программные имитаторы, моделирующие работу упоминаемых в диссертационном исследовании кодеков при прохождении канала с аддитивным белым гауссовским шумом.

### **Теоретическая значимость работы.**

Теоретическая значимость исследования заключается, в первую очередь, в разработке и исследовании новых предложений по модификации методов и алгоритмов декодирования комбинированных помехоустойчивых кодов, создании новых кодовых конструкций. Ряд предложенных нововведений обеспечивает возможность гибкой адаптации корректирующей способности помехоустойчивого кода. Иные модификации повышают корректирующую способность кода без роста избыточности кода и существенного усложнения алгоритмов декодирования.

**Практическая ценность результатов.** Практическая ценность результатов работы состоит в создании новых кодовых конструкций, научно обоснованных методов и рекомендаций по повышению эффективности систем цифровой связи. Достижимое повышение эффективности помехоустойчивых каскадных кодов может быть конвертировано как в увеличение скорости передачи данных, так и в снижение необходимой мощности передатчика.

### **Реализация результатов работы.**

Проведенные исследования, разработанные методы и полученные результаты использованы:

- ООО «Рубеж-Техно» (г. Краснодар);
- ООО «Аргун-Софт» (г. Санкт-Петербург);
- АО «МВП «СВЕМЕЛ» (г. Москва);
- ООО «СТИЛСОФТ» (г. Ставрополь);
- ФГБОУ ВО «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (г. Владимир).

Все результаты внедрения подтверждены соответствующими актами.

В ходе работы над диссертационным исследованием разработаны многочисленные программные продукты. Оформлено 7 свидетельств о государственной регистрации программы для ЭВМ.

### **Степень достоверности и апробация результатов.**

Достоверность результатов обусловлена корректным применением теории информации, теории помехоустойчивого кодирования, теории вероятности и

математической статистики, а также подтверждением теоретических результатов экспериментально полученными данными.

Основные результаты диссертационного исследования обсуждались на следующих конференциях:

– международная научная конференция «Перспективные технологии в средствах передачи информации – ПТСПИ-2017»;

– международная научная конференция «Перспективные технологии в средствах передачи информации – ПТСПИ-2019»;

– международная научная конференция «Перспективные технологии в средствах передачи информации – ПТСПИ-2021»;

– International Conference on Applied Physics, Information Technologies and Engineering (Apitech - 2019) (Красноярск, 25-27 сентября 2019 г.);

– III International Scientific Conference «MIP: Engineering-III-2021: Modernization, Innovations, Progress: Advanced Technologies in Material Science, Mechanical and Automation Engineering» (Красноярск, 29-30 апреля 2021 г.);

– III International Conference on Applied Physics, Information Technologies and Engineering (Apitech-III-2021) (Красноярск, 24 сентября – 3 октября 2021 г.);

– IV International Conference on Applied Physics, Information Technologies and Engineering (Apitech-IV 2022) on October 6-8, 2022, Bukhara, Uzbekistan (два доклада);

– V International Conference on Applied Physics, Information Technologies and Engineering (Apitech-V-2023) on October 26–28, 2023, Bukhara, Uzbekistan (два доклада).

### **Основные положения, выносимые на защиту**

**1. Модифицированный метод** последовательного кодирования путем реализации избирательного кодирования кодовых бит внутренним кодом, который обеспечивает повышение эффективности кодирования путем реализации возможности гибкой адаптации корректирующей способности кода к изменениям характеристик канала передачи данных. Реализована возможность плавного роста скорости передачи данных до 48 %.

**2. Комбинированный код**, построенный на основе блочного или сверточного кода путем повторной передачи информационных бит кодового слова с декодированием в соответствии с критерием максимума апостериорной вероятности. Предложенный вариант построения позволяет адаптировать характеристики кода к ухудшению состояния канала передачи данных. Для случая построения комбинированного кода на основе сверточного кода, снижение вероятности битовой ошибки в декодированном информационном сообщении составило 27 раз с  $4,0 \cdot 10^{-5}$  до  $1,5 \cdot 10^{-6}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ).

**3. Модифицированный метод** декодирования в соответствии с критерием максимума апостериорной вероятности, который позволяет осуществлять декодирование турбокода построенного на основе систематического блочного кода и без усложнения алгоритма выполнять декодирование мягкого решения детектора и декодирование перфорированного кода. При декодировании жесткого решения детектора изменение кодовой скорости возможно от 0,3 до 0,5, то есть на 67 %. При декодировании мягкого решения детектора, изменение кодовой скорости возможно от 0,3 до 0,6, то есть на 100 %.

Осуществлен **вывод формул** для расчета вероятности битовой ошибки при декодировании согласно максимуму апостериорной вероятности для турбокодов на основе систематического сверточного кода и на основе блочного кода Хемминга.

**4. Комбинированный код** на основе турбокода с дополнительным выборочным кодированием информационных бит. При дополнительном кодировании лишь одного информационного бита зафиксировано снижение вероятности появления ошибочного бита в декодированном информационном сообщении в 2,18 раза при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ . При дополнительном кодировании трех информационных бит, вероятность битовой ошибки в декодированном информационном сообщении снизилась в 212 раз. Энергетический выигрыш от применения дополнительного кодирования информационных бит кодового слова турбокода составил 0,28 дБ – при дополнительном кодировании одного бита, 0,78 дБ – при дополнительном

кодировании двух бит, 0,8 дБ – при дополнительном кодировании трех бит.

**5. Модификация методов** декодирования по максимуму правдоподобия и максимизации апостериорной вероятности путем применения к значениям декодируемых символов коэффициентов, пропорциональных их достоверности. Применение модификации метода для декодера, функционирующего согласно принципу максимального правдоподобия при исследовании корректирующей способности последовательного каскадного кода с избирательным кодированием внутренним кодом, привело к снижению вероятности битовой ошибки в декодированном информационном сообщении в 920 раз – с  $1,1 \cdot 10^{-10}$  до  $1,2 \cdot 10^{-13}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ). Применение модификации метода для декодера, функционирующего согласно принципу максимизации апостериорной вероятности при исследовании корректирующей способности турбокода, при дополнительном избирательном кодировании информационных бит кодового слова, привело к снижению вероятности битовой ошибки в декодированном информационном сообщении в 5,3 раза – с  $4,0 \cdot 10^{-9}$  по  $7,6 \cdot 10^{-10}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ).

**6. Модифицированный метод** декодирования, согласно принципу максимизации апостериорной вероятности, путем применения понижающих коэффициентов к значениям символов, достоверность которых, оказывает большее влияние на результат декодирования.

При применении модифицированного метода декодирования турбокода, построенного на основе систематического сверточного кода, зафиксировано снижение вероятности битовой ошибки в 14 раз – с  $8,5 \cdot 10^{-7}$  до  $6,1 \cdot 10^{-8}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ ).

Использование модифицированного метода декодирования комбинированного кода, полученного из сверточного кода путем повторной передачи информационных бит, привело к снижению вероятности битовой ошибки в декодированном информационном сообщении в 208 раз – с  $2,5 \cdot 10^{-4}$  до  $1,2 \cdot 10^{-6}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ ).

**Публикации.** Основные результаты диссертации изложены в 39 опубликованных работах, в том числе в 16 работах, опубликованных в журналах из перечня ВАК Министерства образования и науки Российской Федерации; 7 работах в изданиях, включенных в международные базы цитирования; 7 результатах интеллектуальной деятельности; 9 работах в других научных изданиях и материалах конференций. 2 работы опубликованы в соавторстве.

**Личный вклад автора.** Все основные результаты получены автором самостоятельно.

**Структура и объем работы.** Диссертация состоит из пяти глав, введения, заключения, списка сокращений, списка литературы и приложения.



# Глава 1. СОВРЕМЕННЫЕ ПОДХОДЫ К ПОВЫШЕНИЮ ЭФФЕКТИВНОСТИ ПОМЕХОУСТОЙЧИВЫХ КОДОВ

В настоящей главе коротко рассмотрены причины ухудшения качества передачи сигналов. Дано описание основных направлений повышения помехоустойчивости телекоммуникационных систем. Рассмотрены противоречия между основными характеристиками систем цифровой связи, для разрешения которых может быть использовано помехоустойчивое кодирование. Определены основные критерии эффективности помехоустойчивого кодирования. Рассмотрены основные современные подходы к повышению эффективности помехоустойчивого кодирования.

## 1.1. Причины ухудшения качества передачи сигнала

Источником помех в идеальном канале является тепловой шум, генерируемый в приемнике [21, 22, 23]. Тепловой шум имеет, как правило, постоянную спектральную плотность мощности по всей полосе сигнала и гауссову функцию плотности вероятности напряжения с нулевым средним. Сигнал в идеальном канале затухает с расстоянием точно так же, как при распространении в идеальном свободном пространстве. Мощность сигнала убывает пропорционально квадрату расстояния. При таком идеальном распространении мощность сигнала весьма предсказуема.

Дополнительными источниками потерь в реальном радиоканале являются естественные и искусственные источники шума и помех, негативное влияние которых часто оказываются более значительными, чем тепловой шум приемника [24, 25, 26].

При радиосвязи распространение сигнала происходит в атмосфере и вблизи земной поверхности. Радиосигнал может передаваться от передатчика к приемнику по множеству путей. Это явление, называемое многолучевым распространением,

может вызывать флуктуации амплитуды, фазы и угла прибытия полученного сигнала, что называется замиранием вследствие многолучевого распространения сигнала. Замирание вызывает случайные флуктуации сигнала.

Для типичного радиоканала полученный сигнал состоит из нескольких дискретных многолучевых компонентов, приводящих к расширению сигнала во времени (или дисперсия сигнала).

При дисперсии сигнала типы ухудшений характеристик, возникающих вследствие замирания, разделены на частотно-селективные и частотно-неселективные. При нестационарном поведении канала типы ухудшения характеристик, возникающих вследствие замирания, разделены на быстрые и медленные.

На рисунке 1.1 [12] в логарифмическом масштабе отражены три основные категории характеристик, выраженных через вероятность битовой ошибки  $p_D$  в зависимости от  $E_b/N_0$ , где  $E_b$  это энергия бита, а  $N_0$  спектральная плотность мощности шума.

Крайняя левая кривая, имеющая экспоненциальную форму, соответствует ожидаемому поведению данной зависимости при использовании любых номинальных схем модуляции в канале с белым аддитивным гауссовым шумом. Можно наблюдать, что даже при относительно небольшом уровне  $E_b/N_0$ , достоверность передачи данных высока.

Средняя кривая демонстрирует ухудшение достоверности передачи, вытекающее из уменьшения  $E_b/N_0$ , что характерно для медленного замирания. Кривая является функцией, обратно пропорциональной  $E_b/N_0$ , так что для значений  $E_b/N_0$ , представляющих практический интерес, характеристики будут плохими.

Верхняя кривая, достигающая непоправимого уровня ошибок, соответствует эффекту сильного ухудшения характеристик, который может проявиться при быстром замирании.

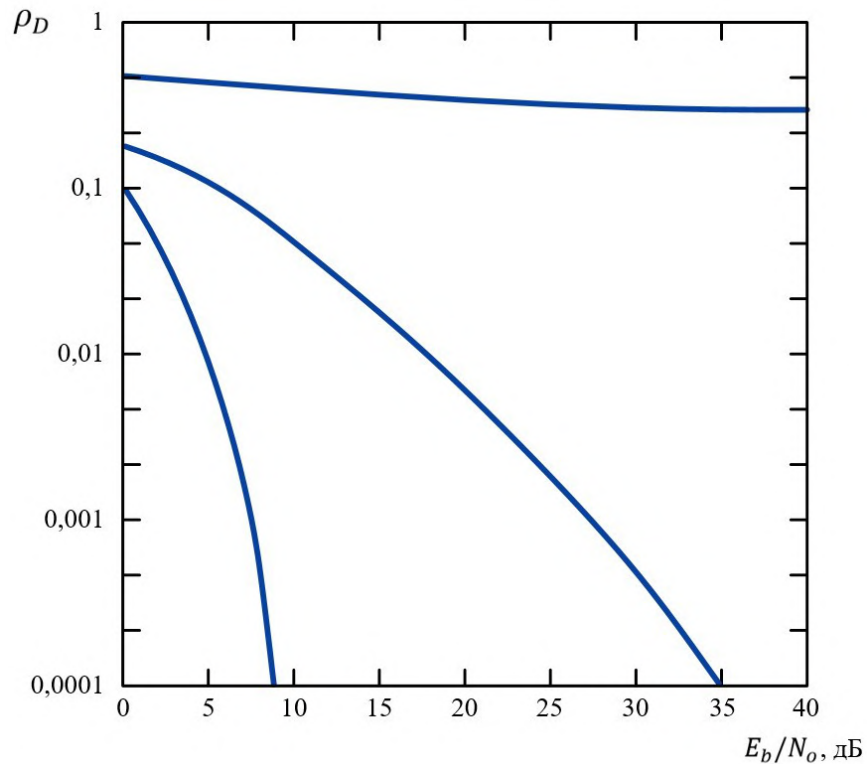


Рисунок 1.1. Достоверность передачи сигналов

Кроме независимых ошибок в канале могут возникать группирующиеся ошибки. Они образуются в каналах с памятью. Одной из основных причин таких ошибок являются перерывы, возникающие из-за плавного снижения уровня сигнала ниже порога чувствительности приемника, когда прием сигнала практически прекращается. Перерывы могут быть различной деятельности, и некоторые из них способны даже вызывать прекращения сеанса связи. Кроме этого причиной перерывов могут быть неисправности в аппаратуре, несовершенство ее эксплуатации, проведении измерений и другое. Перерывы и импульсные помехи являются основной причиной появления группирующихся ошибок при передаче дискретных сообщений по различным видам каналов связи. Импульсные помехи – это помехи, сосредоточенные во времени. Они представляет собой случайную последовательность импульсов, имеющих случайные амплитуды и следующих друг за другом через случайные интервалы времени, причем, вызванные ими переходные процессы не перекрываются во времени. Наиболее распространенные причины появления таких помех: коммутация соединения в электронной

аппаратуре, наводки с высоковольтных линий, грозовые разряды, прием отраженных сигналов.

В параграфе были рассмотрены причины ухудшения качества передаваемого сигнала, из-за которых в детектированном информационном сообщении возникают одиночные и групповые ошибки. Сбой при приеме любого элемента вызывает искажение цифровых данных, что может привести к катастрофическим последствиям. Существуют разные направления повышения помехоустойчивости систем цифровой связи. Основные направления будут рассмотрены в следующем параграфе.

## **1.2. Основные направления повышение помехоустойчивости**

Как уже говорилось ранее (см. п. 1.1), основными факторами, снижающими достоверность передачи информации, являются помехи и замирания в канале передачи информации [27, 28, 29]. Рассмотрим основные направления борьбы с этими отрицательными факторами.

### **1.2.1. Борьба с частотно-селективными замираниями**

Для борьбы с вызванной каналом межсимвольной интерференцией, которая возникает вследствие частотно-селективного замирания, может использоваться выравнивание [25, 30, 31, 32]. Процесс выравнивания для уменьшения воздействия межсимвольной интерференции заключается в использовании методов, собирающих рассеянную энергию символа в ее исходный временной интервал. По сути, эквалайзер (устройство выравнивания) является обратным фильтром канала. Если канал является частотно-селективным, эквалайзер усиливает частотные компоненты с малыми амплитудами и ослабляет с большими. Целью комбинации канала и выравнивающего фильтра является получение плоской частотной характеристики и линейного изменения фазы. Поскольку характеристика канала может со временем меняться, выравнивающий фильтр должен изменяться или

приспосабливаться к нестационарным характеристикам канала. Следовательно, такие фильтры являются адаптивными устройствами. Поскольку ослабление искажений выполняется путем сбора рассеянной энергии символа в исходный временной интервал символа, так, чтобы это не мешало детектированию других символов, эквалайзер попутно предоставляет приемнику энергию символа, которая в противном случае была бы утрачена.

### **1.2.2. Борьба с быстрыми замираниями**

Распространенным методом повышения помехоустойчивости цифровых линий связи в условиях быстрых замираний является разнесенный прием. Вместе с тем в цифровых системах передачи данных могут быть эффективно использованы широкополосные и составные сигналы, увеличение скорости передачи символов, решающая обратная связь в сочетании с помехоустойчивым кодированием, а также методы адаптивного приема [33]. Адаптация может производиться либо при приеме, либо при передаче. В последнем случае для получения информации о состоянии прямого тракта и соответствующего изменения параметров передаваемого сигнала используется канал обратной связи.

Наряду с известными методами разнесения (пространственным, частотным, угловым) особое внимание в цифровых системах уделяется комбинированным методам, например, частотно-временному, при котором предполагается использование последовательных или параллельных многочастотных сигналов (далее – МЧС). Очевидно, применение последовательного или параллельного МЧС предполагает использование соответственно одного или нескольких передающих устройств. Частотное разнесение может осуществляться одним или несколькими передатчиками, но для передачи  $M$  сигналов одним передатчиком требуется его значительная пиковая мощность и высокая линейность амплитудной характеристики. Поэтому, как правило, параллельный МЧС реализуется несколькими передатчиками.

Адаптивный прием сигналов основан на возможности измерения характеристик канала передачи сигналов и использовании полученных данных для

соответствующей регулировки параметров передаваемого сигнала. Условием создания замкнутых адаптивных систем является наличие канала обратной связи. Обратная связь может обеспечивать функционирование линии как с одной, оптимальной в текущий момент времени частотой, так и с группой частот. Достоинство такой схемы – уменьшение габаритов и массы оборудования. Недостатками схемы являются наличие обратного канала, а также переходные процессы вследствие переключений.

### **1.2.3. Борьба с преднамеренными помехами**

Для того, чтобы сохранить полезные свойства используемого класса сигналов и обеспечить их достаточную защищенность от оптимальной преднамеренной помехи, необходимо лишить постановщика помех возможности согласовать структуру помехи с сигналом [34]. Этого эффекта можно достичь, если каждый последовательный информационный символ передавать сложным символом с изменяемой во времени структурой, задаваемой специальным датчиком, строго синхронизированным с аналогичным датчиком на приемной стороне, то есть, организовав в радиоканале режим периодического изменения параметров. Поскольку закон формирования сигнала (изменения частот или форм сигналов) постановщику помех неизвестен, достижение полной оптимизации помех с сигналом достаточно затруднительно.

Еще одним из эффективных методов повышения помехозащищенности средств радиосвязи при воздействии преднамеренных помех является применение псевдослучайного перестроения рабочей частоты (далее – ППРЧ) [35, 36]. В системах с ППРЧ расширение спектра в пределах заданной полосы частот осуществляется посредством скачкообразного изменения частоты сигнала по псевдослучайному закону, который неизвестен постановщику помех. При этом сигнал занимает полосу частот значительно шире, и сопоставим с полосой, которая минимально необходима для передачи информации.

### 1.2.4. Кодирование с исправлением ошибок

На практике для повышения помехоустойчивости все чаще используют помехоустойчивые коды [1, 10, 11]. Помехоустойчивым кодам не важна причина возникновения ошибок. При помехоустойчивом кодировании в информационный сигнал вводятся избыточные символы, что приводит к снижению скорости передачи полезной информации. Для сохранения скорости передачи полезной нагрузки расширяют полосу частот канала или повышают кратность модуляции. Если скорость передачи информации меньше пропускной способности канала, то с использованием кодов, исправляющих ошибки, можно создать систему связи со сколь угодно малой вероятностью ошибки на выходе декодера канала.

На рисунке 1.2. [1] приведен характерный вид зависимости вероятности появления битовой ошибки от нормированного значения энергии бита, для случаев передачи не кодированной и кодированной последовательности бит. Применение кодирования с исправлением ошибок позволяет при тех же значениях энергии бита (излучаемой мощности) получить более высокую достоверность передачи данных.

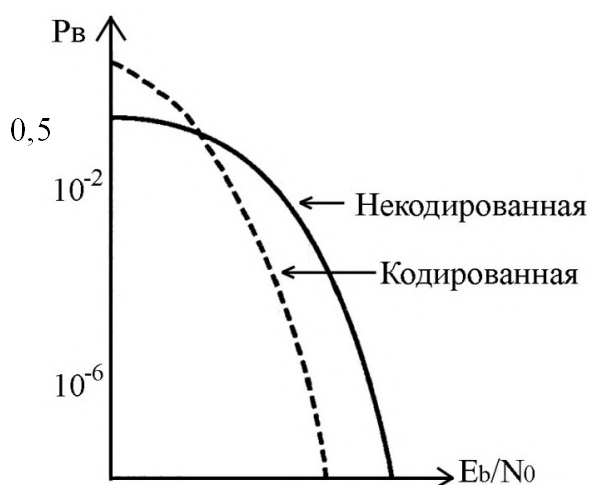


Рисунок 1.2. Достоверность передачи с кодированием и без него

Среди всех направлений повышения помехоустойчивости наибольшее распространение получило помехоустойчивое кодирование, позволяющее получить более высокие качественные показатели работы систем цифровой связи. Коротко рассмотрим основные характеристики помехоустойчивых кодов.

### 1.3. Основные характеристики блочных и сверточных помехоустойчивых кодов

Важное семейство помехоустойчивых кодов образуют линейные двоичные блочные коды, методы построения которых описаны в множестве источников [10, 11, 20, 37, 38, 39]. При блочном кодировании представляя информационные и кодовые слова в форме двоичных векторов, можно описать процессы кодирования и декодирования с помощью аппарата линейной алгебры, при этом компонентами вводимых векторов и матриц являются символы 0 и 1. Операции над двоичными компонентами производятся по правилам двоичной арифметики.

Кодер двоичного блочного  $(n, k)$  – кода (см. рис. 1.3) отображает множество  $2^k$  возможных двоичных информационных слов во множество  $2^n$   $n$  – мерных кодовых слов. Где  $k$  – число информационных символов (символов в информационном слове), а  $n$  – число кодовых символов (символов в кодовом слове). В теории кодирования между этими множествами всегда существует однозначное соответствие.



Рисунок 1.3. Кодер блочного кода

Вместо  $k$  бит информационного вектора в канал передается  $n$  бит кодового вектора. К каждому блоку данных кодирующее устройство прибавляет  $(n - k)$  избыточных бит, которые также называют контрольными битами или битами четности. Отношение числа избыточных бит к числу информационных бит,  $(n - k)/k$  называется избыточностью кода. Отношение числа бит данных к общему числу бит,  $k/n$ , называется степенью кодирования. Степень кодирования показывает долю кода, которая приходится на полезную информацию.



Кодирование линейного блочного  $(n, k)$  – кода задается порождающей матрицей  $G$  размером  $(k \cdot n)$ . Таким образом, кодовое слово  $v$  и информационное слово  $u$  связаны соотношением  $v = u \cdot G$ .

Правильно декодировать можно далеко не все модели ошибок. Возможности кода для исправления ошибок в первую очередь определяются его структурой. Весовой коэффициент Хэмминга  $w(U)$  кодового слова  $U$  определяется как число ненулевых элементов в  $U$ . Для двоичного вектора это эквивалентно числу единиц в векторе. Например, если  $U=100101101$ , то  $w(U) = 5$ . Расстояние Хэмминга между двумя кодовыми словами  $U$  и  $V$ , обозначаемое как  $d(U, V)$ , определяется количеством элементов, в которых они отличаются. Согласно свойствам сложения по модулю 2, можно отметить, что сумма двух двоичных векторов является другим двоичным вектором, двоичные единицы которого расположены на тех позициях, которыми эти векторы отличаются. Таким образом, можно видеть, что расстояние Хэмминга между двумя векторами равно весовому коэффициенту их суммы, то есть  $d(U, V) = w(U + V)$ . Также видно, что весовой коэффициент кодового слова равен его расстоянию до нулевого вектора. Наименьший элемент из множества расстояний между парами кодовых слов называется минимальным расстоянием кода и обозначается  $d_{min}$ . Минимальное расстояние дает меру минимальных возможностей кода и, следовательно, характеризует его помехоустойчивость. Сумма двух произвольных кодовых слов дает другой элемент кодовых слов, что является свойством линейных кодов. Если  $U$  и  $V$  кодовые слова, то и  $W = U + V$  тоже должно быть кодовым словом. Следовательно, расстояние между двумя кодовыми словами равно весовому коэффициенту третьего кодового слова. Таким образом, минимальное расстояние линейного кода соответствует минимальному весу кодового слова. Иными словами, минимальное расстояние соответствует наименьшему из множества расстояний между нулевым кодовым словом и всеми остальными кодовыми словами.

Линейный код с минимальным расстоянием Хэмминга  $d_{min} \geq 2t + 1$  может обнаружить  $d_{min} - 1$  ошибок и исправить  $t$  ошибок. Код, корректирующий ошибки в  $t$  битах, называется совершенным.

Систематический линейный блочный код – это код, в котором  $n$  символов кодового слова содержат  $k$  символов информационного сообщения. Таким образом, кодовое слово систематического кода имеет  $n - k$  бит четности и  $k$  информационных символов.

Порождающую матрицу любого систематического кода всегда можно привести путем перестановки столбцов к виду:

$$G_{k,n} = (P_{k(n-k)} I_k), \quad (1.1.)$$

где нижние индексы обозначают размерность матрицы, а  $I_k$  единичная матрица размером  $(k \cdot k)$ .

Циклические коды являются подмножеством линейных кодов. Линейный  $(n, k)$  – код является циклическим, если циклический сдвиг любого кодового слова также является кодовым словом этого кода. Циклический сдвиг соответствует сдвигу всех компонентов на один разряд вправо, причем, освободившееся место слева занимает крайняя правая компонента.

Сверточные коды [1, 3, 6, 10, 40] это коды, использующие непрерывную обработку потока данных короткими блоками. Сверточный кодер имеет память и символы на его выходе зависят не только от очередного блока символов на входе, но и от предыдущих символов. Сверточное кодирование является отображением информационной последовательности символов в кодовую последовательность с помощью линейной схемы с параметрами, не меняющимися во времени.

При сверточном кодировании очередная группа из  $k$  двоичных символов информационного сообщения путем линейных преобразований отображается в группу из  $n$  двоичных кодовых символов. При этом, как говорилось ранее, в отличие от блочных кодов при формировании группы кодовых символов используется не только данная группа информационных символов, но также  $S - 1$  предшествующих групп. Таким образом, сверточный код характеризуется тремя основными параметрами:  $n$ ,  $k$ ,  $S$ . Величину  $S$  называют длиной кодового ограничения. Произведение  $k$  и  $S$  определяет максимальное число двоичных информационных символов.

При кодировании произвольной информационной последовательности неограниченной длины, каждый из информационных символов участвует в формировании кодовых символов, составляющих  $S$  групп, в том числе группу, формируемую непосредственно в момент появления данного информационного символа, и  $S - 1$  следующих за ней групп. В таком случае рассматриваемый информационный символ оказывается «защищенным»  $S$  кодовыми группами, что и используется при различных методах декодирования сверточных кодов.

Теоретически, входная последовательность бит бесконечна, но на практике сверточным кодам обычно придают блоковую структуру, устанавливая состояние сверточного кодера в некоторое заранее известное состояние (например, нулевое).

Декодирование сверточных кодов обычно производится по алгоритму Витерби согласно критерию максимального правдоподобия.

В настоящее время сверточные коды применяются в разнообразных системах передачи и хранения цифровых данных, в мобильной и спутниковой связи [41].

Помехоустойчивые коды достаточно разнообразны и различаются методом кодирования-декодирования, числом кодируемых информационных бит, вносимой избыточностью, количеством исправляемых ошибок. Параметры помехоустойчивых кодов выбираются исходя из характеристик конкретной системы цифровой связи.

#### **1.4. Противоречие между основными характеристиками систем цифровой связи**

Существуют две основные характеристики системы цифровой связи: скорость передачи и достоверность передачи информации. Первая характеристика определяет количество передаваемой информации, вторая – качество передачи [20].

В качестве количественной меры достоверности можно принять вероятность ошибки сообщения или его элемента. В реальной системе передачи информации

достоверность определяется степенью искажения передаваемого сигнала. Эти искажения зависят от свойств и технического состояния системы, а также от интенсивности и характера помех. В правильно спроектированной и технически исправной системе передачи информации искажения сигналов в большей степени обусловлены воздействием помех. В этом случае достоверность передачи сообщений определяется помехоустойчивостью системы, то есть способностью системы противостоять вредному влиянию помехи на передачу сообщений.

Стремление к достижению высоких показателей для основных характеристик системы цифровой связи содержит в себе противоречие. На практике это противоречие разрешается путем решения одной из двух проблем. Суть первой проблемы заключается в том, чтобы обеспечить предельно возможную достоверность приема сообщений при заданной скорости передачи информации по каналу связи. Эта проблема рассматривается в теории потенциальной помехоустойчивости, которая позволяет сравнить между собой различные системы связи по помехоустойчивости, указывает общие пути повышения помехоустойчивости конкретных систем и позволяет решать задачи синтеза оптимальных по помехоустойчивости отдельных устройств или системы в целом.

Вторая проблема состоит в том, чтобы передать по системе с заданной достоверностью наибольшее количество информации наиболее экономным способом. Эта проблема освещается в теории информации, которая устанавливает зависимость между вероятностными свойствами сообщений и количеством содержащейся в них информации, а также позволяет сравнивать по эффективности различные системы связи и указывает резервы, за счет которых может быть повышена скорость передачи данных. Скорость передачи информации по каналу связи измеряется количеством информации, передаваемой в единицу времени. Максимальная скорость передачи информации, которую может обеспечить канал связи с данными характеристиками, называется его пропускной способностью.

Решение описанных проблем может быть осуществлено путем применения помехоустойчивого кодирования передаваемой информации. Например, снижение вероятности битовой ошибки в декодированном информационном сообщении при

неизменной скорости передачи информации, направлено на разрешение первой проблемы. При этом скорость передачи информации определяется избыточностью, вносимой при помехоустойчивом кодировании информационных сообщений. Минимизация избыточности помехоустойчивого кода, реализованная наиболее экономным способом, при заданной допустимой величине вероятности битовой ошибки в декодированном информационном сообщении, направлена на разрешение второй проблемы.

Параметры помехоустойчивых кодов должны удовлетворять характеристикам конкретной системы связи. Не существует помехоустойчивых кодов, лучших для всех систем цифровой связи. Подходы к оценке эффективности помехоустойчивых кодов могут быть различны.

### **1.5. Теоретические основы оценки эффективности помехоустойчивого кодирования**

Результаты анализа научных трудов и доступных источников в области исследования вопросов оценки эффективности показали, что, несмотря на широкую распространенность данного термина, общепринятого и единого системного подхода к определению понятия «эффективность» нет [42]. Термин «эффективность» (от лат. *effektivus* – исполнение, действие) связывается с системой и рассматривается как «дающий эффект» или «приводящий к нужным результатам», то есть результативный, действенный [43].

В настоящий момент имеется три основных понимания эффективности: эффективность в смысле действенности (*Effectiveness*) – это способность производить эффект (результат) неких действий, которая не всегда может быть измерена при помощи количественных показателей; эффективность в смысле продуктивности, производительности, экономичности (*Efficiency*) – это показатель эффективности деятельности, отражающий сумму выработки на единицу затрат (чем меньше ресурсов затрачено на достижение запланированных результатов, тем

выше продуктивность); эффективность в смысле результативности, оптимальности (Efficacy) – это способность производить намеченный результат в желаемом объеме, может выражаться мерой (процентное соотношение) фактически произведенного результата к нормативному/запланированному. Эта мера фокусируется на достижении как таковом, а не на ресурсах, затраченных на достижение желаемого эффекта. При этом действия, дающие результат, не обязательно будут оптимальны, а то, что оптимально, не обязательно будет экономично. Только сочетание всех этих параметров означает эффективность в полном смысле этого слова [44].

Количественно эффективность оценивается с помощью показателя эффективности [45]. Показатели эффективности – это основные числовые характеристики, с помощью которых оценивается качество функционирования системы. Основные показатели позволяют оценить технологические процессы и операции агрегировано по всем характеристикам, частные характеризуют только ограниченное число свойств. Определение состава и содержания системы показателей, необходимых для проведения оценки эффективности, является классической задачей системного анализа [46, 47].

На основании анализа современной учебной литературы [1, 9, 11], в части, касающейся помехоустойчивого кодирования, основными показателями являются:

- кодовая скорость;
- вероятность битовой ошибки в декодированном информационном сообщении;
- энергетический выигрыш от применения помехоустойчивого кодирования;
- затраченные вычислительные ресурсы;
- сложность аппаратной реализации.

Эффективность помехоустойчивого кодирования оценивается по некоторым критериям оценки. Критерий – 1) средство для вынесения суждения; стандарт для сравнения; правило для оценки; 2) мера степени близости к цели [48, 49].

Возможны следующие варианты оценки эффективности: оценка состояния эффективности на основе сравнения показателей предлагаемого и реализуемого

помехоустойчивого кода; сравнение показателей помехоустойчивого кода с предельно достижимыми; сравнительная оценка показателей двух и более помехоустойчивых кодов, имеющих ряд схожих характеристик (количество информационных бит, кодовая скорость, и пр.).

Основываясь на результатах анализа научных трудов и доступных источников в области исследования вопросов оценки эффективности, исходя из рассмотренных в п. 1.4. направлений разрешения противоречия, возникающего при стремлении к достижению высоких показателей скорости и достоверности передачи данных, предлагается считать основными критериями эффективности помехоустойчивого кодирования:

- степень приближения характеристик кода к пределу К. Шеннона;
- достижение требуемой величины вероятности битовой ошибки;
- достижение требуемого значения кодовой скорости;
- снижение величины вероятности битовой ошибки при заданной избыточности кода;
- снижение избыточности кода для обеспечения требуемой вероятности битовой ошибки;
- снижение энергетических и временных затрат на кодирование-декодирование.

В рассмотренном параграфе произведено формирование перечня критериев эффективности помехоустойчивых кодов. Большинство критериев взаимосвязаны, следовательно, существуют способы повышения эффективности помехоустойчивых кодов сразу по нескольким критериям.

## **1.6. Современные подходы к повышению эффективности помехоустойчивых кодов**

Существует большое число способов повышения эффективности помехоустойчивых кодов. В параграфе на основе анализа научных публикаций

последних лет выделены основные подходы к повышению эффективности помехоустойчивых кодов.

### 1.6.1. Комбинированные помехоустойчивые коды

С целью придания помехоустойчивым кодам нужных характеристик применяют разнообразные сочетания из конструкций кодов, методов кодирования-декодирования и модуляции. К комбинированным кодам относятся: турбокоды, каскадные коды, коды произведения, гибридные коды, сигнально-кодовые конструкции [11].

Одними из самых распространенных комбинированных кодов являются последовательные каскадные коды (далее – ПКК). В ПКК кодирование осуществляется в два уровня [50, 51, 52]. Входные данные сначала кодируются внешним кодом, а затем внутренним, после чего осуществляется модуляция. Декодирование происходит в обратном порядке. Искорженные каналом данные с демодулятора поступают сначала на декодер внутреннего кода, а затем на декодер внешнего кода. Структурная схема канала связи с использованием ПКК изображена на рисунке 1.4.

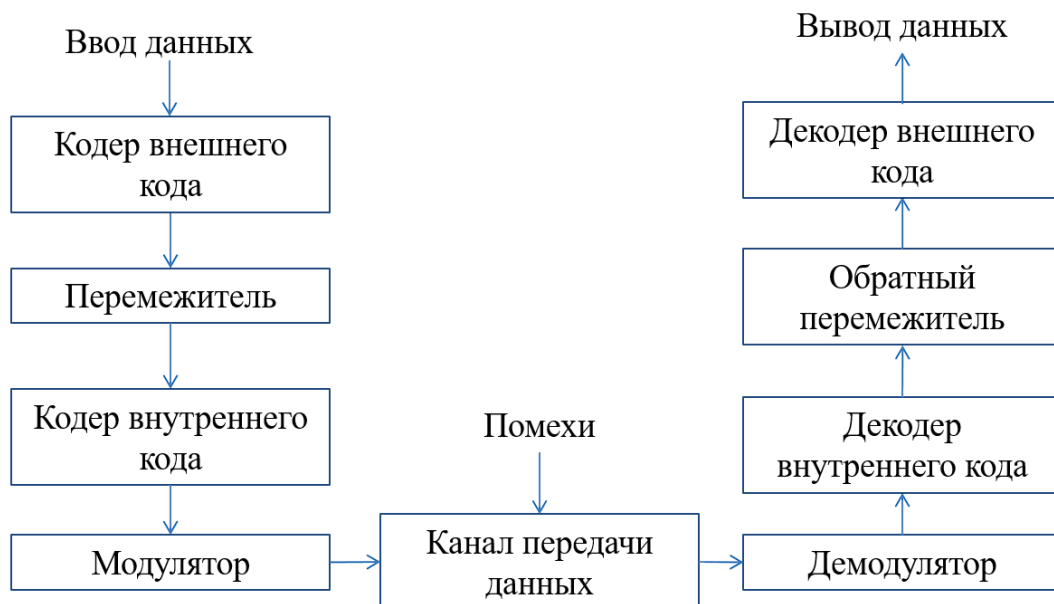


Рисунок 1.4. Структурная схема канала связи с применением ПКК



Достоинством каскадных кодов является относительно низкая сложность кодирующих и декодирующих устройств, так как каскадные коды позволяют выполнить процедуры кодирования и декодирования по этапам. На каждом этапе возможно применение достаточно коротких по сравнению с результирующим кодов.

Использование каскадных кодов позволяет реализовать достаточно большое кодовое расстояние, поэтому их применение в каналах с высоким уровнем помех эффективно.

### **1.6.2. Декодирование мягкого решения детектора**

Существует два варианта детектирования прошедших демодуляцию символов – с принятием жесткого или мягкого решения о значении символа. В случае принятия детектором жесткого решения, для примера, когда один символ равен одному биту, если напряжение, фаза или частота (в зависимости от вида модуляции) превышает пороговое значение, оно декодируется как единица, а в противном случае – как ноль. Таким образом, для случая принятия жесткого решения можно сказать что сигнал квантуется на два уровня. Оценить близость значения символа к порогу при формировании жесткого решения невозможно. Для случая принятия мягкого решения о значении принятого символа, выход детектора представляет собой квантованные на более чем два уровня (как правило, на четыре, восемь или шестнадцать уровней) действительные числа, соответствующие значению напряжения, фазы или частоты принятого сигнала. Мягкое решение дает декодеру больше информации о значении принятого символа, что конвертируется в повышение корректирующей способности кода. Энергетический выигрыш при декодировании мягкого решения составляет единицы децибел [10]. Работа декодера с мягким решением, как правило, более сложна. Для ряда блочных кодов до настоящего времени не создано разумных по сложности декодирования алгоритмов, работающих с мягким решением. Значительным преимуществом мягкого кодирования является факт роста эффективности помехоустойчивого кодирования без увеличения избыточности кода. В последние годы особенности

декодирование мягких решений детекторов рассматривались в научных публикациях разных авторов [53, 54, 55]. В диссертационном исследовании эффективность тех или иных кодов в, большинстве случаев, будет рассматриваться как для случая принятия детектором жесткого решения, так и для случая мягкого решения детектора о значении принятого символа.

### 1.6.3. Перемежение

Распространение сигнала в среде с быстрыми замираниями приводит к появлению коррелированных ошибок, имеющих вид пакетов. Пакеты ошибок также возникают из-за наличия в каналах импульсных помех. Большинство блочных и сверточных кодов разрабатывается для борьбы с независимыми случайными ошибками. Перемежение (чередование) бит [11, 12, 56] кодированного сообщения перед передачей и обратная операция после приема приводят к рассеиванию пакета ошибок во времени: таким образом, они становятся для декодера случайно распределенными. Идея, лежащая в основе метода чередования бит, заключается в разнесении символов кодовых слов во времени. Получаемые промежутки времени точно так же заполняются символами других кодовых слов. Разнесение символов во времени эффективно превращает канал с памятью в канал без памяти и, следовательно, позволяет использовать коды с коррекцией случайных ошибок в системах с замираниями и импульсными помехами.

Устройство чередования смешивает кодовые символы в промежутке нескольких длин блоков (для блочных кодов) или нескольких длин кодового ограничения (для сверточных кодов). Требуемый промежуток определяется длительностью пакета. Подробности структуры перераспределения бит должны быть известны приемнику, чтобы иметь возможность выполнить восстановление порядка бит перед декодированием. На рисунке 1.5 показан простейший пример чередования. В верхней части рисунка показаны исходные кодовые слова от *A* до *E* без чередования. Пусть каждое кодовое слово состоит из пяти кодовых символов. Допустим, что наш код может исправлять один ошибочный символ в кодовом слове. Если промежуток памяти канала равен длительности одного кодового слова

или мы просто имеем дело с импульсной ошибкой длиной в пять символов, то такой пакет может стать причиной ошибок в двух кодовых словах. В нижней части рисунка показаны кодовые слова после перемежения символов. После процедуры перемежения каждый кодовый символ каждого кодового слова отделяется от своего соседа на расстояние пяти символов. Полученный поток затем преобразуется в модулированный сигнал и передается по каналу.

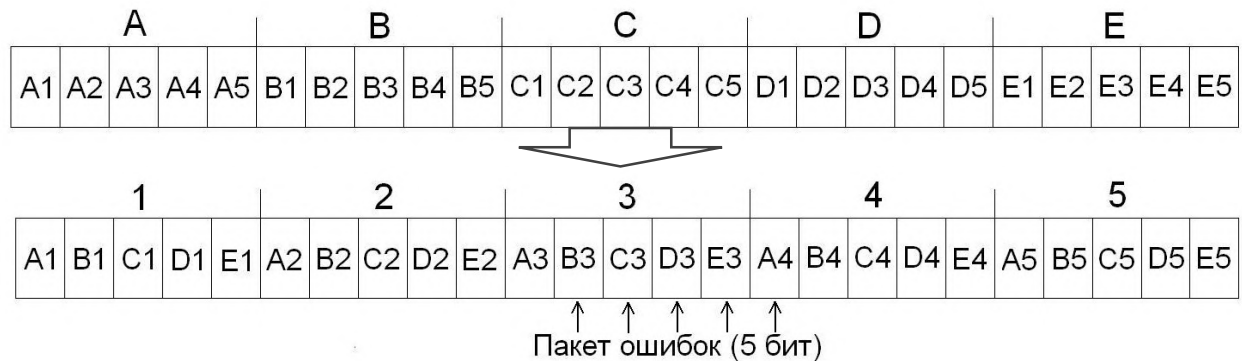


Рисунок 1.5. Пример перемежения бит

В нижней части рисунка 1.5 схематично показано воздействие пакета ошибок длиной в пять символьных промежутков на кодовую последовательность после перемежения. На приемном устройстве в потоке символов вначале восстанавливается исходный порядок, так что он становится похож на исходную кодированную последовательность, изображенную в верхней части рисунка 1.5. Далее поток декодируется. Поскольку в каждом кодовом слове возможно исправление одного ошибочного символа, импульсная помеха не оказывает никакого влияния на результат декодирования. В последние годы алгоритмы перемежения и использование перемежения в комбинированных кодах рассматривались в научных публикациях разных авторов [57, 58, 59].

#### 1.6.4. Адаптация параметров кода

Качество канала передачи данных может значительно варьироваться при изменении характеристик среды распространения и/или расстояния между приемником и передатчиком. Качество каналов связи может существенно ухудшаться при воздействии дестабилизирующих факторов, обусловленных

изменением условий распространения сигнала или дальности передачи для мобильных узлов связи. Большой нестабильностью обладает качество КВ и УКВ радиосвязи, особенно подвижной. При использовании каналов передачи данных, качество которых может изменяться, корректирующая способность кода часто выбирается с целью обеспечения максимально допустимой вероятности появления ошибочного бита в информационном сообщении ( $P_{DMAX}$ ), исходя из прогнозируемого наихудшего состояния канала, которому соответствует наибольшая вероятность битовой ошибки ( $P_{BMAX}$ ). Это может привести к введению значительной избыточности и, как следствие, к существенному снижению скорости передачи данных. Если в данный момент времени вероятность битовой ошибки в канале передачи данных меньше, чем  $P_{BMAX}$ , для обеспечения  $P_{DMAX}$  допустима меньшая избыточность кода. Следовательно, в таком случае эффективность используемого кода снижается из-за неоправданно высокой избыточности. Особенно заметно этот недостаток проявляется при интенсивном информационном обмене в условиях нестационарного канала низкого качества. В некоторых случаях качество канала связи снижается ниже прогнозируемого минимума и декодер помехоустойчивого кода не в состоянии обеспечить допустимую вероятность появления ошибочного бита в информационном сообщении.

Для учета нестабильности канала передачи данных, разрабатываются адаптивные системы кодирования, способные динамически подстраивать уровень вносимой избыточности в зависимости от текущего состояния канала связи. Существуют различные подходы к реализации таких систем, наиболее простой из которых является предварительная подготовка набора кодеков с динамическим переключением между ними. Такой способ не всегда является оптимальным, так как при переходе с одного кода на другой изменять зачастую необходимо лишь некоторые параметры кода [51]. Гораздо более эффективным способом адаптации является динамическая перестройка только тех параметров кода, которые отвечают за его избыточность [61]. В этой связи исследования, направленные на разработку

гибких адаптивных алгоритмов применения помехоустойчивых кодов, являются актуальными.

Процедура выбора помехоустойчивого кода, соответствующего каналу связи, содержит в себе внутреннее противоречие. С одной стороны, на этапе проектирования системы связи выбор кода выполняется исходя из предполагаемых характеристик канала, с другой стороны, на этапе эксплуатации системы, характеристики реального канала связи могут отличаться от тех характеристик, которые были приняты при проектировании системы связи. Разрешение данного противоречия заключается в применении кодов, позволяющих производить коррекцию параметров в ходе эксплуатации, обеспечивая вероятность доведения сообщения, равную заданной величине при минимальной избыточности помехоустойчивого кода. Очевидно, что регулировка скорости кода и избыточности должна быть по возможности плавной, адекватно изменениям качества канала передачи данных. В последние годы исследования адаптивных возможностей помехоустойчивых кодов рассматривались в публикациях разных авторов [60, 62, 63, 64]. Далее в главе будут рассмотрены два наиболее распространенных способа адаптации помехоустойчивых кодов.

#### **1.6.5. Модификация порождающих и проверочных матриц блочного кода**

Важность возможности изменения характеристик кода с целью адаптации его параметров к изменениям качества канала связи была рассмотрена в п. 1.4. Изменение характеристик кода может быть осуществлено путем применения разнообразных модификаций порождающих и проверочных матриц. Наиболее простыми из модификаций матриц являются укорочение и расширение [10, 12]. Рассмотрим суть реализации таких модификаций.

Укорочение блочных кодов реализуется следующим образом. При задании порождающей матрицы блочного кода в систематическом виде из единичной матрицы, происходит удаление ряда столбцов. Код, образующийся с использованием такой модифицированной матрицы, называется укороченным.

Расширение блочных кодов реализуется следующим образом. Проверочная матрица кода расширяется путем добавления определенного числа строк и столбцов. Наиболее известен способ расширения путем добавления общей проверки на четность. Порождающая матрица кода образуется путем преобразования модифицированной проверочной матрицы. В последние годы особенности кодирования и декодирования с использованием модифицированных матриц рассматривались в публикациях разных авторов [65, 66].

### **1.6.6. Перфорация кодовой последовательности**

Наиболее простым способом снижения избыточности кода, например, при улучшении качества канала связи, является перфорация кода. Перфорация заключается в выкалывании определенного количества кодовых бит перед передачей. Регулируя число выколотых бит можно регулировать избыточность, и таким образом изменять корректирующую способность применяемого помехоустойчивого кода. Недостатком перфорации является невозможность с ее помощью улучшить корректирующую способность кода. Можно только регулировать степень ухудшения. В диссертационном исследовании ряд идей по повышению эффективности помехоустойчивых кодов будет основан на анализе эффективности перфорации. В последние годы особенности введения перфорации и декодирования перфорированных кодов рассматривались в научных публикациях разных авторов [67, 68, 69, 70].

В настоящее время с целью повышения эффективности помехоустойчивых кодов используются разнообразные методы. В параграфе были рассмотрены методы, описание использования которых наиболее часто встречалось в публикациях современных ученых.

## Выводы по главе 1

В главе были рассмотрены причины ухудшения качества передаваемого сигнала, из-за которых в детектированном информационном сообщении возникают одиночные и групповые ошибки. Сбой при приеме любого элемента вызывает искажение цифровых данных, что может привести к катастрофическим последствиям. Существуют разные направления повышения помехоустойчивости систем цифровой связи. Среди всех направлений наибольшее распространение получило помехоустойчивое кодирование, позволяющее получить более высокие качественные показатели работы систем цифровой связи. Помехоустойчивые коды достаточно разнообразны и различаются методом кодирования-декодирования, числом кодируемых информационных бит, вносимой избыточностью, количеством исправляемых ошибок. Параметры помехоустойчивых кодов выбираются исходя из характеристик конкретной системы цифровой связи. Не существует помехоустойчивых кодов, лучших для всех систем цифровой связи. Помехоустойчивые коды могут быть использованы для разрешения противоречия между стремлением к росту скорости и повышению достоверности передачи данных.

В главе произведено формирование перечня критериев эффективности помехоустойчивых кодов. Большинство критериев взаимосвязано, следовательно, существуют методы повышения эффективности помехоустойчивых кодов сразу по нескольким критериям. В настоящее время с целью повышения эффективности помехоустойчивых кодов используются разнообразные методы. Были рассмотрены методы, описание использования которых наиболее часто встречалось в публикациях современных ученых. Дальнейшие исследования будут осуществляться с опорой на рассмотренные методы повышения эффективности помехоустойчивых кодов.

Один из распространенных и активно развивающихся методов повышения эффективности помехоустойчивых кодов заключается в комбинировании кодов.

Сверточные коды являются одним из обширных классов помехоустойчивых кодов. В следующей главе будут исследованы возможности по повышению эффективности последовательных каскадных кодов, построенных на основе внешнего сверточного кода.



## Глава 2. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПОСЛЕДОВАТЕЛЬНЫХ КАСКАДНЫХ КОДОВ, ПОСТРОЕННЫХ НА ОСНОВЕ ВНЕШНЕГО СВЕРТОЧНОГО КОДА

Данная глава посвящена вопросам повышения эффективности последовательных каскадных кодов, построенных на основе внешнего сверточного кода. В начале главы рассматривается вариант построения сверточного кода, наиболее часто упоминаемого в дальнейшем тексте главы. Вводятся термины и обозначения, принятые при рассмотрении процессов кодирования и декодирования сверточных кодов. Изложенные в главе 1 способы повышения эффективности, такие как мягкое декодирование и перфорация рассматриваются применительно к сверточным кодам. Дается числовая оценка эффективности рассмотренных способов. Исследуется реализация последовательного каскадного кодирования на основе внешнего сверточного кода. Предлагается модификация метода построения каскадного кода с реализацией плавной адаптации параметров кода. С целью улучшения корректирующей способности предлагается модификация метода декодирования по максимуму правдоподобия с применением коэффициентов, пропорциональных достоверности значений декодируемых бит.

### 2.1. Корректирующая способность сверточных кодов

Сверточные коды получили широкое распространение в современных телекоммуникационных системах [71, 72, 73, 74, 75, 76]. В последние годы по направлению повышения эффективности сверточных кодов опубликованы научные работы разных авторов [77, 78, 79, 80]. Принципы сверточного кодирования были коротко изложены в п. 1.3. В данной главе в качестве примера сверточного кода будет использоваться несистематический код с параметрами  $k = 1$ ,  $n = 2$ ,  $S = 3$ , гарантировано исправляющего два любых подряд следующих

ошибочных бита в бесконечной кодовой последовательности. Рассмотрим процесс кодирования и декодирования для данного кода более подробно, определившись с применяемыми здесь и далее терминологией и обозначениями. Процесс кодирования и декодирования сверточных кодов, подробно описан в разнообразных источниках [81, 82, 83, 84].

В ответ на поступивший в кодер информационный бит формируется группа из двух кодовых бит ( $y_1, y_2$ ). Значения кодовых бит являются линейной комбинацией входного бита  $u_i$  и двух предыдущих информационных бит ( $u_{i+1}, u_{i+2}$ ):

$$y_1 = u_i \oplus u_{i+1} \oplus u_{i+2}. \quad (2.1)$$

$$y_2 = u_i \oplus u_{i+2}. \quad (2.2)$$

Структурная схема кодера изображена на рисунке 2.1.

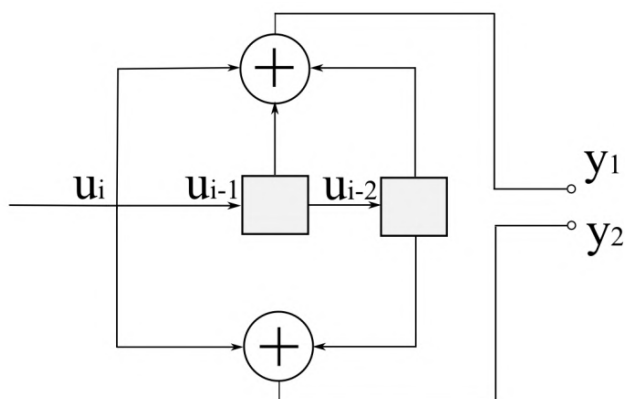


Рисунок 2.1. Структурная схема кодера

Предыдущие информационные биты хранятся в ячейках памяти. Очевидно, что содержимое ячеек памяти кодера может меняться лишь определенным образом. Возможные варианты перехода из одной пары значений ячеек памяти в другую пару показаны на диаграмме состояний кодера (см. рис. 2.2). На диаграмме внутри четырех кругов обозначены все возможные состояния ячеек памяти: 00, 01, 10, 11. Стрелками показаны возможные переходы между состояниями. При этом поступлению на вход кодера нулевого бита соответствует сплошная стрелка перехода, а пунктирная стрелка перехода соответствует поступлению на вход кодера единицы. Пары кодовых бит, образующихся при переходах из одного состояния ячеек памяти в другое, показаны у стрелок переходов курсивом.

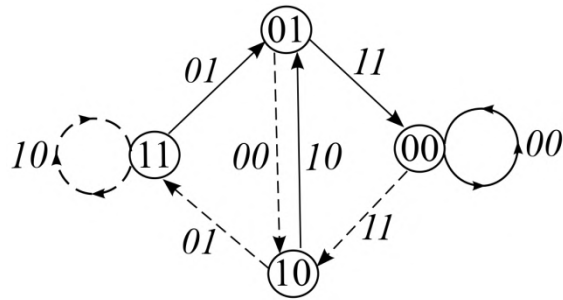


Рисунок 2.2. Диаграмма состояний кодера

Процесс кодирования можно рассматривать как перемещение по решетке, узлами которой являются значения ячеек памяти кодера (см. рис. 2.3). Из каждого узла существует два разрешенных перехода. Переход по верхней ветви соответствует поступлению на вход кодера нулевого информационного бита, по нижней ветви – поступлению на вход кодера единицы. При переходе между узлами происходит генерация пары бит, соответствующих значению ветви. На рисунке 2.3 и далее применены следующие обозначения: цифры в кружках соответствуют значениям ячеек памяти кодера, стрелки (ветви) обозначают разрешенные переходы между значениями ячеек памяти, пары чисел на стрелках обозначают значения ветвей, соответствующие генерируемым битам. Ветви имеют значения, соответствующие всем возможным комбинациям пар бит: «00», «01», «10», «11». В ходе перемещения по решетке слева направо из ветвей образуется путь. Путь, образовавшийся при кодировании информационной последовательности, будем называть истинным. Ветви истинного пути показаны на рисунке сплошными жирными линиями. На рисунке 2.3 схематично показан процесс кодирования последовательности 110110, начавшийся при нулевых значениях ячеек памяти. При этом, произошла генерация кодовой последовательности 11 01 01 00 01 01.

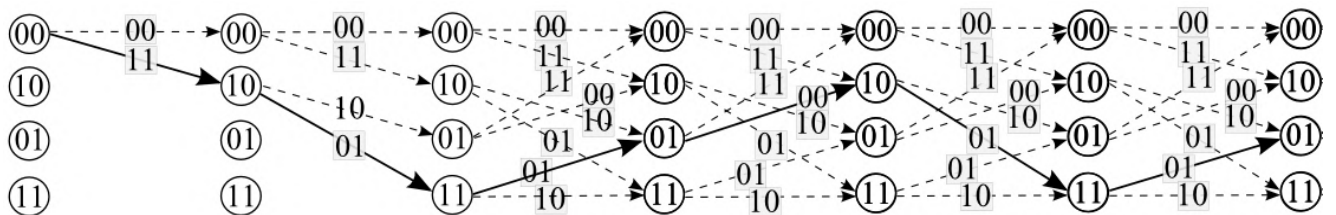


Рисунок 2.3. Кодирование информационной последовательности 110110

В ходе итеративного процесса кодирования при поступлении очередного бита на вход кодера, на выходе кодера формируется пара кодовых бит. Очевидно, что процесс декодирования также итеративен, и на вход декодера биты поступают парами. При декодировании сверточных кодов широкое распространение (протоколы V.32, V.34, ADSL, HDSL) получил алгоритм Витерби. В соответствии с данным алгоритмом, на каждом шаге декодирования для каждой ветви вычисляется метрика – число, характеризующее степень отличия пары бит, генерируемых данной ветвью (значения ветви) от принятой кодовой последовательности. Чем сильнее отличие, тем больше метрика. Метрика ветви вычисляется в общем случае как сумма вычисляемой посимвольно разности значения ветви и символов, поступивших на вход декодера. Рассмотрим вычисление метрики в случае декодирования жесткого решения детектора. Если значение ветви полностью совпадает с принятой парой бит, то метрика данной ветви – нулевая. Если имеется различие в одном бите, то метрика равна единице. Если различие в обоих битах, то метрика равна двум. Рассмотрим пример вычисления метрики. Пусть на вход декодера поступила очередная пара символов «10». Тогда, при значении ветви «10», метрика ветви будет равна 0, так как в парах символов оба символа совпадают. При значении ветви «01», метрика ветви будет равна 2, так как оба символа различны. Сумма метрик ветвей, входящих в путь, образует метрику пути. После каждого шага декодирования (начиная с третьего шага) из двух путей, входящих в один узел решетки, остается путь, имеющий меньшую метрику. Оставшиеся четыре пути из восьми называются выжившими. После последнего шага декодирования из четырех выживших путей определяется один с минимальной метрикой. На основании финального выжившего пути однозначно определяется декодированная информационная последовательность. Таким образом, в соответствии с алгоритмом Витерби, из всех разрешенных последовательностей выбирается наиболее близкая к принятой кодовой последовательности.

Рассмотрим процесс декодирования конечной кодовой последовательности, состоящей из восьми нулей при наличии ошибки в последнем кодовом бите.

Схематично процесс декодирования показан на рисунке 2.4, и кроме последнего шага полностью соответствует ранее рассмотренному процессу декодирования при отсутствии ошибок. В верхней части рисунка здесь и далее курсивом обозначаются пары бит декодируемой последовательности. При наличии ошибок, символ содержащий ошибку подчеркнут. Рядом с узлами решетки приводятся значения метрик путей. Справа от узла приводятся значения метрик путей, исходящих из узла, слева значения метрик путей, входящих в узел. Для рассматриваемого примера, выжившие после последнего шага пути имеют метрики 1, 1, 3, 2. Таким образом, с одинаковой вероятностью финальным выжившим путем может быть признан как путь 00000000, так и путь 00000011. В случае выбора декодером пути 00000011 в качестве финального выжившего, данный путь с использованием диаграммы состояний преобразуется в информационную последовательность 0001. Резюмируем, что в случае наличия ошибки в последнем бите конечной кодовой последовательности вероятность появления ошибочного бита равна 0,5.

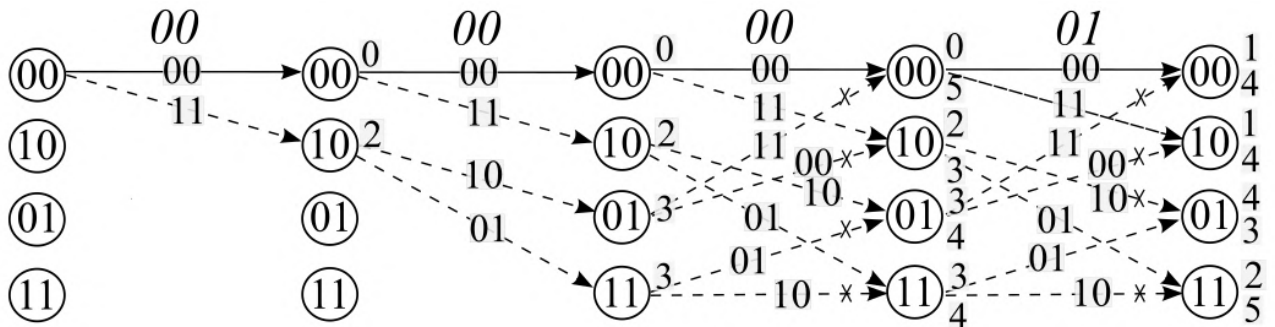


Рисунок 2.4. Декодирование при наличии ошибки в последнем бите

В процессе кодирования для рассматриваемого кода за один такт кодирования формируются два кодовых бита ( $n = 2$ ), в связи с этим возникновение ошибки в предпоследнем бите приводит к тем же последствиям, что и возникновение ошибки в последнем бите (см. рис. 2.5). Аналогичные выводы справедливы для последних бит кодовой последовательности любого сверточного кода.

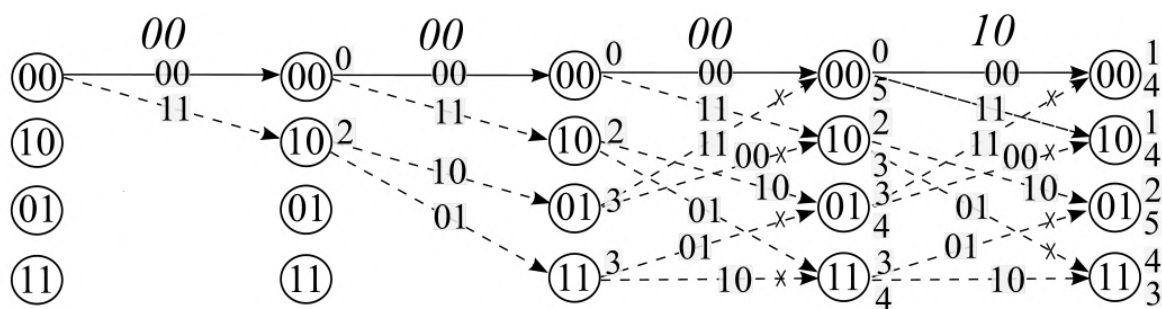


Рисунок 2.5. Декодирование при наличии ошибки в предпоследнем бите

Рассмотрим процесс декодирования конечной кодовой последовательности, состоящей из восьми нулей при наличии ошибки в последнем и предпоследнем кодовых битах. Схематично процесс декодирования показан на рисунке 2.6. После последнего шага декодирования путь 00000011 имеет нулевую метрику и однозначно определяется как финальный выживший. Резюмируем, что наличие ошибки в последнем и предпоследнем кодовых битах всегда приводит к появлению ошибки в последнем бите информационной последовательности.

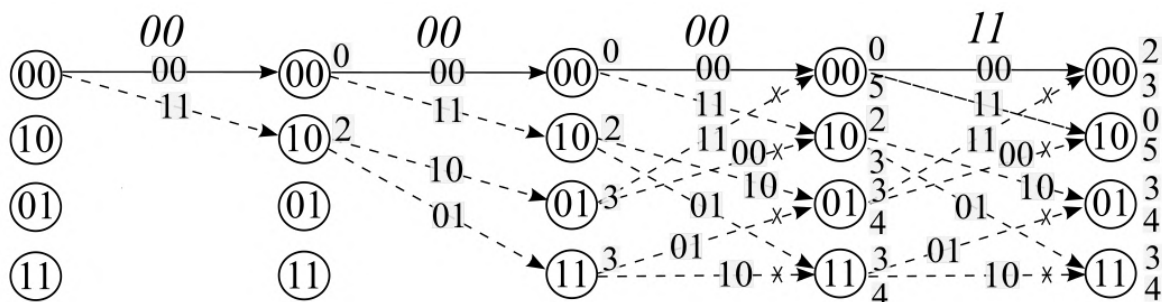


Рисунок 2.6. Декодирование при наличии ошибки в двух последних битах

Рассматриваемый сверточный код, гарантировано исправляющий два любых подряд следующих ошибочных бита в бесконечной кодовой последовательности, не способен эффективно бороться с ошибками в последних битах конечной кодовой последовательности. Очевидно, что чем меньше будет размер блока, тем больше будет вероятность появления ошибочного бита в декодированной информационной последовательности.

С целью реализации исследований корректирующей способности сверточного кода, на языке программирования C++ создан программный имитатор, моделирующий работу кодера-декодера сверточного кода в двоичном

симметричном канале с независимыми ошибками. Программный имитатор содержит в своем составе следующие основные компоненты:

- генератор информационной последовательности;
- кодер;
- блок моделирования выхода детектора приемника с заданной вероятностью появления ошибочного бита  $p_B$ ;
- декодер Витерби.

Описание программного генератора информационной последовательности и моделирования выхода детектора приемника с заданной вероятностью появления ошибочного бита, приведено в п. 2.2.

В таблице 2.1 представлены полученные по итогам работы программы значения вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$  в зависимости от длины информационного сообщения  $L$ , при разном значении вероятности появления ошибочного бита в канале передачи данных [85]. Для удобства восприятия, данные таблицы 2.1 представлены в графическом виде на рисунке 2.7.

Таблица 2.1

Вероятность появления ошибочного бита в декодированном информационном сообщении

№ п/п	$L$ , бит	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
1.	4	$4,22 \cdot 10^{-3}$	$4,31 \cdot 10^{-4}$	$4,90 \cdot 10^{-5}$
2.	10	$1,42 \cdot 10^{-3}$	$1,02 \cdot 10^{-4}$	$8,97 \cdot 10^{-6}$
3.	20	$7,46 \cdot 10^{-4}$	$4,60 \cdot 10^{-5}$	$3,97 \cdot 10^{-6}$
4.	50	$3,44 \cdot 10^{-4}$	$1,90 \cdot 10^{-5}$	$1,93 \cdot 10^{-6}$
5.	100	$1,87 \cdot 10^{-4}$	$9,30 \cdot 10^{-6}$	$8,80 \cdot 10^{-7}$
6.	200	$1,34 \cdot 10^{-4}$	$6,30 \cdot 10^{-6}$	$3,80 \cdot 10^{-7}$
7.	10000	$4,75 \cdot 10^{-5}$	$5,20 \cdot 10^{-8}$	$6,42 \cdot 10^{-11}$

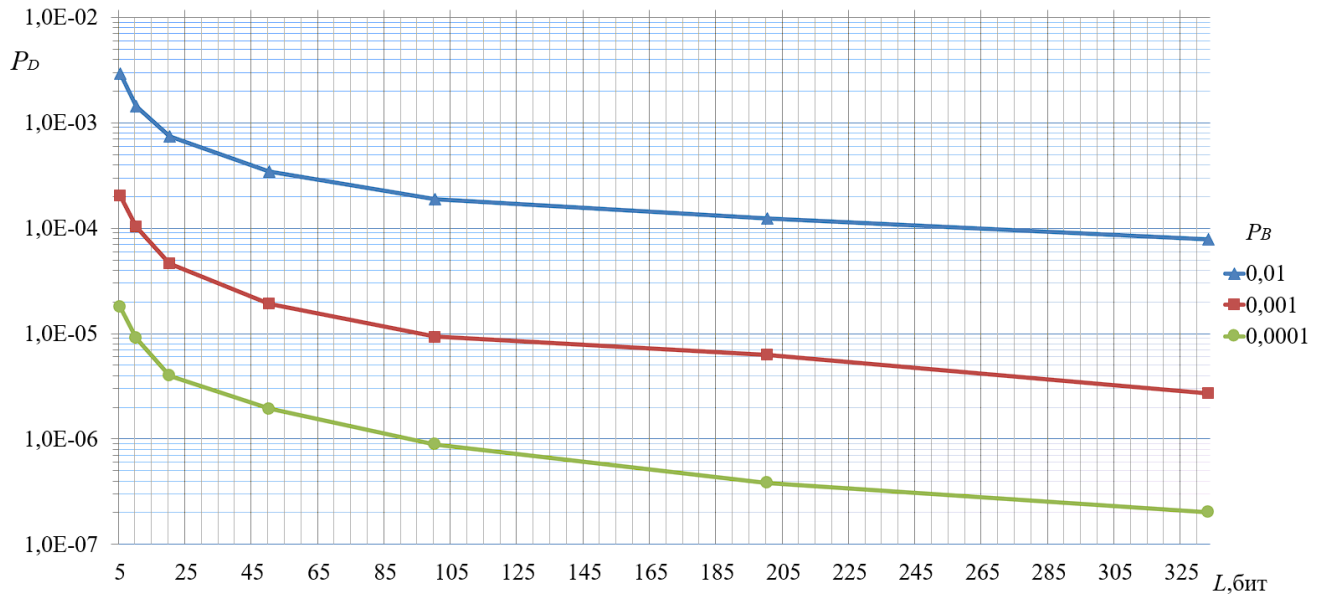


Рисунок 2.7. Зависимость вероятности ошибочного бита от длины информационного сообщения

Компенсировать возросшую вероятность появления ошибочного бита в декодированном информационном сообщении можно снизив вероятность появления ошибочного бита в канале передачи данных за счет увеличения энергии бита. Энергозатраты при этом можно оценить, используя характеристику канала передачи данных, выраженную через зависимость вероятности битовой ошибки в канале передачи данных  $p_B$  от  $E_b/N_0$ . С использованием приведенной на рисунке 2.8 характеристики можно произвести оценку энергетических затрат в децибелах для канала передачи данных с белым аддитивным гауссовым шумом при использовании модуляции BPSK. Для сообщения длиной в единицы и десятки бит потери будут составлять единицы децибел.

В параграфе введены обозначения, применяемые при рассмотрении процесса декодирования сверточного кода в соответствии с алгоритмом Витерби. Произведен расчет корректирующей способности сверточного кода с использованием программного имитатора кодека, разработанного на языке программирования C++. Рассмотрим более подробно реализацию в программном продукте генерации информационной последовательности и моделирования выхода детектора приемника с заданной вероятностью появления ошибочного бита.



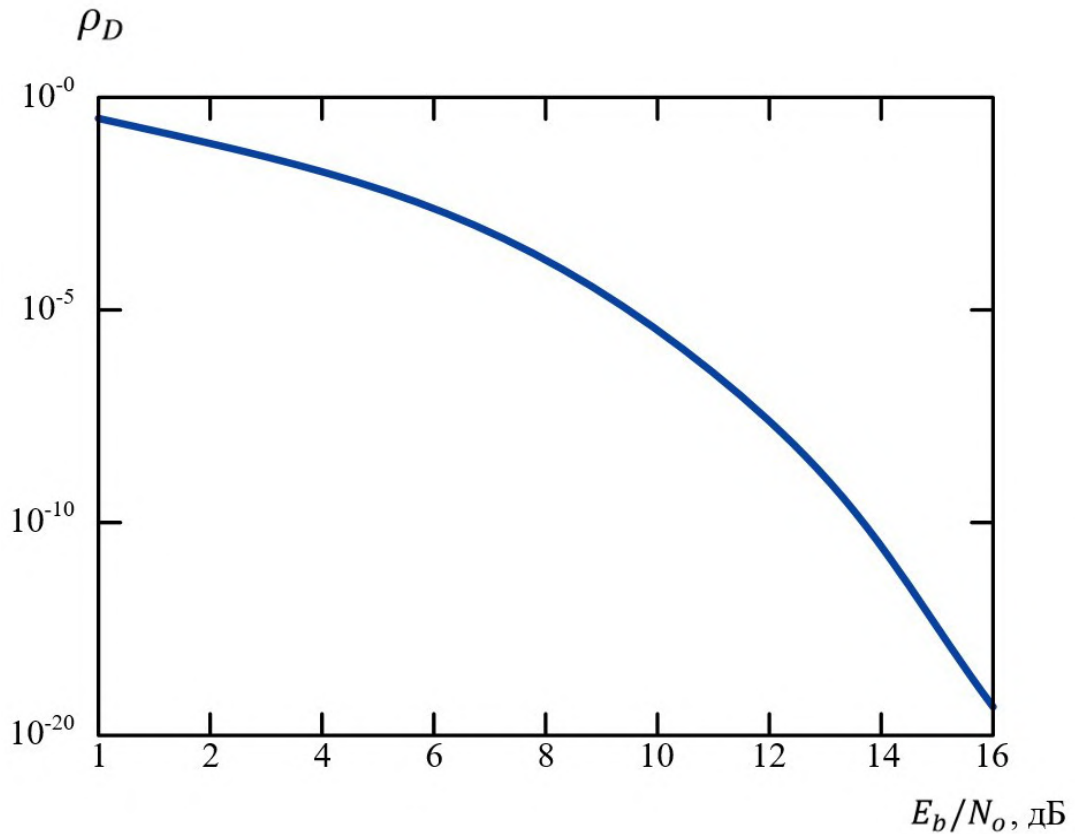


Рисунок 2.8. Характеристика BPSK

## 2.2. Разработка программных генераторов массива информации и выхода детектора

В диссертации рассматриваются результаты работы программных имитаторов, созданных на языках программирования Python [86, 87, 88] и C++ [89, 90, 91, 92], моделирующих работу разнообразных кодеров-декодеров, даются краткие описания таких программных продуктов [93]. Необходимыми компонентами всех разработанных программных имитаторов являются программные генераторы массивов информации и генераторы мягкого или жесткого решения детектора с заданной вероятностью появления ошибочного бита в канале передачи данных. Так как из упоминаемых программных продуктов только один создан на языке программирования C++, более подробно рассмотрим особенности генерации массивов случайных чисел с использованием Python.

Генерация всех массивов случайных чисел на языке программирования Python осуществляется с использованием библиотеки «scipy» и ее субпакета «stats», содержащего статистические функции [94, 95].

Листинг программы, осуществляющей генерацию массива информационных бит, каждый элемент которого это 0 или 1 с одинаковой вероятностью равной 0,5, приведен ниже:

```

infmassiv=[0 for val in range(0, loc)]# массив информации размера loc обнулیم
from scipy import stats as st
chance = 0.5 # вероятность генерации «1»
iter = 1 # количество итераций
count = st.rv_discrete(values=([0,1], [1-chance, chance]))
sample = [count.rvs(size=loc) for val in range(0,iter)]
for value in sample:
infmassiv=value # присваиваем значение очередному элементу массива»

```

В случае исследования декодера, работающего с жестким решением детектора, данный этап может быть реализован путем введения ошибок (инверсии бит) в массив кодовых бит с вероятностью, равной вероятности появления ошибочного бита в канале передачи данных. Для этого вначале генерируется вспомогательный массив двоичных чисел, в котором единичные биты встречаются с вероятностью появления ошибочного бита в канале передачи данных, а остальные биты – нулевые. Размерность вспомогательного массива совпадает с размерностью массива кодовых бит. Далее, вспомогательный массив побитово складывается с массивом кодовых бит по правилу сложения по модулю 2. В результате сложения, в кодовом массиве происходит инверсия бит, номера которых соответствуют номерам единичных бит вспомогательного массива. Таким образом, в кодовом массиве появляются ошибки с заданной вероятностью, моделируя передачу данных в двоичном симметричном канале с аддитивным белым гауссовским шумом. Сформированная таким образом последовательность бит будет являться входной последовательностью для декодера, работающего с

жестким решением детектора. Листинг программы, осуществляющей генерацию массива ошибок с заданной вероятностью, приведен ниже:

```

«errormassiv=[0 for val in range(0, loc)]# массив ошибок размера loc обнуллим
from scipy import stats as st
chancerr = 0.1 # вероятность генерации «1»
iter = 1 # количество итераций
count = st.rv_discrete(values=([0,1], [1-chancerr, chancerr]))
sample = [count.rvs(size=loc) for val in range(0,iter)]
for value in sample:
errormassiv=value # присваиваем значение очередному элементу массива».

```

В п. 1.6. в качестве одного из подходов повышения эффективности помехоустойчивых кодов рассматривалось декодирование мягкого решения детектора, квантованное на более чем два уровня. Далее будут рассмотрены особенности моделирования последовательностей на выходе детектора, квантованных на разное число уровней. Формирование входной последовательности для исследования эффективности программного декодера, работающего с мягким решением детектора, процесс более сложный. При этом, в ходе создания модели выхода детектора, имея сформированную кодовую последовательность, необходимо каждый ее бит с определенной вероятностью заменить символом, которому соответствует группа из  $\log_2 N$  бит, где  $N$  – число уровней квантования. Например, «0» или «1» кодовой последовательности в случае квантования на восемь уровней необходимо заменить на один из восьми символов: «000» («0»), «001» («1»), «010» («2»), «011» («3»), «100» («4»), «101» («5»), «110» («6»), «111» («7»). Вероятность преобразования бита кодовой последовательности в тот или иной символ на выходе детектора зависит от особенностей канала передачи данных и метода квантования.

Для удобства нахождения вероятностей преобразования, примем некоторые допущения. Канал передачи информации является двоичным симметричным каналом. Шум в канале является аддитивным белым гауссовским шумом. Передача бит по каналу ведется противоположными сигналами. Отдельный бит кодового

слова «1» или «0», после прохождения канала без шума и детектирования, принимает аналоговое значение  $r_0$  или  $-r_0$ . Из-за наличия шума в канале прошедший детектор сигнал  $r$  является непрерывной стохастической переменной с плотностями условных вероятностей, равными  $f(r/0)$  и  $f(r/1)$  [96, 97, 98]:

$$f(r/0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r+r_0)^2}{2\sigma^2}\right). \quad (2.3)$$

$$f(r/1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r-r_0)^2}{2\sigma^2}\right). \quad (2.4)$$

Графики плотностей условных вероятностей, построенные по данным формулам, изображены на рисунке 2.9. В случае квантования принятого сигнала на два уровня (случай принятия «жесткого решения») на выходе детектора формируется «1», при попадании сигнала в область справа от точки пересечения графиков плотностей условных вероятностей (в нашем случае в положительную область оси ординат), и формируется «0», при попадании сигнала в область слева от точки пересечения графиков плотностей условных вероятностей (в нашем случае в отрицательную область оси ординат). Вероятность появления ошибочного бита при этом равна вероятности попадания принятого сигнала в положительную область оси ординат, при передаче «0», и в отрицательную область оси ординат, при передаче «1». Очевидно, что в данном случае эти вероятности равны между собой.

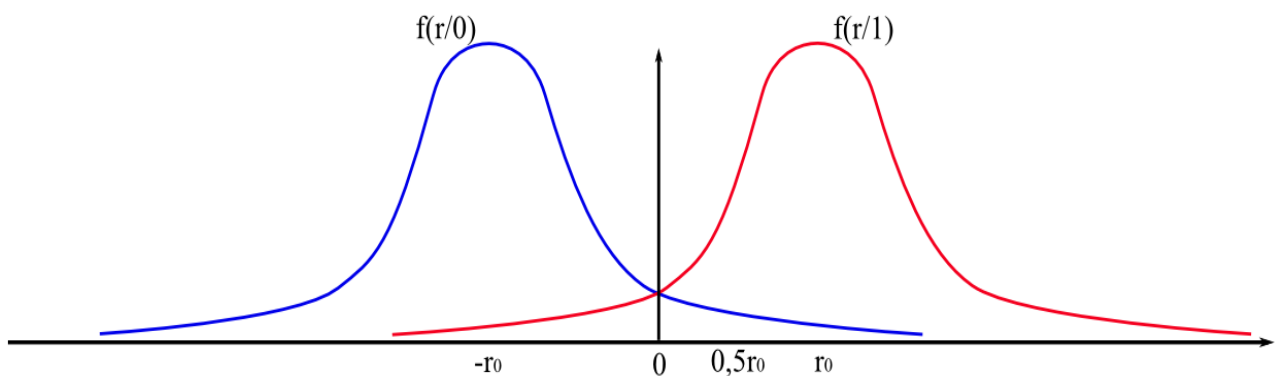


Рисунок 2.9. Плотности условных вероятностей  $f(r/0)$  и  $f(r/1)$ . Квантование на 2 уровня

В случае принятия детектором мягкого решения и квантования принятого сигнала на 4 уровня на выходе детектора, формируется «00» («0») или «01» («1»),

при попадании сигнала в отрицательную область оси ординат, и формируется «10» («2») или «11» («3»), при попадании сигнала в положительную область оси ординат (см. рис. 2.10).

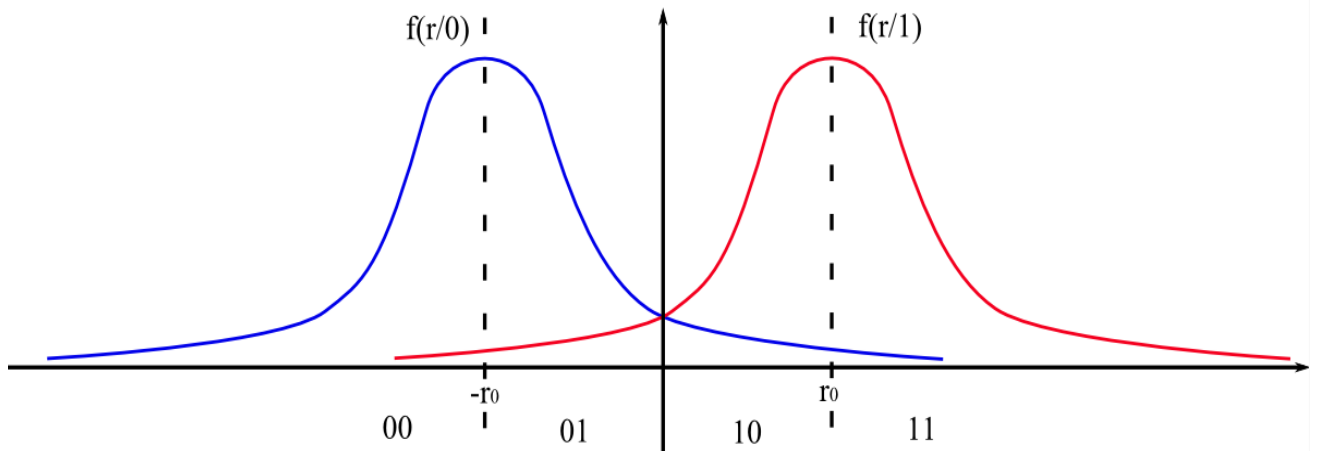


Рисунок 2.10. Плотности условных вероятностей  $f(r/0)$  и  $f(r/1)$ . Квантование на 4 уровня

В случае принятия детектором мягкого решения и квантования принятого сигнала на восемь уровней, на выходе детектора формируется «000» («0»), «001» («1»), «010» («2») или «011» («3»), при попадании сигнала в отрицательную область оси ординат, и формируется «100» («4»), «101» («5»), «110», («6») или «111» («7»), при попадании сигнала в положительную область оси ординат (см. рис. 2.11).

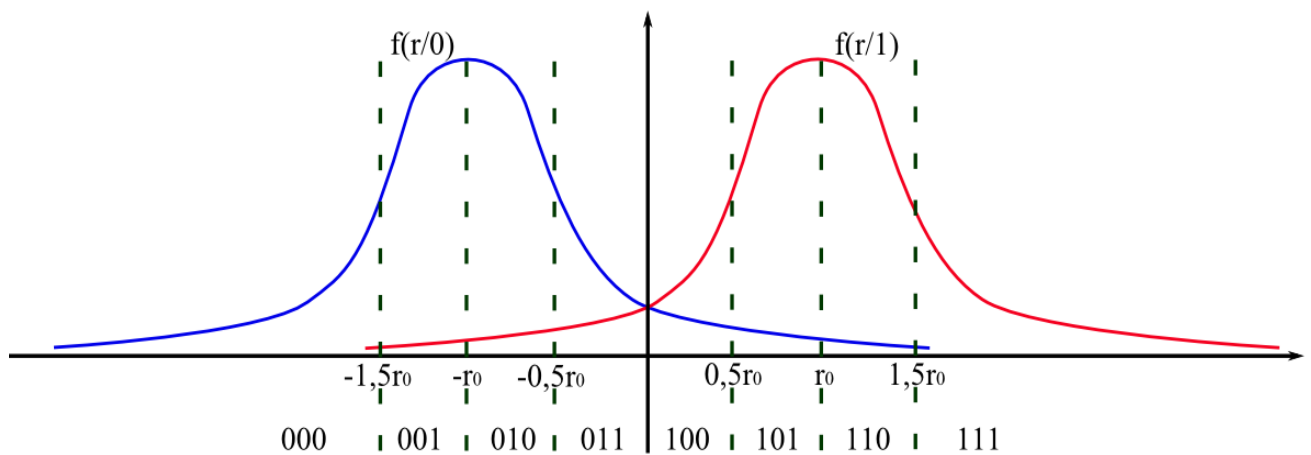


Рисунок 2.11. Плотности условных вероятностей  $f(r/0)$  и  $f(r/1)$ . Квантование на 8 уровней

Правила квантования на разное число уровней, соответствующие рисункам 2.9–2.11 представлены для удобства в табличном виде (см. таблицу 2.2).

Существуют и другие правила квантования, в которых уровни квантования располагаются иначе.

Таблица 2.2

Правила квантования сигнала на 2, 4 и 8 уровней

Число уровней	$<-1,5r_0$	$(-1,5r_0, -r_0)$	$(-r_0, -0,5r_0)$	$(-0,5r_0, 0)$	$(0, 0,5r_0)$	$(0,5r_0, r_0)$	$(r_0, 1,5r_0)$	$>1,5r_0$
2	0				1			
4	0		1		2		3	
8	0	1	2	3	4	5	6	7

При передаче «0» или «1» значение символа на выходе детектора зависит от того, какой области квантования будет соответствовать уровень принятого сигнала, а вероятность попадания сигнала в ту или иную область квантования определяется площадью под графиком функции плотности условных вероятностей на соответствующем участке, и вычисляется через интеграл от плотности условных вероятностей. Интеграл от плотности условных вероятностей для нормального распределения известен как интегральная функция Лапласа (см. рис. 2.12). Значение интегральной функции Лапласа  $F$  в некоторой точке  $A$  равно вероятности того, что случайная величина примет значение от  $-\infty$  до  $A$ .

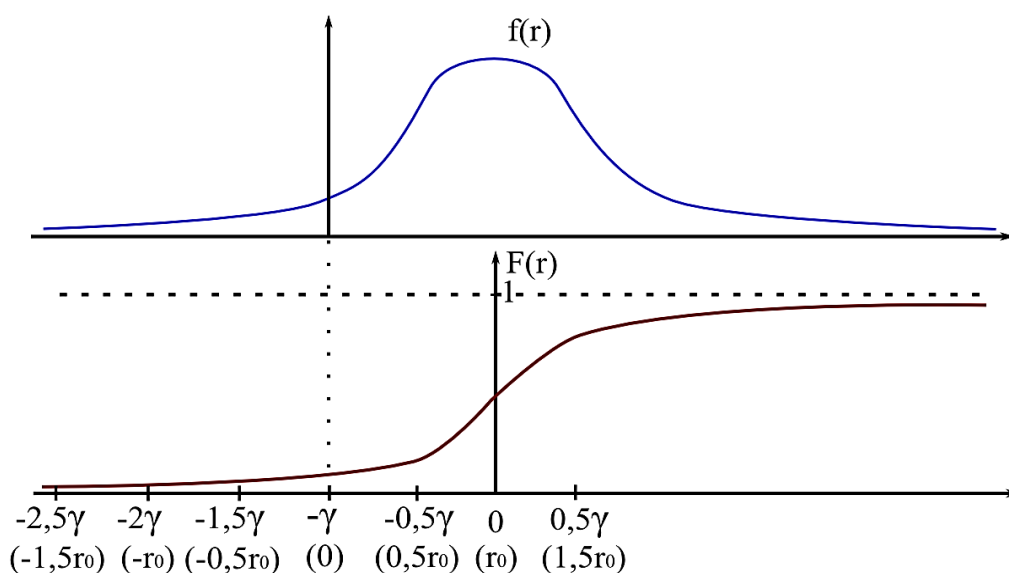


Рисунок 2.12. Нормальное распределение плотности условных вероятностей  $f(r|1)$  (вверху) и интегральная функция Лапласа (внизу)

Интегральная функция Лапласа соответствует нормальному распределению плотностей условных вероятностей, имеющему максимум при нулевом значении переменной. Рассматриваемые распределения плотностей условных вероятностей имеют сдвиг на величину  $\pm r_0$ , который необходимо учитывать при вычислениях. На рисунке 2.12 показано соответствие нормального распределения плотности условных вероятностей  $f(r|1)$  графику интегральной функции Лапласа  $F(\gamma)$ .

Вычислим все значения интегральной функции Лапласа необходимые для нахождения вероятности появления символов на выходе детектора, соответствующие разным значениям вероятности появления ошибочного бита  $p_B$  на выходе детектора, работающего по жесткой схеме принятия решения. Для этого, задав величину  $p_B$ , находим значение  $-\gamma$  такое, чтобы  $F(-\gamma) = p_B$ . Далее, зная значение  $-\gamma$ , находим  $F(-2,5\gamma)$ ,  $F(-2\gamma)$ ,  $F(-1,5\gamma)$ ,  $F(-0,5\gamma)$ ,  $F(0,5\gamma)$  (см. таблицу 2.3). Для нахождения значения интегральной функции Лапласа использовалась встроенная функция Microsoft Excel =НОРМСТРАСП(...).

Обозначим  $P_{XY}$  вероятность появления на выходе детектора символа  $X$  при передаче символа  $Y$ . Тогда из-за симметричности расположения графиков плотностей условных вероятностей (см. рис. 2.10), при квантовании на 4 уровня:  $P_{0/1} = P_{3/0}$ ,  $P_{1/1} = P_{2/0}$ ,  $P_{2/1} = P_{1/0}$ ,  $P_{3/1} = P_{0/0}$ . Аналогичная зависимость будет и при квантовании на другое число уровней.

Таблица 2.3

Значение интегральной функции Лапласа

$F(-\gamma) = p_B$	$-\gamma$	$F(-2,5\gamma)$	$F(-2\gamma)$	$F(-1,5\gamma)$	$F(0)$	$F(-0,5\gamma)$	$F(0,5\gamma)$
$1 \cdot 10^{-1}$	-1,282	$6,78 \cdot 10^{-4}$	0,00519	0,0273	0,5	0,2608	0,739
$1 \cdot 10^{-2}$	-2,325	$3,08 \cdot 10^{-9}$	$1,66 \cdot 10^{-6}$	$2,44 \cdot 10^{-4}$	0,5	0,122	0,877
$1 \cdot 10^{-3}$	-3,09	$5,59 \cdot 10^{-15}$	$3,21 \cdot 10^{-10}$	$1,78 \cdot 10^{-6}$	0,5	0,061	0,938
$1 \cdot 10^{-4}$	-3,718	$7,36 \cdot 10^{-21}$	$5,19 \cdot 10^{-14}$	$1,22 \cdot 10^{-8}$	0,5	0,0315	0,968
$1 \cdot 10^{-5}$	-4,264	$7,83 \cdot 10^{-27}$	$7,44 \cdot 10^{-18}$	$7,98 \cdot 10^{-11}$	0,5	0,0165	0,983

Формулы для вычисления вероятностей появления символов на выходе детектора, с использованием значений интегральной функции Лапласа, для случаев

квантования на четыре и восемь уровней приведены в таблицах 2.4 и 2.5 соответственно.

Таблица 2.4

Вычисление вероятности появления символов на выходе детектора

$P_{0/1}=P_{3/0}$	$P_{1/1}=P_{2/0}$	$P_{2/1}=P_{1/0}$	$P_{3/1}=P_{0/0}$
$F(-2\gamma)$	$F(-\gamma) - F(-2\gamma)$	$0,5 - F(-\gamma)$	$0,5$

Таблица 2.5

Вычисление вероятности появления символов на выходе детектора

$P_{000/1}=P_{111/0}$	$P_{001/1}=P_{110/0}$	$P_{010/1}=P_{101/0}$	$P_{011/1}=P_{100/0}$	$P_{100/1}=P_{011/0}$	$P_{101/1}=P_{010/0}$	$P_{110/1}=P_{001/0}$	$P_{111/1}=P_{000/0}$
$F(-2,5\gamma)$	$F(-2\gamma) - F(-2,5\gamma)$	$F(-1,5\gamma) - F(-2\gamma)$	$F(-\gamma) - F(-1,5\gamma)$	$F(-0,5\gamma) - F(-\gamma)$	$F(0) - F(-0,5\gamma)$	$F(0,5\gamma) - F(0)$	$0,5 - P_{110/1}$

Введем параметр  $h$ , характеризующий качество приема бита. Зададим параметру  $h$  число градаций по количеству уровней квантования. Для квантования на четыре уровня, нулевое значение параметра  $h$  будет соответствовать максимальной ошибке при приеме символа, что соответствует появлению символа «0» при передаче «1» и появлению символа «3» при передаче «0», а значение  $h = 3$  будет соответствовать высокому качеству приема символа, что соответствует появлению символа «3» при передаче «1» и появлению символа «0» при передаче «0».

Рассчитанные, по выше приведенным формулам, вероятности появления символов на выходе детектора, для квантования на четыре и восемь уровней, приведены в таблицах 2.6 и 2.7 соответственно. Аналогично можно произвести расчет вероятности появления символов на выходе детектора, для квантования на другое число уровней.

Рассчитав вероятность появления символов, рассмотрим один из вариантов формирования последовательности на выходе детектора. Вначале формируется вспомогательный массив уровней качества приема бит. Вероятность появления того или иного символа в массиве задается на основании таблиц 2.6 и 2.7 или аналогичной таблицы, рассчитанной для другого числа уровней квантования. Размерность массива уровней качества приема равна размерности массива кодовых



бит. Далее, массив кодовых бит поэлементно складывается с массивом уровней качества, в соответствии с особым правилом сложения, приведенным для случая квантования на четыре уровня и восемь уровней в таблице 2.8. В результате процедуры сложения образуется массив, представляющий собой последовательность символов на выходе детектора.

Таблица 2.6

Вероятность появления символов на выходе детектора при передаче «1» («0»)

СИМВОЛ	0 (3)	1 (2)	2 (1)	3 (0)
$p_B=1 \cdot 10^{-1}$	0,00519	0,09481	0,4	0,5
$p_B=1 \cdot 10^{-2}$	$1,66 \cdot 10^{-6}$	$1 \cdot 10^{-2}$	0,49	0,5
$p_B=1 \cdot 10^{-3}$	$3,21 \cdot 10^{-10}$	$1 \cdot 10^{-3}$	0,499	0,5
$p_B=1 \cdot 10^{-4}$	$5,19 \cdot 10^{-14}$	$1 \cdot 10^{-4}$	0,4999	0,5
$p_B=1 \cdot 10^{-5}$	$7,44 \cdot 10^{-18}$	$1 \cdot 10^{-5}$	0,5	0,5
$h$	0	1	2	3

Таблица 2.7

Вероятность появления символов на выходе детектора при передаче «1» («0»)

СИМВОЛ	«0» («7»)	«1» («6»)	«2» («5»)	«3» («4»)	«4» («3»)	«5» («2»)	«6» («1»)	«7» («0»)
$p_B=1 \cdot 10^{-1}$	$6,78 \cdot 10^{-4}$	0,004512	0,02211	0,0727	0,1608	0,2392	0,239	0,261
$p_B=1 \cdot 10^{-2}$	$3,08 \cdot 10^{-9}$	$1,66 \cdot 10^{-6}$	$2,42 \cdot 10^{-4}$	$0,98 \cdot 10^{-2}$	0,112	0,378	0,377	0,123
$p_B=1 \cdot 10^{-3}$	$5,59 \cdot 10^{-15}$	$3,21 \cdot 10^{-10}$	$1,78 \cdot 10^{-6}$	$1 \cdot 10^{-3}$	0,06	0,44	0,438	0,062
$p_B=1 \cdot 10^{-4}$	$7,36 \cdot 10^{-21}$	$5,19 \cdot 10^{-14}$	$1,22 \cdot 10^{-8}$	$1 \cdot 10^{-4}$	0,0314	0,4686	0,468	0,032
$p_B=1 \cdot 10^{-5}$	$7,83 \cdot 10^{-27}$	$7,44 \cdot 10^{-18}$	$7,98 \cdot 10^{-11}$	$1 \cdot 10^{-5}$	0,0165	0,4835	0,483	0,017
$h$	0	1	2	3	4	5	6	7

Правило сложения

Уровень качества:		0	1	2	3	4	5	6	7
Биты кодового массива (квантование на 4 уровня)	0	3	2	1	0	-	-	-	-
	1	0	1	2	3	-	-	-	-
Биты кодового массива (квантование на 8 уровней)	0	7	6	5	4	3	2	1	0
	1	0	1	2	3	4	5	6	7

Рассмотрим пример сложения массива кодовых бит с массивом уровней качества. Имеется массив кодовых бит 10011010 и вспомогательный массив уровней качества (при  $p_B = 1 \cdot 10^{-2}$ ), сгенерированный в соответствии с данными таблицы 2.6: 23321232. Поэлементное сложение массивов согласно таблице 2.8 даст массив: 20021131.

Ниже приведен фрагмент программы на языке Python, осуществляющий генерацию массива случайных целых чисел значением от 0 до 7 с заданной вероятностью появления каждого числа:

```

«level=[0 for val in range(0, dlin)]# массив информации размера dlin обнулим
from scipy import stats as st
iter = 1 # количество итераций
count=st.rv_discrete(values=(0,1,2,3,4,5,6,7),[0.00069,0.0045,0.02211,0.0727,0.1608,0.239
2,0.239, 0.261]
sample = [count.rvs(size= dlin) for val in range(0,iter)]
for value in sample:
level =value # присваиваем значение очередному элементу массива».

```

В текст программы необходимо ввести значения вероятностей  $P[0] \dots P[7]$  появления элемента массива с значением от 0 до 7. Например, с целью генерации массива уровней качества для приведенного выше примера (квантование на четыре уровня,  $p_B = 1 \cdot 10^{-2}$ ), необходимо, согласно таблице 2.6, задать параметры:  $P[0] = 1,66 \cdot 10^{-6}$ ;  $P[1] = 1 \cdot 10^{-2}$ ;  $P[2] = 0,49$ ;  $P[3] = 0,5$ ;  $P[4] = 0$ ;  $P[5] = 0$ ;  $P[6] = 0$ ;

$P[7] = 0$ . В случае необходимости генерации вспомогательного массива для эмуляции жесткого решения детектора (например, в случае  $p_B = 1 \cdot 10^{-2}$ ), необходимо задать параметры:  $P[0] = 0,99$ ;  $P[1] = 1 \cdot 10^{-2}$ ;  $P[2] = 0$ ;  $P[3] = 0$ ;  $P[4] = 0$ ;  $P[5] = 0$ ;  $P[6] = 0$ ;  $P[7] = 0$ .

На языке программирования C++ генерация массивов случайных чисел реализуется с использованием функции `rand()` [99]. Ниже приведен фрагмент программы на языке C++, осуществляющий генерацию массива случайных целых чисел значением от 0 до 7 с заданной вероятностью появления каждого числа:

```

«int *MC; // массив случайных чисел
int kdb=1000; // задание размерности массива
double P[ARRAYSIZE(samples)];
double lower[ARRAYSIZE(samples)];
double upper[ARRAYSIZE(samples)];
cout << "Vvedi " << ARRAYSIZE(samples) << "Ввод всех вероятностей, в сумме
дающих =1.0" << endl;
double sum = 0;
for (int q = 0; q < ARRAYSIZE(samples); q++) {
cout << "Input P[" << samples[q] << "]" << endl;
cin >> P[q];
lower[q] = sum;
sum += P[q];
upper[q] = sum; }
MC = new int[kdb];
for (int w = 0; w < kdb-4; w++) {
int r = rand();
double d = r / (double)RAND_MAX;
for (int k = 0; k < ARRAYSIZE(samples); k++) {
if (d >= lower[k] && d < upper[k]) {
MC[w] = samples[k];
if (MC[w]<2) cher=cher+1;

```

break;}}}}».

В параграфе были рассмотрены особенности построения программных имитаторов, в частности, подробно описан процесс генерации жесткого и мягкого решения детектора для заданной вероятности появления ошибочного бита в канале передачи данных. В п. 1.6 были описаны преимущества декодирования мягкого решения детектора. Рассмотрим преимущества декодирования мягкого решения детектора в части, касающейся декодирования сверточного кода согласно алгоритму Витерби.

### **2.3. Сравнение корректирующей способности сверточного кода при декодировании мягкого и жесткого решения детектора**

С выхода детектора приемника, принятый сигнал, квантованный на определенное число уровней, поступает на вход декодера. Очевидно, что обработка квантованного на большее число уровней сигнала процесс более сложный и ресурсозатратный, что может быть оправдано только ростом корректирующей способности (снижением числа ошибок в декодированном информационном сообщении). Данный параграф посвящен рассмотрению особенностей декодирования мягкого решения и сравнению корректирующей способности при декодировании мягкого и жесткого решения детектора для сверточного кода с параметрами:  $k = 1$ ,  $n = 2$ ,  $S = 3$ , подробно рассмотренного в п. 2.1. Декодирование мягкого решения рассмотрено во множестве источников [10, 53, 100, 101].

В случае мягкого решения, при квантовании принятого сигнала на четыре уровня, на выходе детектора формируются символы «0» («хороший ноль»), «1» («плохой ноль»), «2» («плохая единица»), «3» («хорошая единица»). Тогда пары символов на выходе детектора имеют значения «00» «01» «02» «03» «10» «11» «12» «13» «14» «20» «21» «22» «23» «30» «31» «32» «33». Рассмотрим одно из возможных правил вычисления метрики. Метрика ветви вычисляется аналогично случаю с жестким решением, но разность символа ветви и символа на входе

декодера находится в соответствии с определенными в таблице 2.9 правилами. Для лучшего понимания, рассмотрим пример вычисления метрики. Пусть значение очередной пары символов, поступивших на вход декодера, «02». Используя таблицу 2.9, рассчитаем метрику ветви «10». Разность символов на входе декодера «0» и символа ветви «1» равна 3. Разность символов на входе декодера «2» и символа ветви «0» равна 2. Тогда метрика ветви равна  $3 + 2 = 5$ .

Таблица 2.9

Вычисление разности символа ветви и символа на входе декодера

Значение символа ветви	Значение символа на входе декодера			
	0	1	2	3
0	0	1	2	3
1	3	2	1	0

В случае мягкого решения детектора, при квантовании принятого сигнала на восемь уровней, на его выходе формируются символы от «0» до «7». Метрика ветви вычисляется аналогично рассмотренному выше. При этом разность символа ветви и символа на входе декодера находится в соответствии с определенными в таблице 2.10 правилами.

В случае жесткого решения максимальное значение метрики ветви равно двум. Как видно из таблицы 2.9, в случае квантования сигнала на выходе детектора на четыре уровня, максимальное значение метрики ветви равно шесть, например, когда имеем пару символов на выходе детектора «30», а значение символов ветви «01». Как видно из таблицы 2.10, в случае квантования сигнала на выходе детектора на восемь уровней, максимальное значение метрики ветви равно четырнадцати.

Таблица 2.10

Вычисление разности символа ветви и символа на входе декодера

Значение символа ветви	Значение символа на входе декодера							
	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	7	6	5	4	3	2	1	0

Рассмотрим декодирование жесткого решения детектора, при возникновении одиночной ошибки в первой паре бит. Схематично процесс декодирования показан на рисунке 2.13. Как было сказано ранее, при отсутствии ошибок метрика истинного пути меньше, чем у ложных путей на 2 и 3. В рассматриваемом случае такая разница суммарных метрик истинного и ложных путей восстанавливается не более чем за 5 шагов декодирования. При этом можно говорить о полном окончании воздействия одиночной ошибки на процесс декодирования.

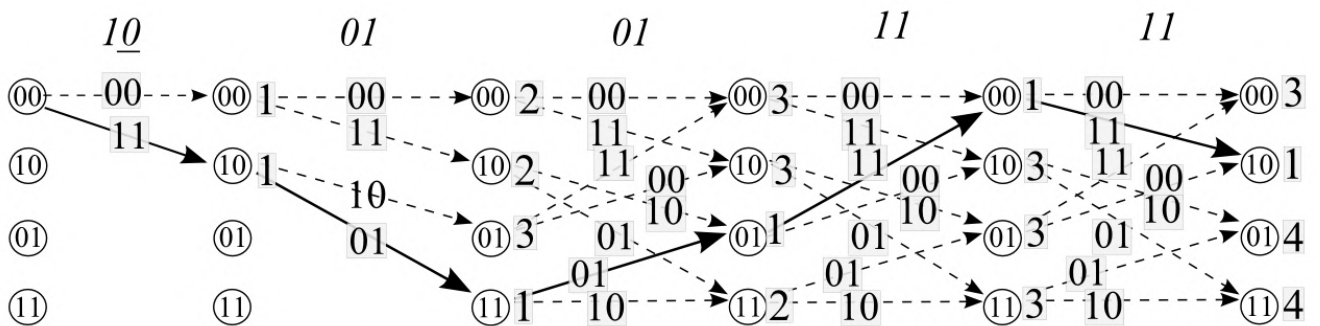


Рисунок 2.13. Декодирование принятой последовательности с одной ошибкой

В случае декодирования мягкого решения, воздействие одиночной ошибки на разницу метрик путей может длиться дольше, чем в случае декодирования жесткого решения, что негативно отражается на успешности декодирования при возникновении последующих ошибок, а также значительно повышает вероятность возникновения ошибки при декодировании конечной последовательности. Сказанное, в первую очередь, относится к наличию грубой ошибки (наличие на выходе детектора «3» при передаче «0», или «0» при передаче «1»). Рассмотрим на примерах декодирования жесткого и мягкого решения детектора последствия возникновения ошибочных бит в конечной последовательности.

В процессе декодирования жесткого решения детектора наличие ошибочного бита с большой вероятностью приводит к ошибке декодирования, как было рассмотрено в п. 2.1.

В случае мягкого решения, наличие грубой ошибки даже в шестой с конца паре символов, может привести к ошибке декодирования (см. рис. 2.14). Следует иметь в виду, что при допущении о том, что шум в канале является аддитивным

белым гауссовским шумом, вероятность грубой ошибки достаточно мала (см. п. 2.2).

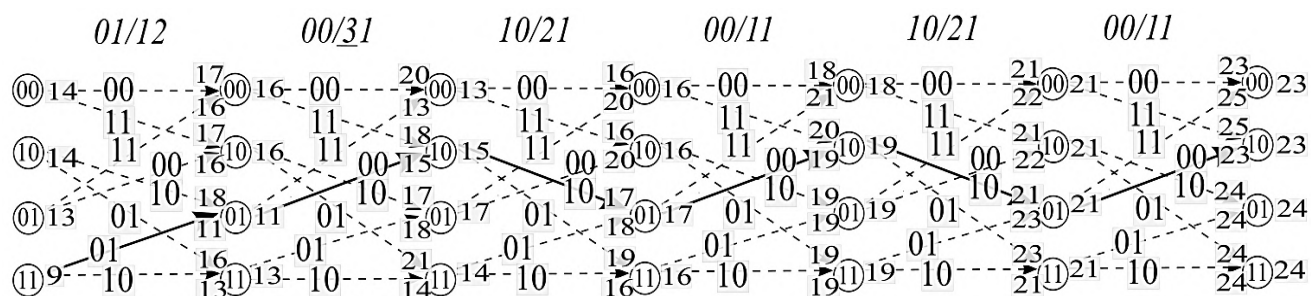


Рисунок 2.14. Декодирование мягкого решения

Декодер сверточного кода эффективно исправляет одиночные ошибки. Декодер, работающий с жестким решением детектора, обеспечивает безошибочное декодирование в случае наличия во входной последовательности двух подряд следующих ошибочных бит, а также при наличии по одному ошибочному биту в каждой из двух подряд следующих пар кодовых бит. Наличие ошибочного бита в каждой из трех подряд следующих пар кодовых бит в принятой последовательности, в большинстве случаев, приводит к ошибке декодирования. Рассмотрим выше сказанное на примере.

Пусть передана кодовая последовательность 11 01 01 00 01 01. Рассмотрим декодирование жесткого и мягкого решения детектора, квантованного на четыре уровня, при наличии ошибок в битах с номерами 2, 3 и 5 (то есть в трех подряд следующих парах кодовых бит). Последовательности представлены в таблице 2.11. Ячейки таблицы, содержащие ошибки, заштрихованы.

Таблица 2.11

Формирование жесткого и мягкого решения детектора

Информационная последовательность	1		1		0		1		1		0	
Кодовая последовательность	1	1	0	1	0	1	0	0	0	1	0	1
Жесткое решение детектора	1	0	1	1	1	1	0	0	0	1	0	1
Мягкое решение детектора	2	1	2	3	2	3	1	0	1	2	0	3

Рассмотрим процесс декодирования жесткого решения, представленного в таблице 2.11. На рисунке 2.15 представлены три итерации процесса декодирования указанной последовательности. Как показано на рисунке, на третьем шаге декодирования метрика истинного пути равна трем, при этом, ложный путь, входящий в один узел с истинным, имеет меньшую метрику (равную двум). В соответствии с алгоритмом декодирования, из двух путей, входящих в один узел, выживает путь, имеющий меньшую метрику. Таким образом, истинный путь будет отброшен, что приведет к ошибке декодирования.

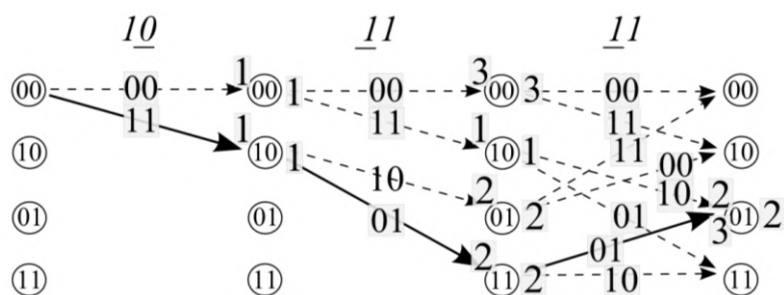


Рисунок 2.15. Декодирование жесткого решения

Процесс декодирования мягкого решения детектора (последовательности из таблицы 2.11), имеющего ошибки в тех же символах  $\underline{21}$   $\underline{23}$   $\underline{23}$   $10$   $12$   $03$ , представлен на рисунке 2.16. После шести шагов декодирования, истинный путь выжил и имеет наименьшую метрику. В данном случае можно наблюдать преимущества мягкого решения.

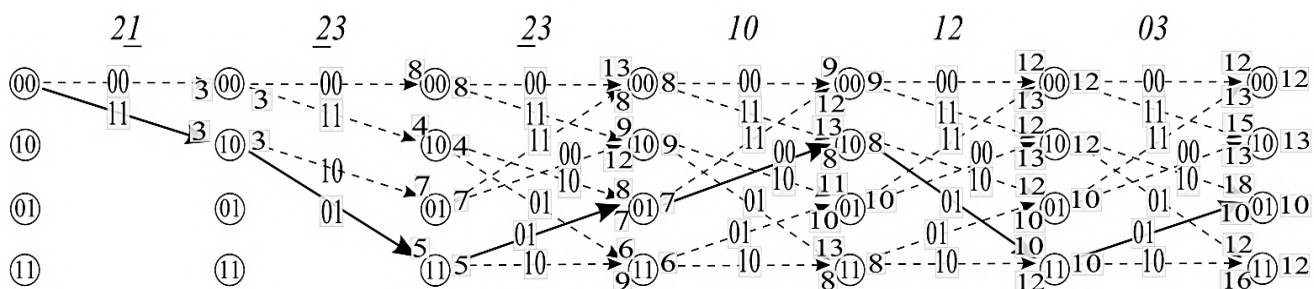


Рисунок 2.16. Декодирование мягкого решения

Наличие трех ошибочных бит подряд может привести к ошибке декодирования как жесткого, так и мягкого решения детектора. Однако, в случае мягкого решения детектора вероятность возникновения ошибки будет меньше. Подтвердим сказанное на примере декодирования последовательности  $11\ 01\ 01\ 00$



01 01, имеющей ошибки в битах с номерами 1, 2 и 3. Тогда, в случае жесткого решения это будет последовательность 00 11 01 00 01 01, а рассматриваемым вариантом мягкого решения будет последовательность 11 23 03 00 12 03. Процесс декодирования жесткого решения показан на рисунке 2.17. На третьем шаге декодирования истинный путь будет отброшен, что приведет к ошибке декодирования. Процесс декодирования мягкого решения показан на рисунке 2.18. После шести шагов декодирования последовательность декодирована без ошибок. Три подряд следующие ошибки исправлены.

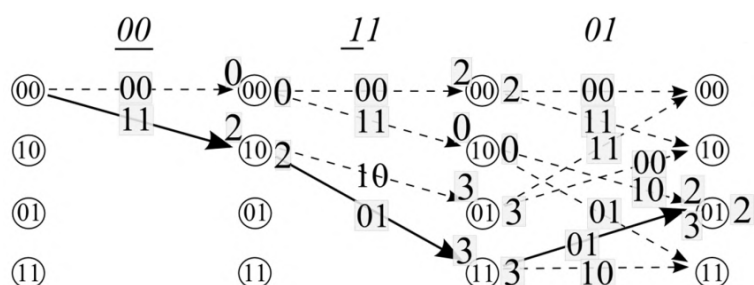


Рисунок 2.17. Декодирование жесткого решения детектора

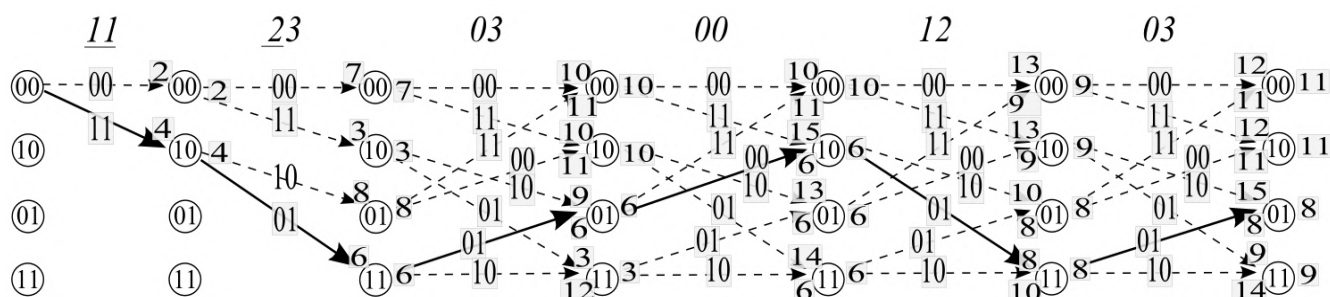


Рисунок 2.18. Декодирование мягкого решения детектора

С использованием описанного в п. 2.1 программного имитатора произведено вычисление вероятности битовой ошибки для сверточного кода при декодировании мягкого решения детектора, квантованного на четыре уровня [102]. В таблице 2.12 приведены данные для сравнительного анализа эффективности декодирования жесткого и мягкого решения при различных значениях вероятности появления ошибочного бита в канале передаче данных. В таблице введены следующие обозначения:  $p_B$  – вероятность появления ошибочного бита в канале передачи данных,  $P_{DHD}$  – вероятность появления ошибочного бита в информационном сообщении в случае декодирования жесткого решения детектора,  $P_{DSD}$  –

вероятность появления ошибочного бита в информационном сообщении в случае декодирования мягкого решения детектора,  $L$  – длина информационного сообщения.

Для удобства восприятия, данные таблицы 2.12 представлены в графическом виде на рисунке 2.19 в части, касающейся результатов декодирования мягкого решения детектора.

Таблица 2.12

Вероятность появления ошибочного бита в информационном сообщении

$p_B$	$L$	$P_{DHD}$	$P_{DSD}$	$P_{DHD}/P_{DSD}$
$1 \cdot 10^{-2}$	4	$4,22 \cdot 10^{-3}$	$3,03 \cdot 10^{-4}$	13,92
	10	$1,42 \cdot 10^{-3}$	$1,02 \cdot 10^{-4}$	13,92
	50	$3,44 \cdot 10^{-4}$	$2,47 \cdot 10^{-5}$	13,89
	100	$1,87 \cdot 10^{-4}$	$1,35 \cdot 10^{-5}$	13,85
	200	$1,34 \cdot 10^{-4}$	$1,29 \cdot 10^{-5}$	9,61
$1 \cdot 10^{-3}$	4	$4,31 \cdot 10^{-4}$	$4,02 \cdot 10^{-5}$	10,7
	10	$1,02 \cdot 10^{-4}$	$1,39 \cdot 10^{-5}$	7,34
	50	$1,90 \cdot 10^{-5}$	$2,60 \cdot 10^{-6}$	7,30
	100	$9,30 \cdot 10^{-6}$	$1,50 \cdot 10^{-6}$	6,20
	200	$6,30 \cdot 10^{-6}$	$1,31 \cdot 10^{-6}$	4,81
$1 \cdot 10^{-4}$	4	$4,90 \cdot 10^{-5}$	$2,03 \cdot 10^{-5}$	2,4
	10	$8,97 \cdot 10^{-6}$	$6,06 \cdot 10^{-6}$	1,48
	50	$1,93 \cdot 10^{-6}$	$1,41 \cdot 10^{-6}$	1,37
	100	$8,60 \cdot 10^{-7}$	$5,99 \cdot 10^{-7}$	1,40
	200	$3,80 \cdot 10^{-7}$	$2,89 \cdot 10^{-7}$	1,35

При декодировании мягкого решения детектора вероятность битовой ошибки в декодированном информационном сообщении снижается. Снижение особенно велико при высоких значениях вероятности битовой ошибки в канале и для сообщений небольшой длины. При  $p_B = 1 \cdot 10^{-2}$  и длине информационного сообщения равной четырем битам, получено снижение вероятности битовой ошибки в декодированном информационном сообщении в 14 раз. При снижении вероятности битовой ошибки в канале передачи данных, выигрыш от декодирования мягкого решения детектора снижается.

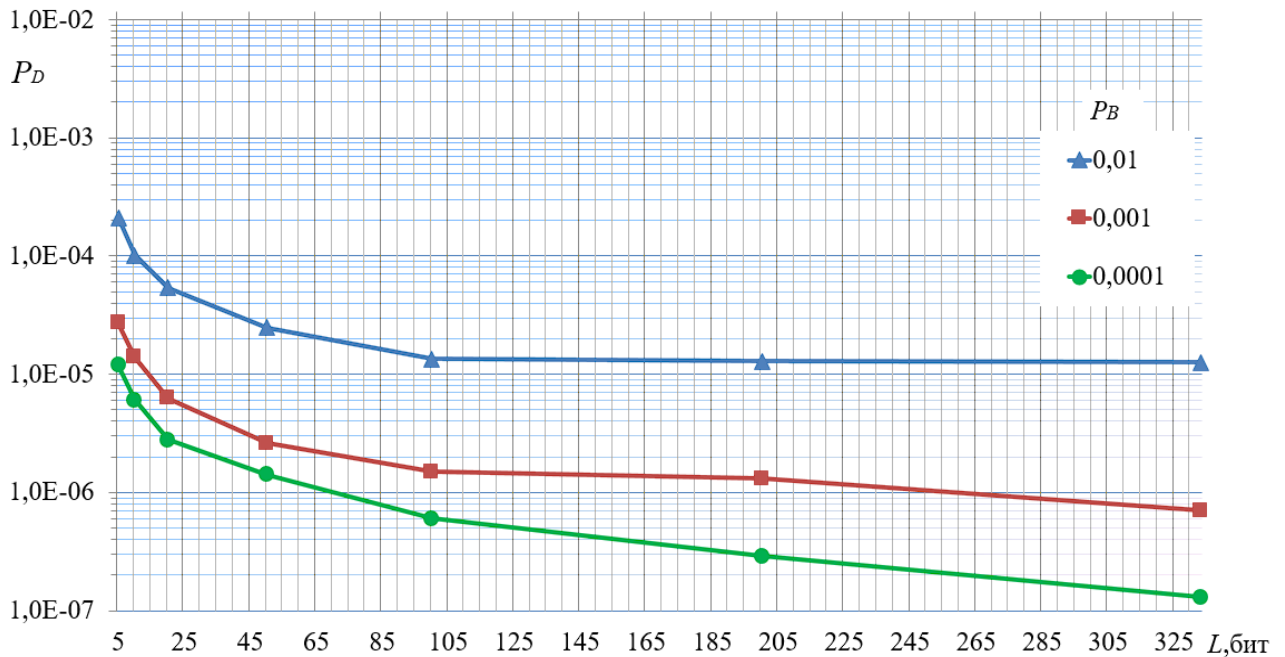


Рисунок 2.19. Зависимость вероятности ошибочного бита от длины информационного сообщения при декодировании мягкого решения детектора

В параграфе произведено сравнение корректирующей способности сверточных кодов при декодировании мягкого и жесткого решения детектора. Корректирующая способность сверточного кода значительно возрастает при декодировании мягкого решения детектора. Помимо корректирующей способности, важной характеристикой помехоустойчивого кода является его способность к адаптации параметров при изменении характеристик канала передачи данных. Наиболее простым способом адаптации является перфорация кодовой последовательности. Рассмотрим применение перфорации в сверточных кодах.

#### 2.4. Исследование механизма декодирования перфорированных сверточных кодов

О важности перфорации как средства повышения эффективности помехоустойчивых кодов было сказано в п. 1.6. Описание перфорированных сверточных кодов приведено во множестве источников [10, 12, 54, 103, 104].

Перфорированные сверточные коды применяются в разнообразных стандартах связи, например, в IEEE 802.11a (Wi-Fi) и IEEE 802.16e (WiMAX).

Возможны несколько вариантов декодирования кодовой последовательности, содержащей выколотые символы. Применительно к выбранному коду, рассмотрим два возможных варианта. В первом варианте при декодировании выколотые биты исключаются из процесса расчета метрики ветви. Во втором варианте выколотые биты участвуют в процессе расчета метрики ветви, для чего им присваивается численное значение среднее между значением, детектированного как «0» бита, и значением детектированного как «1» бита.

Для реализации первого варианта, в алгоритме декодирования необходимо установить запрет на участие стертых символов в вычислении метрики ветви.

Рассмотрим изменения, которые необходимо внести в алгоритм декодирования для реализации второго варианта. В случае жесткого решения детектора, стертому биту присваивается значение «1». Биту, детектированному как «0» присваивается значение «0». Биту, детектированному как «1» присваивается значение «2». Далее задается правило вычисления вклада значения декодируемого символа в значение слагаемого метрики ветви (см. таблицу 2.13) или правило вычисления значения метрики ветви (см. таблицу 2.14). При изменении алгоритма выкалывания бит, правила вычисления метрики остаются неизменными. Случай мягкого решения детектора рассмотрим на примере решения, квантованного на четыре уровня. При этом, стертому биту присваивается значение «2». Биту, детектированному как «0» присваивается значение «0» или «1». Биту, детектированному как «1» присваивается значение «3» или «4».

Таблица 2.13

Расчет вклада символа в значение метрики ветви

Значение символа на входе декодера		0	1	2
Значение символа ветви	0	0	1	2
	1	2	1	0

Расчет значения метрики ветви

Вход декодера:		00	01	02	10	11	12	20	21	22
Значение ветви	00	0	1	2	1	2	3	2	3	4
	01	2	1	0	3	2	1	4	3	2
	10	2	3	4	1	2	3	0	1	2
	11	4	3	2	3	2	1	2	1	0

Рассмотрим примеры декодирования двумя вариантами рассмотренной ранее кодовой последовательности 11 01 01 00 01 01 при выкалывании каждого второго бита и отсутствии ошибок. В случае первого варианта, последовательность на входе декодера имеет вид 1X 0X 0X 0X 0X 0X. Процесс декодирования показан на рисунке 2.20. По итогам декодирования, финальный выживший путь соответствует истинному пути. В случае второго варианта, последовательность на входе декодера имеет вид 21 01 01 01 01 01. Процесс декодирования показан на рисунке 2.21. Символы, поступившие на вход декодера разбиты на пары в соответствии с итерациями декодирования, и приведены вверху схемы. По итогам декодирования, финальный выживший путь соответствует истинному пути.

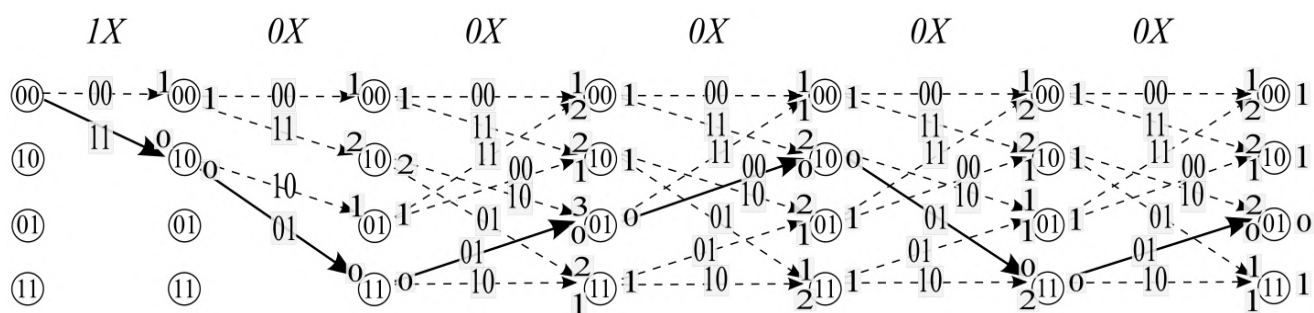


Рисунок 2.20. Первый вариант декодирования

Произведем сравнение рассмотренных примеров декодирования. В обоих примерах декодирование произведено успешно. Полученная разница метрики истинной ветви и других ветвей в обоих вариантах соответствует величине метрики одного ошибочного бита. В первом варианте выколотые биты не участвуют в вычислении метрики, и, следовательно, вычислительные затраты снижены в два

раза, по сравнению со вторым вариантом. В дальнейших рассмотренных примерах декодирование будет производиться согласно первому варианту.

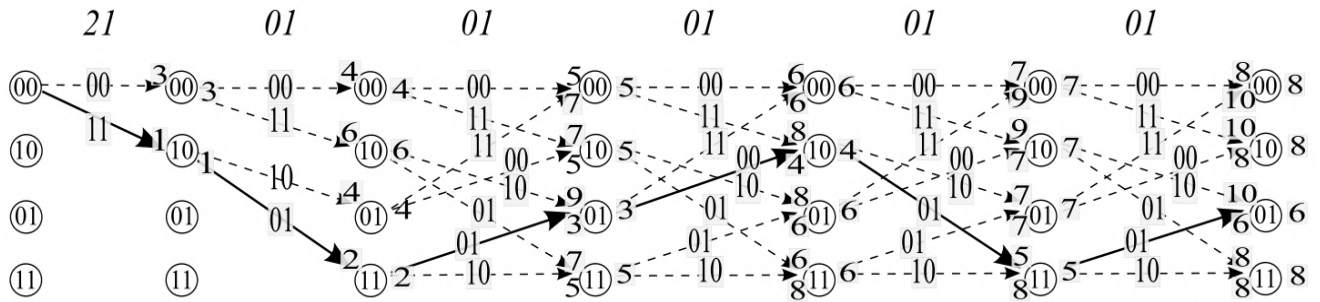


Рисунок 2.21. Второй вариант декодирования

Очевидно, что выкалывание бит отрицательно сказывается на достоверности передачи данных и ведет к росту числа ошибок декодирования. Логично предположить, что с увеличением числа выкалываний эффективность декодирования снижается. При большом числе выколотых бит между кодовыми словами снижается расстояние Хемминга ( $d_{min}$ ), из-за чего вероятность появления ошибочного бита в декодированном информационном сообщении растет [11].

Для исследования корректирующей способности перфорированного кода был использован описанный в п. 2.1 программный имитатор. В таблице 2.15 приведены полученные в результате работы программы значения вероятности появления ошибочного бита в декодированном информационном сообщении при использовании перфорации. Результаты расчетов подтверждают высказанное предположение как глобальную тенденцию, однако, в некоторых частных случаях противоречат ему, что более наглядно продемонстрировано на диаграмме, построенной на основании данных из таблицы 2.15, и иллюстрирующей зависимость  $p_D$  от скорости кода  $R$  (см. рис. 2.22). Отсутствие монотонного характера роста числа ошибок с увеличением числа выколотых бит объясняется тем, что для разных типов кодов существуют более или менее эффективные алгоритмы выкалывания. Для наиболее известных видов сверточных кодов алгоритмы выкалывания были определены на основании результатов компьютерного поиска [10].

Вероятность появления ошибочного бита в декодированном информационном сообщении

№ п/п	Вид выкалывания	$R$	Вероятность появления ошибочного бита в канале передачи данных ( $p_B$ )		
			$p_B = 10^{-2}$	$p_B = 10^{-3}$	$p_B = 10^{-4}$
1.	Без выкалывания	0,5	$4,75 \cdot 10^{-5}$	$5,20 \cdot 10^{-8}$	$6,42 \cdot 10^{-11}$
2.	Каждый 24-ый	0,52	$1,26 \cdot 10^{-4}$	$7,71 \cdot 10^{-7}$	$8,52 \cdot 10^{-9}$
3.	Каждый 12-ый	0,55	$2,45 \cdot 10^{-4}$	$1,34 \cdot 10^{-6}$	$1,52 \cdot 10^{-8}$
4.	Каждый 8-ой	0,57	$5,10 \cdot 10^{-4}$	$3,20 \cdot 10^{-6}$	$2,51 \cdot 10^{-8}$
5.	Каждый 6-ой	0,6	$1,12 \cdot 10^{-3}$	$1,11 \cdot 10^{-5}$	$1,11 \cdot 10^{-7}$
6.	Каждый 5-ый	0,63	$8,02 \cdot 10^{-4}$	$7,20 \cdot 10^{-6}$	$5,27 \cdot 10^{-8}$
7.	Каждый 4-ый	0,67	$1,54 \cdot 10^{-1}$	$7,70 \cdot 10^{-3}$	$9,28 \cdot 10^{-5}$
8.	Каждый 3-ий	0,75	$5,38 \cdot 10^{-3}$	$6,22 \cdot 10^{-5}$	$6,47 \cdot 10^{-7}$

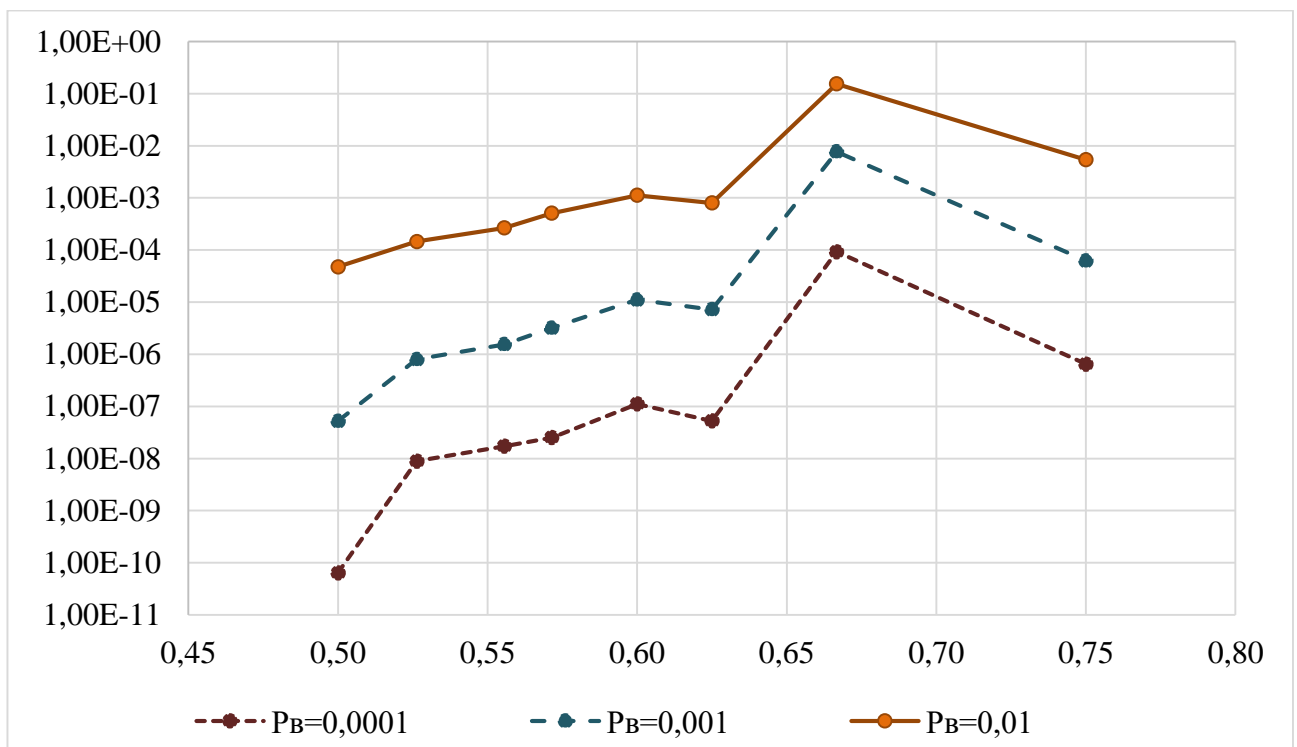


Рисунок 2.22. Зависимость  $p_D$  от скорости кода

Выше был рассмотрен пример декодирования двумя вариантами кодовой последовательности при выкалывании каждого второго бита и отсутствии ошибок. При этом скорость кода составила 1. Такое значение скорости кода является предельной величиной и не позволяет осуществлять исправление ошибок, что не

имеет смысла при практическом применении. Подтвердим сказанное, рассмотрев пример воздействия одиночной ошибки на приведенную выше кодовую последовательность. На рисунке 2.23 схематично показан процесс декодирования принятой последовательности. По итогам последней итерации декодирования финальный выживший путь (имеющий минимальную метрику) отличен от истинного пути, что приведет к ошибкам в декодировании информационной последовательности.

При применении выкалывания бит, приводящего к росту скорости кода до значений меньше 1, исправление ошибок становится возможным. В качестве примера, рассмотрим процесс декодирования кодовой последовательности 11 01 01 00 01 01 11 (соответствующая информационная последовательность: 1101100) при наличии ошибки в пятом бите и выкалывании каждого шестого бита. Тогда последовательность на входе декодера будет выглядеть следующим образом: 11 01 1X 00 01 0X 11. Процесс декодирования схематично представлен на рисунке 2.24. По итогам четвертой (с момента появления ошибочного бита) итерации декодирования истинный путь имеет минимальную метрику, что делает возможным исправление ошибочного бита.

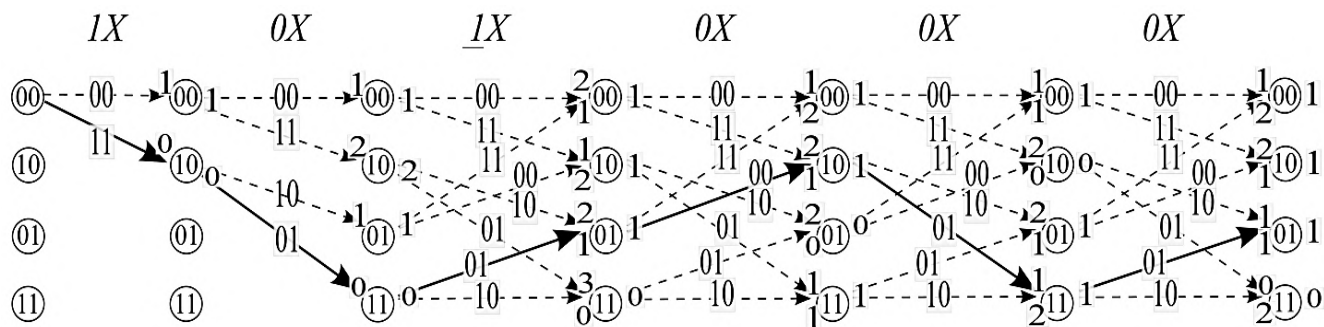


Рисунок 2.23. Декодирование при наличии ошибочного бита и скорости кода 1

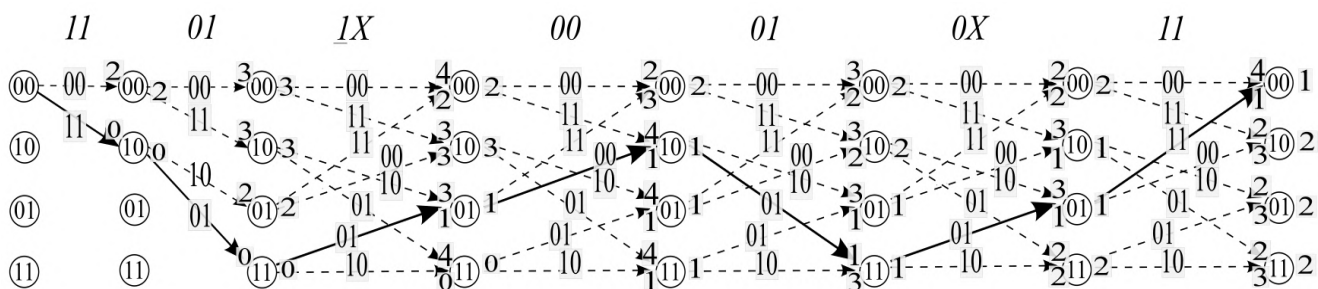


Рисунок 2.24. Декодирование при наличии ошибочного бита и скорости кода 0,6



При реализации перфорированных кодов, декодеру потребуется определить тип алгоритма перфорации. Это может быть осуществлено передачей служебных символов перед пакетом данных, или подсчетом числа бит в пакете. Для некоторых алгоритмов перфорации может потребоваться дополнительная синхронизация принятых данных по длительности цикла перфорации.

В параграфе рассмотрены механизмы декодирования перфорированной последовательности, закодированной сверточным кодом. Далее произведем исследование корректирующей способности сверточного кода в зависимости от числа выколотых бит при декодировании мягкого и жесткого решения детектора.

## **2.5. Исследование корректирующей способности перфорированного сверточного кода**

Проведем исследование эффективности декодирования мягких и жестких решений детектора в перфорированных сверточных кодах [104, 105].

Правило выкалывания бит может задаваться в виде матрицы перфорации [10, 106] и действует циклично. Число строк матрицы перфорации равно числу выходов кодера, каждый столбец соответствует очередной итерации кодирования. Примеры матриц перфорации для рассматриваемого сверточного кода приведены в таблице 2.16. Поясним работу алгоритма выкалывания на примере кодовой последовательности 11 01 01 00 01 01 для матрицы перфорации № 2 из таблицы 2.16. Если значение элемента матрицы равно «1», то символ будет передан, если «0», то нет. После завершения процедуры выкалывания получаем последовательность 11 0X 01 0X 01 0X, из которой символы, помеченные X, передаваться не будут, то есть будет передана последовательность 110010010. Для удобства, далее алгоритм выкалывания согласно матрице перфорации №  $i$  будем именовать – «алгоритм  $i$ ».

В качестве примера, рассмотрим процесс декодирования согласно алгоритму Витерби кодовой последовательности 11010100010111001110111101100100

(которая соответствует информационной последовательности 1101100010011101) при применении алгоритма 5, при наличии ошибочного бита, для случаев жесткого решения детектора и мягкого решения детектора, квантованного на 4 уровня.

Таблица 2.16

Матрицы перфорации стандартного кода скорости 1/2

№	Матрица перфорации	Скорость кодирования
1.	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	4/7
2.	$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	2/3
3.	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	3/4
4.	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$	4/5
5.	$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$	5/6
6.	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$	7/8

Рассмотрим декодирование жесткого решения детектора. С учетом шаблона выкалывания и наличия ошибочного бита, на вход декодера поступает последовательность 110XX00XX1011XX01XX0111XX11XX100. Ошибочный бит выделен подчеркиванием. Процесс декодирования схематично показан на рисунке 2.25. По итогам тринадцатой (с момента появления ошибочного бита) итерации декодирования истинный путь имеет минимальную метрику, что делает возможным исправление ошибочного бита.

Рассмотрим декодирование мягкого решения детектора. С учетом шаблона выкалывания и наличия ошибочного бита, на вход декодера поступает последовательность 231XX10XX2033XX12XX032... соответствующая мягкому решению детектора, квантованному на 4 уровня (от 0 до 3). Ошибочный бит выделен подчеркиванием. Процесс декодирования схематично показан на рисунке 2.26. По итогам восьмой (с момента появления ошибочного бита) итерации

декодирования истинный путь имеет минимальную метрику, что делает возможным исправление ошибочного бита.

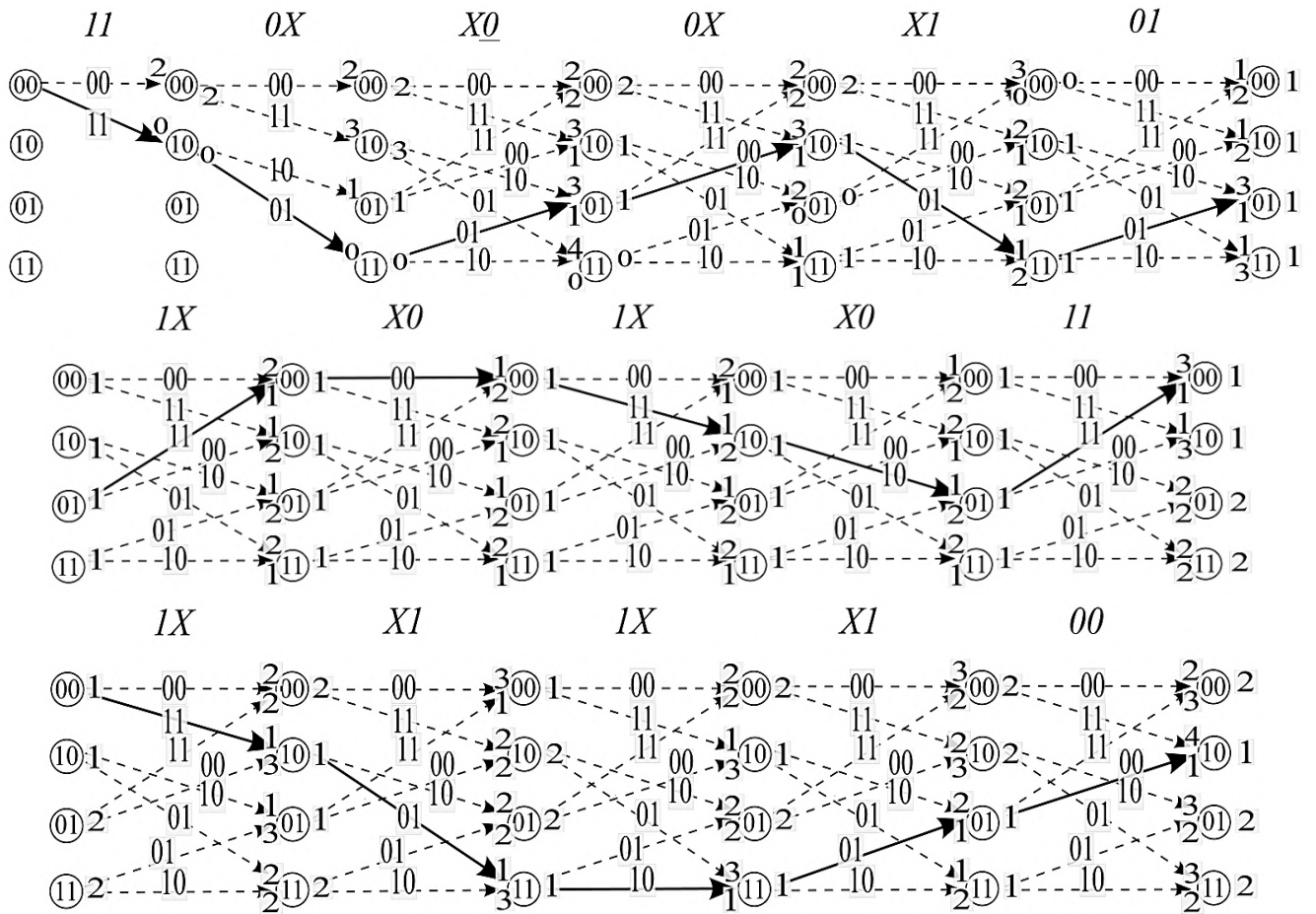


Рисунок 2.25. Декодирование жесткого решения детектора

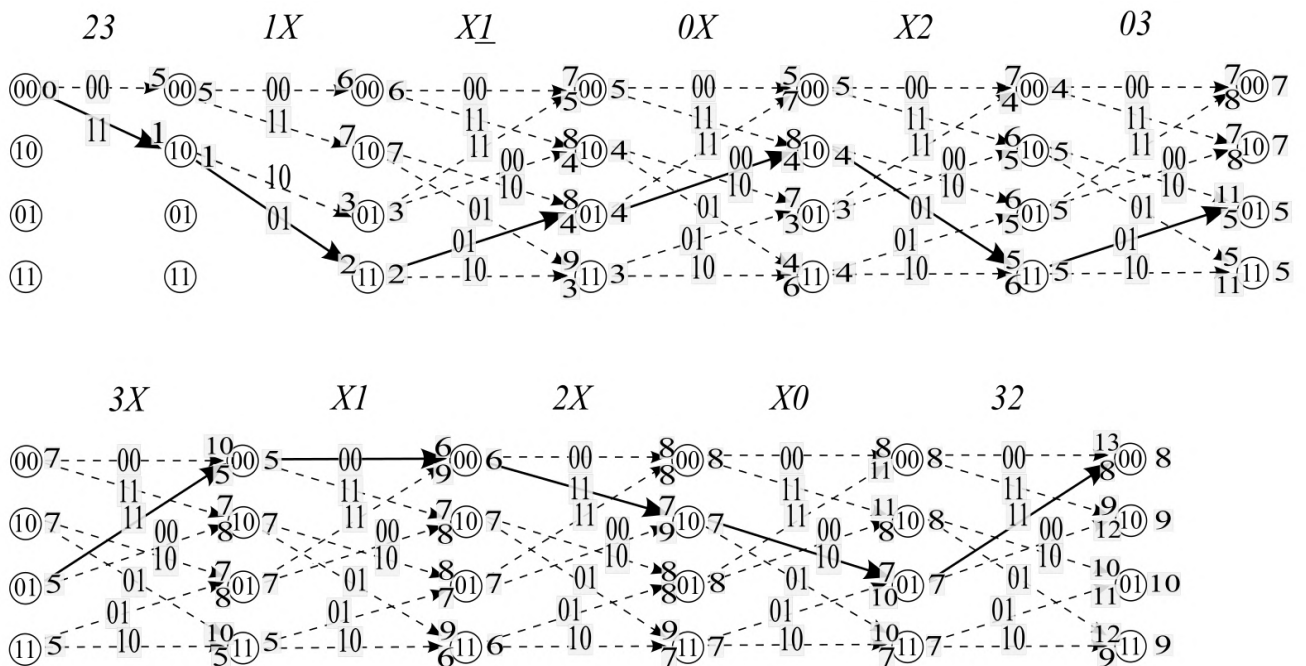


Рисунок 2.26. Декодирование мягкого решения детектора

Рассмотренные примеры декодирования с выкалываниями показали большую эффективность случая декодирования мягкого решения детектора (возникшая ошибка исправлена на 5 тактов раньше). Для случая применения сверточного кода без выкалываний, декодирование мягкого решения детектора также более эффективно [10]. Для исследования эффективности использования разных алгоритмов выкалывания был разработан программный имитатор, состоящий из кодера, блока генерации ошибок и декодера. В таблице 2.17 приведены полученные в результате работы программного имитатора значения вероятности появления ошибочного бита в декодированном информационном сообщении при использовании различных алгоритмов выкалывания и разных значениях вероятности появления ошибочного бита в канале передачи данных. Применение выкалывания может быть более или менее целесообразно до тех пор, пока в результате декодирования происходит снижение вероятности появления ошибочного бита. То есть до тех пор, пока  $p_D < p_B$ . Для удобства восприятия, в таблице заштрихованы ячейки для случаев, когда  $p_D > p_B$ . При перфорации возможно изменение кодовой скорости с 0,5 до 0,8 (то есть на 60 %), при декодировании жесткого решения детектора, и с 0,5 до 0,83 (то есть на 66 %), при декодировании мягкого решения детектора.

При отсутствии перфорации каждый помехоустойчивый код имеет определенные конструктивно заложенные избыточность и корректирующую способность. Сравним корректирующую способность перфорированных сверточных кодов (случай декодирования мягкого решения детектора) и блочных кодов Хемминга при условии равенства скорости кодирования сравниваемых кодов. Описание кода Хемминга дано в п. 3.1.1. Корректирующая способность здесь и далее будет характеризоваться величиной значения вероятности битовой ошибки в декодированном информационном сообщении  $p_D$ . Для удобства сравнения полученных значений  $p_D$  для перфорированного сверточного кода ( $P_{DCK}$ ) и кода Хемминга ( $P_{DX}$ ) произведем расчет различия эффективности декодирования в децибелах ( $K_{\Delta}$ ), вычислив значение  $K_{\Delta} = |10 \cdot \log(P_{DCK}/P_{DX})|$ . В таблице 2.18

приведены параметры сверточных кодов и номера соответствующих им по скорости кода алгоритмов выкалывания.

Таблица 2.17

Вероятность появления ошибочного бита в декодированном информационном сообщении

№ алгоритма выкалывания по таблице 2.16	Вероятности появления ошибочного бита в канале передачи данных					
	$p_B = 1 \cdot 10^{-2}$		$p_B = 1 \cdot 10^{-3}$		$p_B = 1 \cdot 10^{-4}$	
	жесткое решение	мягкое решение	жесткое решение	мягкое решение	жесткое решение	мягкое решение
Без выкалывания	$4,75 \cdot 10^{-5}$	$1,26 \cdot 10^{-5}$	$5,20 \cdot 10^{-8}$	$4,82 \cdot 10^{-9}$	$6,42 \cdot 10^{-11}$	$7,40 \cdot 10^{-13}$
1.	$5,10 \cdot 10^{-4}$	$1,03 \cdot 10^{-4}$	$3,20 \cdot 10^{-6}$	$7,20 \cdot 10^{-7}$	$2,51 \cdot 10^{-8}$	$4,62 \cdot 10^{-9}$
2.	$1,49 \cdot 10^{-3}$	$5,20 \cdot 10^{-4}$	$1,29 \cdot 10^{-5}$	$2,40 \cdot 10^{-6}$	$1,00 \cdot 10^{-7}$	$2,00 \cdot 10^{-8}$
3.	$4,80 \cdot 10^{-3}$	$2,70 \cdot 10^{-3}$	$4,84 \cdot 10^{-5}$	$2,51 \cdot 10^{-5}$	$4,00 \cdot 10^{-7}$	$2,44 \cdot 10^{-7}$
4.	$1,52 \cdot 10^{-2}$	$9,22 \cdot 10^{-3}$	$8,74 \cdot 10^{-4}$	$5,16 \cdot 10^{-4}$	$9,20 \cdot 10^{-5}$	$5,00 \cdot 10^{-5}$
5.	$2,03 \cdot 10^{-2}$	$1,20 \cdot 10^{-2}$	$1,30 \cdot 10^{-3}$	$8,84 \cdot 10^{-4}$	$1,44 \cdot 10^{-4}$	$8,08 \cdot 10^{-5}$
6.	$2,10 \cdot 10^{-1}$	$2,10 \cdot 10^{-1}$	$3,80 \cdot 10^{-2}$	$3,73 \cdot 10^{-2}$	$4,01 \cdot 10^{-3}$	$3,91 \cdot 10^{-3}$

Произведем сравнение эффективности декодирования сверточного кода с выкалываниями согласно алгоритму 1 и кода Хемминга с параметрами:  $k = 4$ ,  $n = 7$ . Данные для сравнения приведены в таблице 2.19 и на рисунке 2.27. Значения вероятности появления ошибочного бита в декодированном сообщении для кода Хемминга взяты из главы 3. Перфорированный сверточный код показал большую эффективность декодирования при всех значениях  $p_B$ . С ростом  $p_B$  наблюдается снижение величины различия эффективности.

Таблица 2.18

Кодовые скорости перфорированного сверточного кода и кодов Хемминга

№п/п	Параметры сверточного кода		Параметры кода Хемминга		
	№ алгоритма	Кодовая скорость	$k$	$n$	Кодовая скорость
1.	1.	4/7	4	7	4/7
2.	3.	3/4	11	15	11/15
3.	5.	5/6	26	31	26/31

Таблица 2.19

Вероятность появления ошибочного бита в декодированном сообщении

	$p_B = 1 \cdot 10^{-2}$	$p_B = 1 \cdot 10^{-3}$	$p_B = 1 \cdot 10^{-4}$
$p_D$ , сверточный код	$1,0 \cdot 10^{-4}$	$7,2 \cdot 10^{-7}$	$4,6 \cdot 10^{-9}$
$p_D$ , код Хемминга ( $k=4, n=7$ )	$8,6 \cdot 10^{-4}$	$8,6 \cdot 10^{-6}$	$8,6 \cdot 10^{-8}$
$K_э$ , дБ	9,3	10,1	12,7

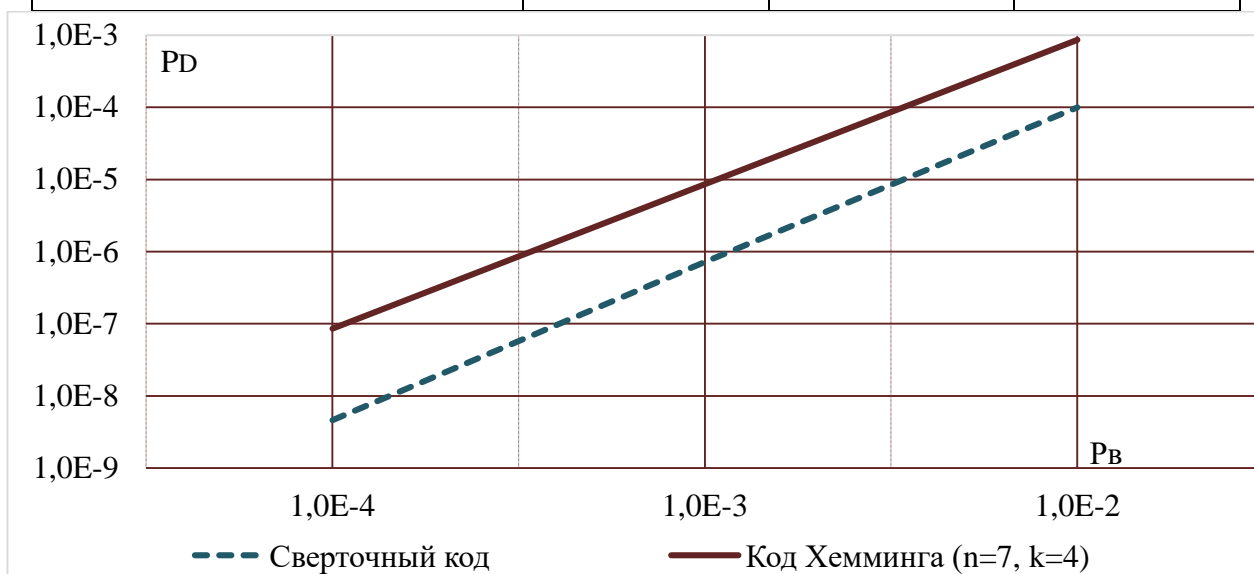


Рисунок 2.27. Сравнение эффективности перфорированного сверточного кода и кода Хемминга

Произведем сравнение эффективности декодирования сверточного кода с выкалываниями согласно алгоритму 3 и кода Хемминга с параметрами:  $k = 11$ ,  $n = 15$ . Данные для сравнения приведены в таблице 2.20 и на рисунке 2.28. Перфорированный сверточный код показал несколько меньшую эффективность декодирования при всех значениях  $p_B$ . С ростом  $p_B$  наблюдается незначительный рост величины различия эффективности.

Таблица 2.20

Вероятность появления ошибочного бита в декодированном сообщении

	$p_B = 1 \cdot 10^{-2}$	$p_B = 1 \cdot 10^{-3}$	$p_B = 1 \cdot 10^{-4}$
$p_D$ , сверточный код	$2,7 \cdot 10^{-3}$	$2,5 \cdot 10^{-5}$	$2,4 \cdot 10^{-7}$
$p_D$ , код Хемминга ( $k = 11, n = 15$ )	$2,0 \cdot 10^{-3}$	$2,0 \cdot 10^{-5}$	$2,0 \cdot 10^{-7}$
$K_э$ , дБ	1,3	0,97	0,79

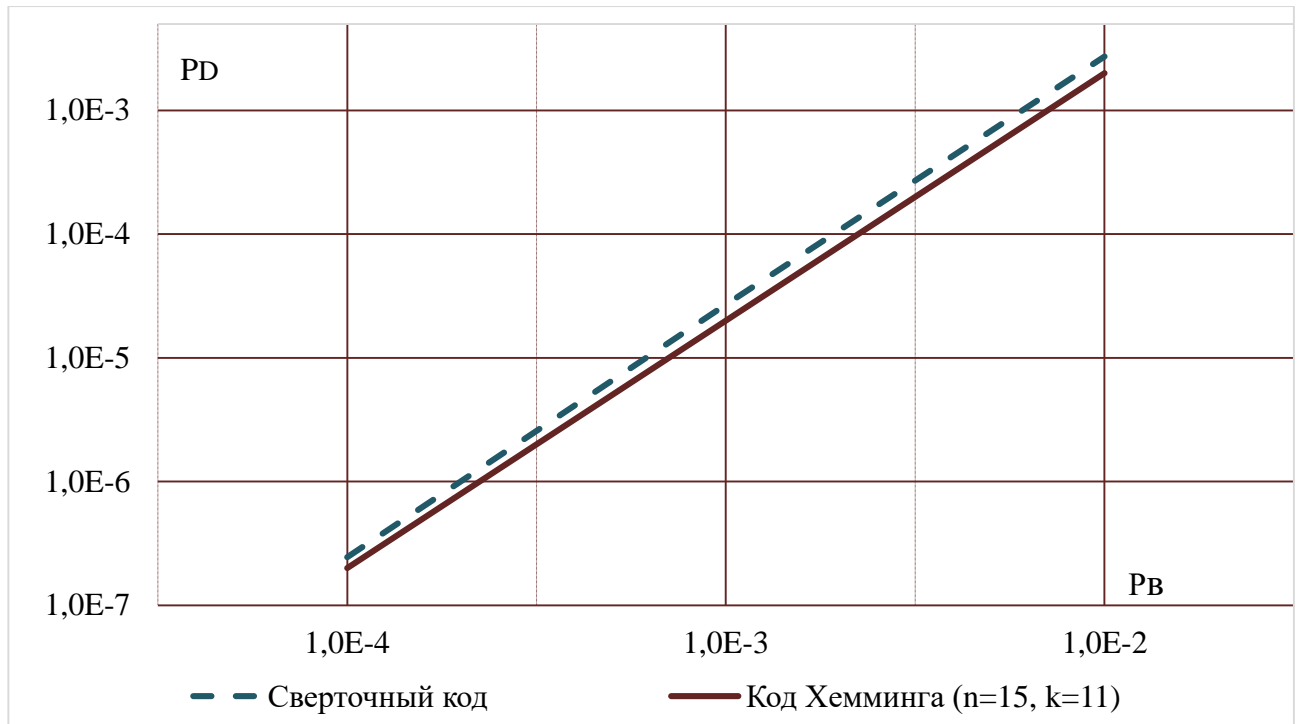


Рисунок 2.28. Сравнение эффективности перфорированного сверточного кода и кода Хемминга

Произведем сравнение эффективности декодирования сверточного кода с выкалываниями согласно алгоритму 5 и кода Хемминга с параметрами:  $k = 26$ ,  $n = 31$ . Данные для сравнения приведены в таблице 2.21 и на рисунке 2.29. Перфорированный сверточный код показал значительно меньшую эффективность декодирования при всех значениях  $p_B$ . С ростом  $p_B$  наблюдается уменьшение величины различия эффективности.

Таблица 2.21

Вероятность появления ошибочного бита в декодированном сообщении

	$p_B = 1 \cdot 10^{-2}$	$p_B = 1 \cdot 10^{-3}$	$p_B = 1 \cdot 10^{-4}$
$p_D$ , сверточный код	$1,2 \cdot 10^{-2}$	$8,8 \cdot 10^{-4}$	$8,1 \cdot 10^{-5}$
$p_D$ , код Хемминга ( $k = 26, n = 31$ )	$4,2 \cdot 10^{-3}$	$4,2 \cdot 10^{-5}$	$4,2 \cdot 10^{-7}$
$K_{\Sigma}$ , дБ	5,4	13,2	22,8

Подведем общий итог сравнения эффективности декодирования перфорированных сверточных кодов и блочных кодов Хемминга. При скорости кодирования равной  $4/7$  эффективность декодирования сверточного кода

значительно выше, однако она резко снижается при росте скорости кода. Учитывая простоту реализации кодера-декодера кода Хемминга, можно сделать вывод, что применение перфорированных сверточных кодов со скоростями выше  $3/4$  менее целесообразно.

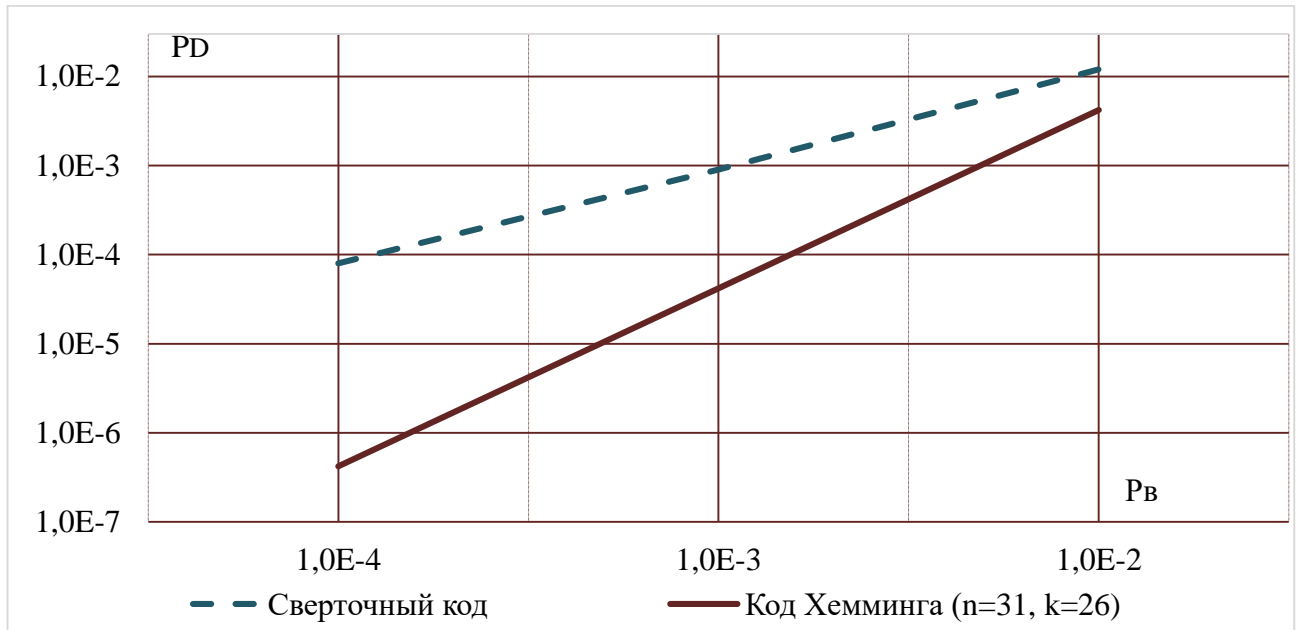


Рисунок 2.29. Сравнение эффективности перфорированного сверточного кода и кода Хемминга

В параграфе получены числовые результаты вероятности битовой ошибки в декодированной информационной последовательности перфорированного сверточного кода. При перфорации рассмотренного в параграфе сверточного кода возможно изменение кодовой скорости с 0,5 до 0,8 (то есть на 60 %), при декодировании жесткого решения детектора, и с 0,5 до 0,83 (то есть на 66 %), при декодировании мягкого решения детектора. Подробно рассмотрев особенности декодирования сверточного кода в соответствии с алгоритмом Витерби, функционирующим по принципу максимального правдоподобия, применим полученные знания с целью повышения эффективности последовательных каскадных кодов, созданных на основе внешнего сверточного кода.



## **2.6. Повышение адаптивных возможностей последовательных каскадных кодов, созданных на основе внешнего сверточного кода**

Как было сказано в п. 1.6, для улучшения корректирующей способности помехоустойчивые коды соединяют в разнообразные комбинации. Там же было приведено краткое описание последовательного каскадного кода. Структурная схема канала связи с ПКК изображена на рисунке 1.4. Известно, что даже при соединении двух относительно простых кодов в ПКК можно получить код с весьма высокой корректирующей способностью [107, 108, 109]. Отрицательной стороной такого соединения является значительный рост избыточности. Поясним сказанное на примере. Пусть у каждого из кодов, соединенных в каскад, число кодовых бит в два раза превышает число информационных, то есть кодовая скорость внутреннего и внешнего кодов одинакова и равна 0,5. Тогда кодовая скорость последовательного каскадного соединения таких кодов равна 0,25.

Важность наличия у кода возможностей по адаптации параметров была описана в п. 1.6. Качество канала связи может изменяться под воздействием разнообразных факторов. Большой нестабильностью обладает качество КВ и УКВ радиосвязи, особенно подвижной [110, 111, 112, 113]. Предлагается модифицированный метод последовательного кодирования, позволяющий регулировать избыточность и корректирующую способность ПКК, изменяя долю кодовых бит внешнего кода, кодируемых внутренним кодом [114]. Для удобства назовем такой ПКК гибким ПКК (далее – ГПКК). Применение ГПКК позволяет плавно изменять скорость передачи информации в зависимости от состояния канала передачи данных, повышая таким образом эффективность кодирования. Структурная схема канала связи с применением ГПКК изображена на рисунке 2.30.

Идея реализации ГПКК в определенном смысле схожа с идеей перфорации кода. В обоих случаях возможно гибкое изменение избыточности и корректирующей способности кода.

Разберем построение исследуемого ГПКК. Примем допущение, что ошибки, имеющиеся в символах на выходе декодера внутреннего кода, после прохождения

обратного перемежителя (см. рис. 2.30) стали независимыми. Тогда, зная значения вероятности битовой ошибки  $p_V$  в декодированном информационном сообщении для того или иного внутреннего кода, можно смоделировать его работу, сгенерировав появление ошибок с вероятностью  $p_V$  на входе декодера внешнего кода. Для остальных символов, поступающих на вход декодера внешнего кода, минуя декодер внутреннего кода, смоделируем прохождение канала передачи данных, генерируя ошибки в соответствии с вероятностью появления ошибочного бита в канале  $p_B$ . В качестве внешнего кода будем рассматривать сверточный код, декодирование которого осуществляется в соответствии с рассмотренным ранее алгоритмом Витерби. По итогам декодирования будет осуществляться оценка корректирующей способности кода путем нахождения вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$ . Исследования будут проводиться при разном соотношении количества бит, кодированных и не кодированных внутренним кодом.

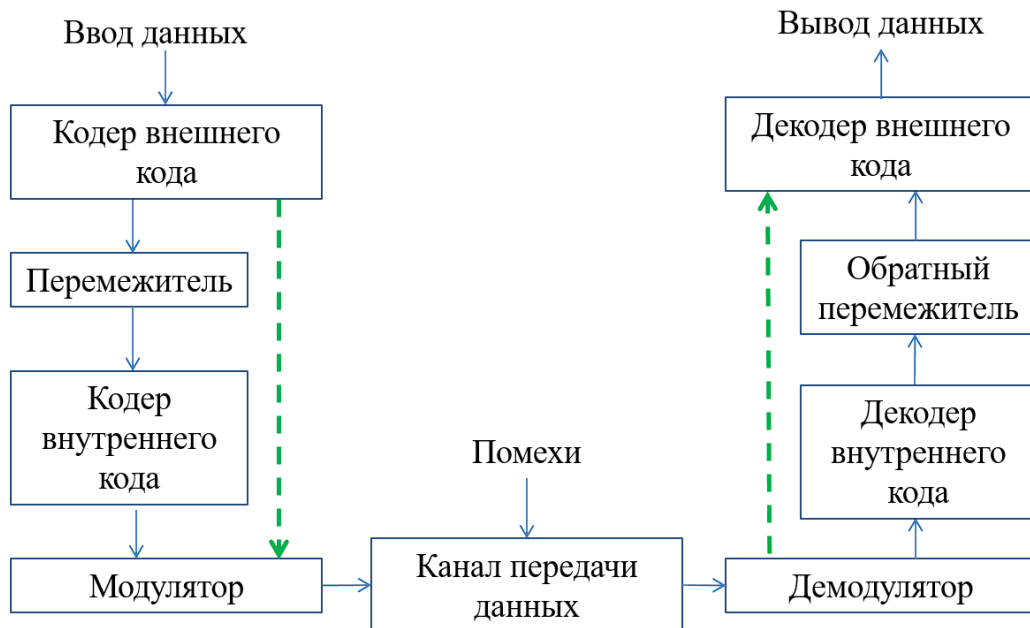


Рисунок 2.30. Структурная схема канала связи с применением ГПКК

В качестве внешнего кода используем несистематический сверточный код, рассмотренный в п. 2.1.

Проведем исследования дающие количественную информацию о значении  $p_D$  ГПКК в зависимости от следующих параметров:

- вероятности ошибки в канале передачи данных  $p_B$ ;
- вероятности ошибки на выходе внутреннего декодера  $p_V$ ;
- количества символов с выхода внутреннего кода, присутствующих во входной последовательности внешнего кода.

С целью реализации заявленных исследований для рассмотренного внешнего кода на языке программирования Python создан программный имитатор, моделирующий работу ГПКК, с учетом принятых допущений [115]. Программа содержит в своем составе следующие основные компоненты:

- генератор информационной последовательности;
- кодер внешнего кода;
- блок моделирования выхода детектора приемника с заданной вероятностью появления ошибочного бита  $p_B$ ;
- блок моделирования выхода декодера внутреннего кода с вероятностью появления ошибочного бита  $p_V$ ;
- декодер внешнего кода.

В программном имитаторе реализация генерации информационной последовательности, генерации вектора ошибок с заданной вероятностью и мягкого выхода детектора подробно описана в п. 2.2.

В качестве внутреннего кода будем использовать поочередно один из трех хорошо известных кодов, рассмотренных в главе 3: Хемминга, Голея и Рида-Соломона. Дадим короткую характеристику выбранным кодам. Кодовое слово кода Хемминга состоит из четырех информационных и трех проверочных бит. Такой код Хемминга способен исправить одну ошибку в кодовом слове. Кодовое слово кода Голея состоит из двенадцати информационных и одиннадцати проверочных бит. Код Голея способен исправить три ошибки в кодовом слове при любом их расположении. Код Рида-Соломона относится к недвоичным кодам. Выбран код, структура кодового слова которого состоит из пяти информационных и четырех проверочных байт. Такой код способен исправить два любых ошибочных байта. Код Рида-Соломона эффективен в каналах с высокой степенью группирования ошибок и обладает низкой корректирующей способностью при наличии большого

числа независимых ошибок. Рассмотрим корректирующую способность трех описанных кодов. В таблице 2.22 приведены значения вероятности битовой ошибки при наличии в канале независимых ошибок для выбранных внутренних кодов и внешнего сверточного кода. Значения, приведенные в таблице взяты из результатов исследований, приведенных в п. 2.1 и п. 3.2 и источника [116].

Таблица 2.22

Вероятность появления ошибочного бита в декодированном информационном сообщении

Вид кода	$p_B = 1 \cdot 10^{-1}$	$p_B = 1 \cdot 10^{-2}$	$p_B = 1 \cdot 10^{-3}$
Хэмминга	$6,7 \cdot 10^{-2}$	$8,6 \cdot 10^{-4}$	$9,0 \cdot 10^{-6}$
Голея	$3,1 \cdot 10^{-2}$	$1,9 \cdot 10^{-5}$	$2,3 \cdot 10^{-9}$
Рида-Соломона	$2,7 \cdot 10^{-1}$	$4,9 \cdot 10^{-3}$	$1,0 \cdot 10^{-5}$
Сверточный код	$6,27 \cdot 10^{-2}$	$4,75 \cdot 10^{-5}$	$5,2 \cdot 10^{-8}$

В ходе исследований выборка бит для кодирования внутренним кодом осуществлялась для каждого: десятого, восьмого, шестого, четвертого и второго бита из последовательности, закодированной с использованием внешнего кода. Обозначим период кодирования внутренним кодом как  $T_N$ . Тогда, например, при  $T_N$  равном 2, кодируется внутренним кодом каждый второй бит (через одного), а при  $T_N$  равном 4, кодируется внутренним кодом каждый четвертый бит (через три) и т.д.

Результаты исследования ГПКК для случая использования в качестве внутреннего кода Рида-Соломона, кода Хемминга и кода Голея приведены в таблицах 2.23, 2.24 и 2.25, соответственно. Проанализировав результаты, можно сделать вывод о возможности изменения в широких пределах корректирующей способности, путем осуществления кодирования внутренним кодом только определенной части бит. Диапазон таких изменений увеличивается при снижении вероятности ошибочного бита в декодируемой внешней кодом последовательности. С учетом избыточности, вносимой при кодировании внутренним кодом, плавное изменение скорости передачи информации в

зависимости от состояния канала возможно до 48 % при применении кода Голея в качестве внутреннего кода, что соответствует энергетическому выигрышу до 1,7 дБ.

Таблица 2.23

Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Рида-Соломона

$T_N$	$p_B=1 \cdot 10^{-2}, p_V=4,9 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-3}, p_V=1,0 \cdot 10^{-5}$
10	$4,40 \cdot 10^{-5}$	$4,5 \cdot 10^{-8}$
8	$3,44 \cdot 10^{-5}$	$3,8 \cdot 10^{-8}$
6	$2,72 \cdot 10^{-5}$	$2,9 \cdot 10^{-8}$
4	$2,35 \cdot 10^{-5}$	$2,6 \cdot 10^{-8}$
2	$1,55 \cdot 10^{-5}$	$8,50 \cdot 10^{-9}$

Таблица 2.24

Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Хемминга

$T_N$	$p_B=1 \cdot 10^{-1}, p_V=6,7 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-2}, p_V=8,6 \cdot 10^{-4}$	$p_B=1 \cdot 10^{-3}, p_V=9,0 \cdot 10^{-6}$
10	$5,59 \cdot 10^{-2}$	$4,30 \cdot 10^{-5}$	$4,4 \cdot 10^{-8}$
8	$5,39 \cdot 10^{-2}$	$3,04 \cdot 10^{-5}$	$3,6 \cdot 10^{-8}$
6	$5,37 \cdot 10^{-2}$	$2,36 \cdot 10^{-5}$	$2,6 \cdot 10^{-8}$
4	$4,92 \cdot 10^{-2}$	$2,05 \cdot 10^{-5}$	$2,2 \cdot 10^{-8}$
2	$3,78 \cdot 10^{-2}$	$7,25 \cdot 10^{-6}$	$2,28 \cdot 10^{-9}$

Для улучшения восприятия, данные из таблицы 2.25 представлены в графическом виде на рисунке 2.31.

Можно предложить следующую стратегию по адаптации параметров сверточного кода. В случае улучшения качества канала передачи данных увеличить скорость кода можно, применяя тот или иной алгоритм перфорации, описанный в п. 2.5. В случае ухудшения качества канала передачи данных, вводить в действие каскадное кодирование, и, в зависимости от степени ухудшения качества канала, изменяется число бит, кодированных внутренним кодом. В каналах, где

периодически фиксируются случаи высокой степени группирования ошибок, можно рекомендовать применение кода Рида-Соломона в качестве внутреннего кода. При использовании описанной стратегии обеспечивается плавная регулировка скорости кода и корректирующей способности, что является важным преимуществом адаптивных возможностей, изложенным в п. 1.6.

Таблица 2.25

Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Голя

$T_N$	$p_B=1 \cdot 10^{-1}, p_V=3,1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-2}, p_V=1,9 \cdot 10^{-5}$	$p_B=1 \cdot 10^{-3}, p_V=2,3 \cdot 10^{-9}$
10	$4,98 \cdot 10^{-2}$	$3,23 \cdot 10^{-5}$	$4,0 \cdot 10^{-8}$
8	$4,81 \cdot 10^{-2}$	$3,34 \cdot 10^{-5}$	$3,4 \cdot 10^{-8}$
6	$4,39 \cdot 10^{-2}$	$2,43 \cdot 10^{-5}$	$2,4 \cdot 10^{-8}$
4	$3,61 \cdot 10^{-2}$	$1,61 \cdot 10^{-5}$	$1,9 \cdot 10^{-8}$
2	$1,67 \cdot 10^{-2}$	$4,38 \cdot 10^{-6}$	$1,1 \cdot 10^{-10}$

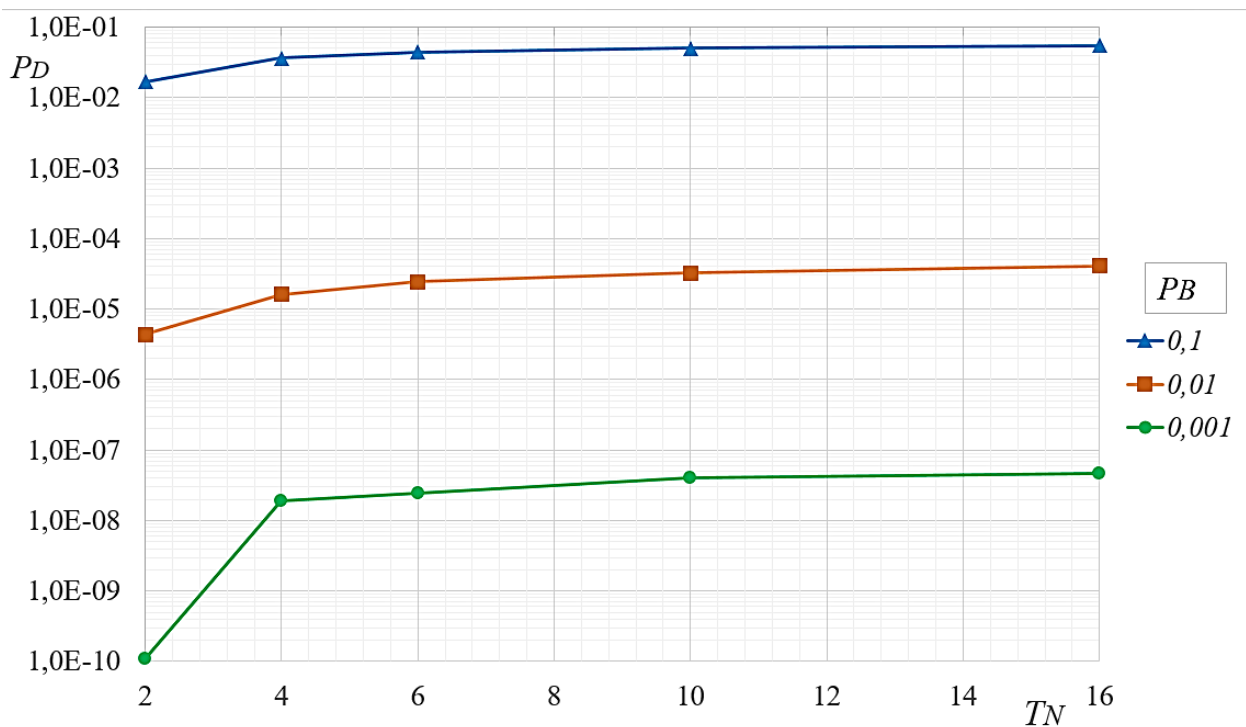


Рисунок 2.31. Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Голя

В параграфе рассмотрен модифицированный метод последовательного кодирования, обеспечивший возможность повышения адаптивности последовательного каскадного кода, созданного на основе внешнего сверточного кода. Далее рассмотрим возможности повышения корректирующей способности, используя принципы построения ГПКК.

### **2.7. Повышение корректирующей способности ГПКК при декодировании по максимуму правдоподобия путем применения к значениям декодируемых бит коэффициентов**

В случае, когда на вход декодера внешнего кода в некотором количестве поступают символы с выхода декодера внутреннего кода, то значения таких декодированных символов являются более достоверными, чем значения остальных символов, которые не подвергались кодированию-декодированию внутренним кодом. Можно предположить, что чем выше коэффициент, на который умножается значение декодируемого символа, тем выше влияние этого символа на результат декодирования. При сверточном кодировании, как это было описано в п. 2.1, ближайшие кодовые биты связаны между собой, поэтому повышение достоверности отдельных бит может быть распространено и на соседние биты. Учесть большую степень достоверности символов, дополнительно закодированных внутренним кодом, можно путем умножения их значений на повышающий коэффициент  $K > 1$ , или путем умножения значений менее достоверных символов на понижающий коэффициент  $K < 1$  [114]. Предполагается, что при этом корректирующая способность ГПКК должна повыситься. Рассмотрим справедливость данного предположения на примере декодирования кодовой последовательности по решетке сверточного кода в соответствии с алгоритмом Витерби.

При поступлении на вход кодера одного бита информации на выходе формируются два кодовых бита. Соответственно, при подаче на вход кодера

информационной последовательности, например, 110101, состоящей из шести бит, формируется кодовая последовательность 11 01 01 00 10 00, состоящая из двенадцати бит. При передаче биполярными сигналами такая кодовая последовательность имеет вид: 11 -11 -11 -1-1 1-1 -1-1. В процессе каскадного кодирования вся кодовая последовательность с выхода кодера сверточного кода или ее часть подается на вход кодера внутреннего кода. На вход декодера внешнего кода поступают значения символов как из канала передачи данных (с выхода детектора приемника), так и с выхода декодера внутреннего кода. Было выдвинуто предположение о влиянии значений описанных ранее коэффициентов  $K$  на результат декодирования. Подтвердим данное предположение, рассмотрев практические примеры декодирования декодером внешнего сверточного кода. При принятии детектором жесткого решения о значении символа и наличии в принятой кодовой последовательности двух ошибок, входная последовательность декодера сверточного кода имеет вид: 11 -11 11 -1-1 11 -1-1. Значения ошибочных бит подчеркнуты. Произведем декодирование согласно алгоритму Витерби принятой кодовой последовательности, схематично изобразив процесс декодирования по решетке на рисунке 2.32. Выход декодера формируется на основе пути, имеющего наименьшую метрику. В рассмотренном примере сразу два пути имеют минимальную метрику, следовательно, кодовая последовательность может быть декодирована ошибочно.

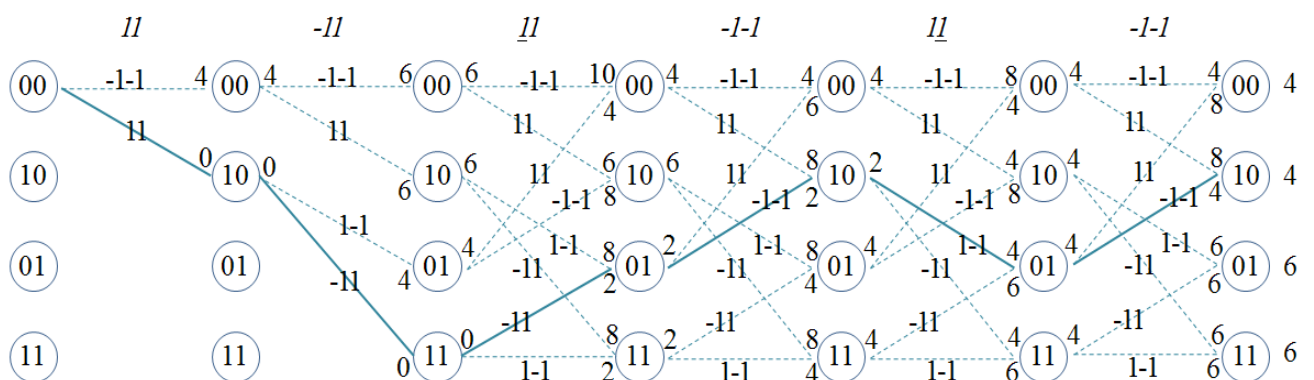


Рисунок 2.32. Структура декодирования кодового слова, содержащего две ошибки



Покажем, как применение коэффициентов к рассмотренной последовательности, содержащей две ошибки, приведет к повышению корректирующей способности. Пусть каждый шестой бит входной последовательности декодера сверточного кода кодирован внутренним кодом, а значения символов, не подвергшихся кодированию внутренним кодом, а значит, менее достоверных, умножаются на коэффициент  $K = 0,5$ . Тогда входная последовательность декодера сверточного кода будет иметь следующий вид:  $0,5 \ 0,5 \ -0,5 \ 0,5 \ \underline{0,5} \ 1 \ -0,5 \ -0,5 \ 0,5 \ \underline{0,5} \ -0,5 \ -1$ . Процесс декодирования схематично показан на рисунке 2.33. Путь, соответствующий переданной кодовой последовательности, имеет минимальную метрику. Ошибки исправлены.

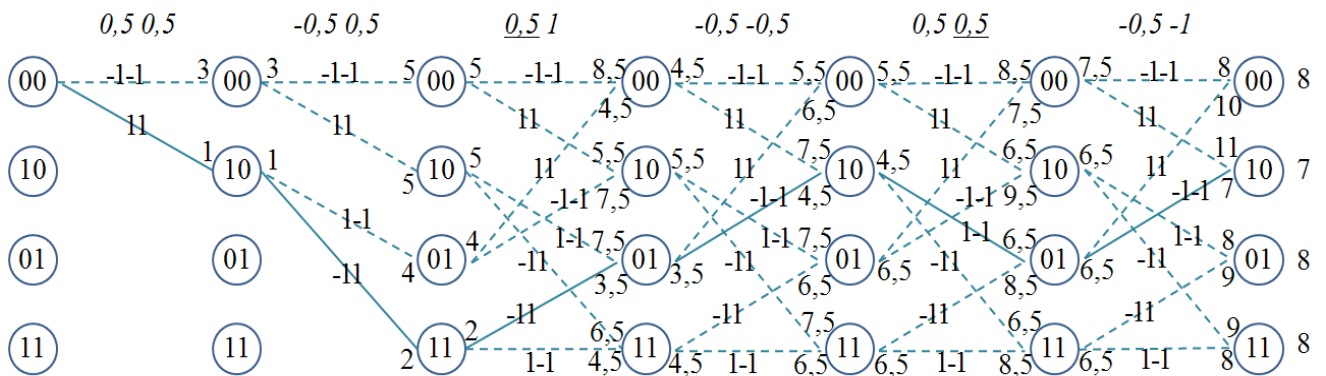


Рисунок 2.33. Структура декодирования кодового слова при  $K = 0,5$

Ошибки в символах, дополнительно кодированных внутренним кодом маловероятны, но возможны, и их появление при использовании коэффициентов чаще приводит к ошибкам декодирования. Подтвердим сказанное на примере, демонстрирующем ошибочное декодирование при наличии в кодовой последовательности двух ошибок. Пусть в рассмотренной входной последовательности первая ошибка располагается в той же третьей паре бит, но в бите с  $K = 1$ , а вторая ошибка, как в предыдущем примере, в десятом бите. Тогда декодируемая последовательность имеет вид:  $0,5 \ 0,5 \ -0,5 \ 0,5 \ -0,5 \ \underline{-1} \ -0,5 \ -0,5 \ 0,5 \ \underline{0,5} \ -0,5 \ -1$ . Процесс декодирования схематично показан на рисунке 2.34. Путь с минимальной метрикой отличается от переданной кодовой последовательности. Декодирование ошибочно.

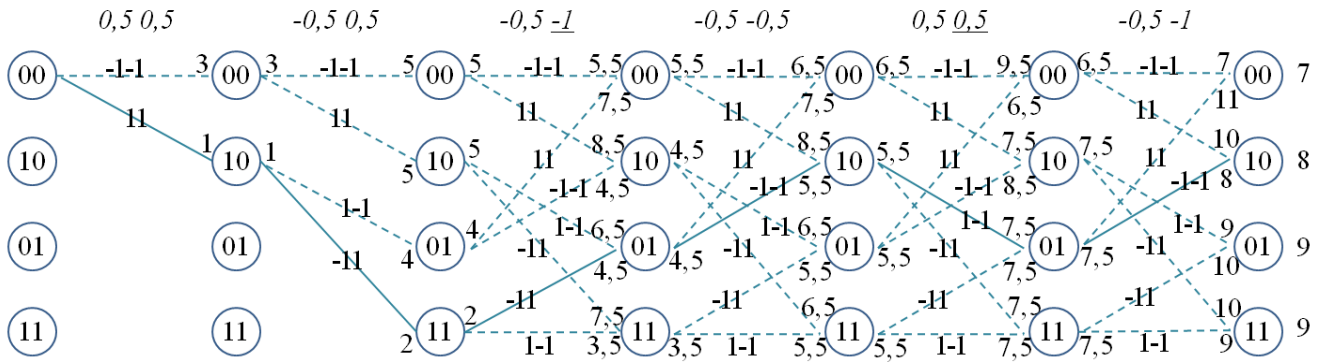


Рисунок 2.34. Структура декодирования кодового слова при ошибке в бите с  $K = 1$

Приведем пример успешного декодирования последовательности, содержащей ошибку в каждой паре из трех подряд следующих пар бит, на входе декодера сверточного кода. Пусть каждый третий бит входной последовательности кодирован внутренним кодом, а коэффициент  $K = 0,3$ . Тогда входная последовательность декодера будет иметь следующий вид:  $0,3 \ 0,3 \ -1 \ -0,3 \ 0,3 \ 1 \ -0,3 \ 0,3 \ 1 \ -0,3 \ -0,3 \ -1 \ 0,3 \ -0,3 \ -1 \ -0,3$ . Процесс декодирования схематично показан на рисунке 2.35. Путь, соответствующий переданной кодовой последовательности, имеет минимальную метрику. Три ошибки исправлены. Для данного случая следует отметить эффективность кода в противостоянии группирующимся ошибкам.

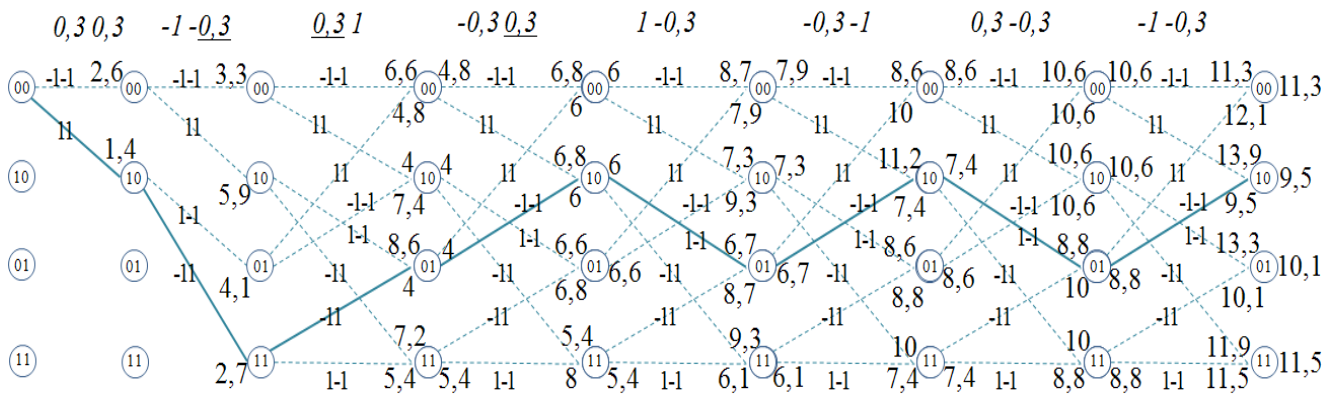


Рисунок 2.35. Структура декодирования кодового слова при наличии трех ошибок

Рассмотрев представленные примеры декодирования, можно сделать вывод об определенных качественных изменениях в результатах декодирования ГПКК при изменении числа бит, кодированных внутренним кодом, и использовании коэффициентов.

Произведем исследования корректирующей способностей ГПКК, аналогичные проведенным исследованиям в п. 2.6, при умножении закодированных внутренним кодом бит на описанные ранее коэффициенты. Исследования будут проводиться при разном соотношении количества бит, кодированных и не кодированных внутренним кодом, и разном значении коэффициента  $K$ . В отличие от рассмотренных примеров декодирования, в программном имитаторе вместо умножения на понижающие коэффициенты значений символов, не подвергшихся кодированию внутренним кодом, было реализовано умножение на коэффициент  $K > 1$  значений символов, закодированных внутренним кодом [115].

Результаты исследования ГПКК для случая использования по очереди кода Рида-Соломона, кода Хемминга и кода Голя в качестве внутреннего кода приведены в таблицах 2.26, 2.27 и 2.28, соответственно. В таблицах в скобках приведены конкретные значения  $K$ , для которых получены наилучшие результаты корректирующей способности.

Таблица 2.26

Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Рида-Соломона

$T_N$	$p_B=1 \cdot 10^{-2}, p_V=4,9 \cdot 10^{-3}$		$p_B=10^{-3}, p_V=1 \cdot 10^{-5}$	
	$K=1$	$K>1$	$K=1$	$K>1$
10	$4,40 \cdot 10^{-5}$	$3,02 \cdot 10^{-5}$ ( $K=1,2$ )	$4,5 \cdot 10^{-8}$	$2,51 \cdot 10^{-8}$ ( $K=1,3$ )
8	$3,44 \cdot 10^{-5}$	$2,65 \cdot 10^{-5}$ ( $K=1,2$ )	$3,8 \cdot 10^{-8}$	$1,79 \cdot 10^{-8}$ ( $K=1,5$ )
6	$2,72 \cdot 10^{-5}$	$2,58 \cdot 10^{-5}$ ( $K=1,3$ )	$2,9 \cdot 10^{-8}$	$1,71 \cdot 10^{-8}$ ( $K=1,5$ )
4	$2,35 \cdot 10^{-5}$	$1,83 \cdot 10^{-5}$ ( $K=1,3$ )	$2,6 \cdot 10^{-8}$	$3,32 \cdot 10^{-9}$ ( $K=1,5$ )
2	$1,55 \cdot 10^{-5}$	$7,00 \cdot 10^{-6}$ ( $K=1,3$ )	$8,50 \cdot 10^{-9}$	$1,69 \cdot 10^{-10}$ ( $K=1,5$ )

Проанализировав результаты, можно сделать вывод, что применение повышающего коэффициента к более достоверным значениям символов, закодированных внутренним кодом, приводит к росту корректирующей способности. При кодировании каждого второго бита внутренним кодом Голя,

наблюдается снижение вероятности битовой ошибки в 1000 раз (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ). Эффект от применения коэффициентов увеличивается при снижении вероятности ошибочного бита в декодируемой последовательности, а также при увеличении разницы между значениями  $p_B$  и  $p_V$ .

Таблица 2.27

Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Хемминга

$T_N$	$p_B=10^{-1}, p_V=6,7 \cdot 10^{-2}$		$p_B=10^{-2}, p_V=8,6 \cdot 10^{-4}$		$p_B=10^{-3}, p_V=9,0 \cdot 10^{-6}$	
	$K=1$	$K>1$	$K=1$	$K>1$	$K=1$	$K>1$
10	$5,59 \cdot 10^{-2}$	$5,42 \cdot 10^{-2}$ ( $K=1,3$ )	$4,30 \cdot 10^{-5}$	$2,01 \cdot 10^{-5}$ ( $K=1,5$ )	$4,40 \cdot 10^{-8}$	$1,33 \cdot 10^{-8}$ ( $K=1,5$ )
8	$5,39 \cdot 10^{-2}$	$5,22 \cdot 10^{-2}$ ( $K=1,3$ )	$3,04 \cdot 10^{-5}$	$2,00 \cdot 10^{-5}$ ( $K=1,8$ )	$3,6 \cdot 10^{-8}$	$7,18 \cdot 10^{-9}$ ( $K=1,8$ )
6	$5,37 \cdot 10^{-2}$	$5,36 \cdot 10^{-2}$ ( $K=1,3$ )	$2,36 \cdot 10^{-5}$	$1,98 \cdot 10^{-5}$ ( $K=1,8$ )	$2,6 \cdot 10^{-8}$	$6,86 \cdot 10^{-9}$ ( $K=1,8$ )
4	$4,92 \cdot 10^{-2}$	$4,65 \cdot 10^{-2}$ ( $K=1,3$ )	$2,05 \cdot 10^{-5}$	$6,13 \cdot 10^{-6}$ ( $K=1,8$ )	$2,2 \cdot 10^{-8}$	$4,25 \cdot 10^{-9}$ ( $K=1,8$ )
2	$3,78 \cdot 10^{-2}$	$3,45 \cdot 10^{-2}$ ( $K=1,3$ )	$7,25 \cdot 10^{-6}$	$5,00 \cdot 10^{-8}$ ( $K=1,8$ )	$2,28 \cdot 10^{-9}$	$1,44 \cdot 10^{-10}$ ( $K=1,8$ )

Таблица 2.28

Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Голя

$T_N$	$p_B=10^{-1}, p_V=3,1 \cdot 10^{-2}$		$p_B=10^{-2}, p_V=1,9 \cdot 10^{-5}$		$p_B=10^{-3}, p_V=2,3 \cdot 10^{-9}$	
	$K=1$	$K>1$	$K=1$	$K>1$	$K=1$	$K>1$
10	$4,98 \cdot 10^{-2}$	$4,65 \cdot 10^{-2}$ ( $K=1,3$ )	$3,23 \cdot 10^{-5}$	$1,99 \cdot 10^{-5}$ ( $K=2$ )	$4,0 \cdot 10^{-8}$	$1,33 \cdot 10^{-8}$ ( $K=1,8$ )
8	$4,81 \cdot 10^{-2}$	$4,33 \cdot 10^{-2}$ ( $K=1,8$ )	$3,34 \cdot 10^{-5}$	$1,80 \cdot 10^{-5}$ ( $K=1,8$ )	$3,4 \cdot 10^{-8}$	$1,6 \cdot 10^{-8}$ ( $K=2$ )
6	$4,39 \cdot 10^{-2}$	$4,05 \cdot 10^{-2}$ ( $K=1,8$ )	$2,43 \cdot 10^{-5}$	$1,78 \cdot 10^{-5}$ ( $K=2$ )	$2,4 \cdot 10^{-8}$	$1,55 \cdot 10^{-8}$ ( $K=3$ )
4	$3,61 \cdot 10^{-2}$	$3,02 \cdot 10^{-2}$ ( $K=1,5$ )	$1,61 \cdot 10^{-5}$	$1,75 \cdot 10^{-6}$ ( $K=3$ )	$1,9 \cdot 10^{-8}$	$1,46 \cdot 10^{-10}$ ( $K=4$ )
2	$1,67 \cdot 10^{-2}$	$1,18 \cdot 10^{-2}$ ( $K=1,8$ )	$4,38 \cdot 10^{-6}$	$2,67 \cdot 10^{-8}$ ( $K=4$ )	$1,1 \cdot 10^{-10}$	$1,2 \cdot 10^{-13}$ ( $K=5$ )

Для улучшения восприятия, данные из таблицы 2.28 представлены в графическом виде на рисунке 2.36. Пунктиром приведены графики для  $K=1$ .

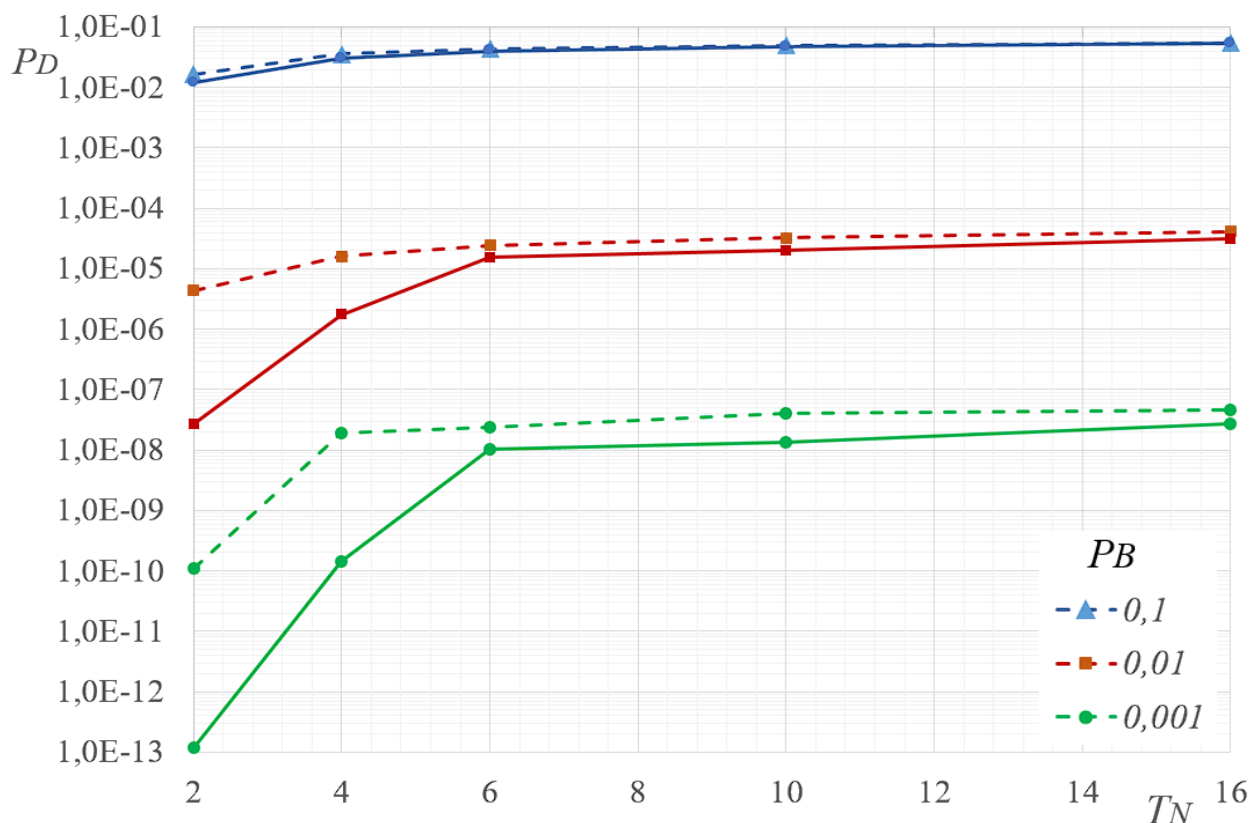


Рисунок 2.36. Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Голя

В параграфе рассмотрены изменения метода декодирования по максимуму правдоподобия при применении к значениям символов коэффициентов, пропорциональных их достоверности. В результате чего наблюдался рост корректирующей способности ГПКК. При кодировании каждого второго бита внутренним кодом Голя, вероятность битовой ошибки снизилась в 920 раз – с  $1,1 \cdot 10^{-10}$  до  $1,2 \cdot 10^{-13}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ).

## Выводы по главе 2

В начале главы введены обозначения, применяемые при изучении процесса декодирования сверточного кода в соответствии с алгоритмом Витерби,

функционирующего по принципу максимального правдоподобия. Произведен расчет корректирующей способности сверточного кода. Описаны программные генераторы массивов информации и формователи мягкого или жесткого решения детектора, созданные на языках программирования Python и C++. Данные фрагменты программ используются в большинстве программных продуктов, разработанных в ходе диссертационного исследования.

Качественно показано повышение корректирующей способности кода при декодировании мягкого решения детектора. Произведено сравнительное исследование корректирующей способности сверточного кода при декодировании мягкого и жесткого решения детектора. Зафиксирован рост корректирующей способности при декодировании мягкого решения детектора. При  $p_B = 1 \cdot 10^{-2}$  и длине информационного сообщения равной четырем битам получено снижение вероятности битовой ошибки в декодированном информационном сообщении в 14 раз. При снижении вероятности битовой ошибки в канале передачи данных, выигрыш от применения мягкого решения детектора снижается.

Подробно рассмотрен механизм декодирования при применении выкалывания бит из кодовой последовательности. Произведен расчет вероятности появления ошибочного бита в декодированном информационном сообщении при применении выкалывания бит. По итогам проведенного анализа эффективности декодирования перфорированных сверточных кодов, сделан вывод о нецелесообразности применения алгоритмов перфорации, приводящих к росту скорости кодирования выше  $3/4$ .

Предложен модифицированный метод последовательного каскадного кодирования, позволяющий изменять параметры кода, путем осуществления кодирования внутренним кодом только определенной части бит с выхода кодера внешнего кода. В качестве внешнего кода использован несистематический сверточный код. С учетом избыточности, вносимой при кодировании внутренним кодом, плавное изменение скорости передачи информации, в зависимости от состояния канала, возможно до 48 % при применении кода Голея в качестве внутреннего кода, что соответствует энергетическому выигрышу до 1,7 дБ.

Проведенное моделирование продемонстрировало существенное увеличение корректирующей способности при умножении значений декодируемых внешним кодом символов на коэффициенты, различные для символов, подвергшихся и не подвергшихся кодированию внутренним кодом. Определены величины коэффициентов, обеспечивающие минимальную величину  $p_D$ . Выявлено, что эффект роста корректирующей способности увеличивается при снижении вероятности битовой ошибки в канале передачи данных. При кодировании каждого второго бита внутренним кодом Голя, наблюдается снижение вероятности битовой ошибки в 920 раз – с  $1,1 \cdot 10^{-10}$  до  $1,2 \cdot 10^{-13}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ ).

В следующей главе будет рассмотрен обширный класс корректирующих кодов – блочные коды. Будут проведены исследования возможности по повышению эффективности последовательных каскадных кодов, построенных на основе внешнего блочного кода.

## Глава 3. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПОСЛЕДОВАТЕЛЬНЫХ КАСКАДНЫХ КОДОВ, ПОСТРОЕННЫХ НА ОСНОВЕ ВНЕШНЕГО БЛОЧНОГО КОДА

Данная глава посвящена вопросам повышения эффективности последовательных каскадных кодов, построенных на основе внешнего блочного кода. В начале главы дается обзор ряда известных двоичных блочных кодов, а также недвоичного кода Рида-Соломона. Производится оценка корректирующей способности рассмотренных блочных кодов. Изложенные в главе 1 методы повышения эффективности, такие как мягкое декодирование, перфорация рассматриваются применительно к блочным кодам. Дается числовая оценка эффективности рассмотренных методов. Производится оценка эффективности каскадного кода, состоящего из двух блочных кодов. Описывается построение блочного кода и его декодирование согласно алгоритму Витерби с использованием синдромной решетки. Рассматривается реализация последовательного каскадного кодирования на основе внешнего блочного кода. Предлагается модификация метода построения каскадного кода с реализацией плавной адаптации параметров кода. С целью улучшения корректирующей способности предлагается модификация метода декодирования по максимуму правдоподобия с применением коэффициентов, пропорциональных достоверности значений декодируемых бит.

### **3.1. Обзор часто применяемых на практике блочных кодов**

Блочные коды получили широкое распространение в разнообразных телекоммуникационных системах, системах хранения и обработки данных [117, 118, 119, 120, 121, 122, 123]. В последние годы по направлению повышения эффективности блочных кодов опубликованы научные работы разных авторов



[124, 125, 126]. Произведем краткий обзор некоторых наиболее распространенных блочных кодов, упоминаемых в разных разделах диссертационного исследования.

### 3.1.1. Коды Хемминга

Коды Хемминга образуют одно из самых известных семейств линейных блочных кодов [12, 127, 128, 129, 130]. Для каждого натурального числа  $m \geq 3$  существует двоичный код Хэмминга со следующими параметрами:

- длина кодовых слов  $n = 2^m - 1$ ;
- число информационных разрядов  $k = 2^m - 1 - m$ ;
- число проверочных разрядов  $m = n - k$ ;
- корректирующая способность  $t = 1, d_{min} = 3$ .

Код Хэмминга требует минимальной избыточности при заданной длине блока для исправления одной ошибки. Код Хемминга является совершенным кодом.

Исходя из особенностей кода Хемминга, изложенных выше, в таблице 3.1 представлены варианты основных параметров данного кода.

Таблица 3.1

Основные параметры кодов Хемминга

Число проверочных разрядов $m$	3	4	5	6
Число информационных разрядов $k$	$2 \div 4$	$5 \div 11$	$12 \div 26$	$27 \div 57$
Длина кодового слова $n$	$5 \div 7$	$9 \div 15$	$17 \div 31$	$33 \div 63$

Очевидно, что с точки зрения энергетической эффективности для каждого значения  $m$  целесообразно выбирать максимальное число информационных разрядов. Например, при  $m = 3$ :  $k = 4$   $n = 7$ . Такие варианты кода Хемминга представлены в таблице 3.2. Таким образом, при одинаковой корректирующей способности код будет обладать большей скоростью.

Преимуществом данного кода является его простота, и, как следствие, высокие скорости кодирования и декодирования. Недостатком является способность исправлять лишь одиночные ошибки.

Наиболее эффективные коды Хемминга

Число проверочных разрядов $t$	3	4	5	6
Число информационных разрядов $k$	4	11	26	57
Длина кодового слова $n$	7	15	31	63

### 3.1.2. Коды Голея

Код Голея [131, 132, 133, 134, 135] – это совершенный код с параметрами  $n = 23$ ,  $k = 12$  и минимальным расстоянием Хэмминга, равным семи. Этот код гарантирует исправление всех трехбитовых ошибок. Преимуществом данного кода является относительно простой алгоритм декодирования и способность противостоять трехбитовым ошибкам. Код Голея (23,12) имеет генерирующий полином  $g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + x^1$ . Кодирование кода Голея происходит путем реализации полиномиального деления. Декодирование кода Голея как правило осуществляется с помощью декодера Меггита.

### 3.1.3. Коды Боуза-Чоудхури-Хоквенгема

Коды Боуза-Чоудхури-Хоквенгема (далее – БЧХ) [136, 137, 138, 139] представляют собой развитие блочных кодов Хэмминга. Данный вид кодов предоставляет большую свободу выбора длины блока, степени кодирования, размеров алфавита и возможностей коррекции ошибок.

Теоретически коды БЧХ могут исправлять произвольное количество ошибок. В случае, когда кодовые слова состоят из нескольких сотен символов, коды БЧХ дают значительный выигрыш по сравнению с другими блочными кодами, имеющими ту же длину и степень кодирования. Наиболее часто в кодах БЧХ применяются кодовые слова длиной  $n = 2^h - 1$ , где  $h = 3, 4, 5 \dots$ . Для кодов БЧХ максимальная эффективность кодирования достигается при степенях кодирования между  $1/3$  и  $3/4$ . С математической точки зрения в основе построения кодов БЧХ

лежит операция вычисления остатка от деления вектора информационного слова на генерирующий полином.

Преимуществами двоичных кодов БЧХ являются их разнообразие и хорошие возможности по борьбе с одиночными ошибками. Недостатками являются довольно сложные алгоритмы декодирования (особенно для кодов большой длины) и невозможность противостоять пакетам ошибок.

Символы кода БЧХ берутся из конечного поля Галуа [140, 141, 142]. Полем называют множество элементов, если для любых элементов этого множества определены операции сложения и умножения, а также выполняется ряд аксиом (замкнутости, ассоциативности, коммутативности, дистрибутивности).

Одним из подклассов кодов БЧХ с недвоичными символами являются коды Рида-Соломона. Код Рида-Соломона [143, 144, 145] – это недвоичный случай кода БЧХ. Символы недвоичных кодов представляют собой многобитовые ( $m$ -битовые) последовательности. Коды Рида-Соломона имеют минимальное расстояние  $d_{min} = n - k + 1$  и способны исправлять  $t = \lfloor (n - k)/2 \rfloor$  ошибок.

В каналах связи множество передаваемых сигналов всегда конечно. Поля с конечным числом элементов  $q$  называют полями Галуа по имени их первого исследователя Эвариста Галуа и обозначают  $GF(q)$ . Число элементов поля  $q$  называют порядком поля. Конечные поля используются для построения ряда известных кодов и их декодирования. Двоичное поле  $GF(2)$  является простейшим полем Галуа, операции сложения и умножения в котором проводятся по правилам арифметики по модулю 2. Двоичное поле применяется при построении двоичных кодов БЧХ. Код Рида-Соломона, рассматриваемый в диссертационном исследовании имеет размер символа, равным одному байту, и строится с использованием поля Галуа  $GF(2^8)$ .

Преимуществом кодов Рида-Соломона является их возможность противостоять пакетам ошибок. Недостатками являются сложные алгоритмы декодирования.

В параграфе рассмотрен ряд наиболее распространенных блочных кодов. Рассмотренные коды упоминаются в разных разделах диссертационного

исследования. Далее произведем количественную оценку корректирующей способности упомянутых блочных кодов.

### 3.2. Сравнение корректирующей способности блочных кодов

Вероятность появления битовой ошибки в декодированном информационном сообщении является одной из основных величин, характеризующих корректирующую способность помехоустойчивых кодов. В диссертационном исследовании будут использоваться значения вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$  для ряда известных блочных кодов: Хэмминга, Голея, Рида-Соломона. В данном параграфе будет рассмотрен вывод формулы для расчета значения  $p_D$  для упомянутых кодов [146].

#### 3.2.1. Расчет корректирующей способности кода Хемминга

С учетом изложенного в п. 3.1 очевидно, что наибольшей корректирующей способностью обладает код Хемминга при  $m = 3$ . Оценим изменение корректирующей способности кода при росте  $m$ . Выведем формулу для нахождения вероятности ошибочного бита в декодированном сообщении.

Пусть  $PK_i$  – вероятность появления в принятом кодовом сообщении  $i$  ошибок, а  $PE_i$  – вероятность ошибочного декодирования сообщения при наличии в нем  $i$  ошибок. Тогда для кодового слова, состоящего из  $n$  кодовых бит, вероятность его ошибочного декодирования  $P_{BK}$  будет равна:

$$P_{BK} = PK_1 \cdot PE_1 + PK_2 \cdot PE_2 + PK_3 \cdot PE_3 + \dots + PK_n \cdot PE_n = \sum_{i=1}^n PK_i \cdot PE_i. \quad (3.1)$$

С учетом способности кода гарантированно исправлять  $t$  ошибочных бит формула будет иметь вид:

$$P_{BK} = \sum_{i=t+1}^n PK_i \cdot PE_i. \quad (3.2)$$

Пусть  $K_E$  – среднее количество ошибочных бит, возникающих при ошибочном декодировании одного кодового слова, тогда вероятность появления одного ошибочного бита при декодировании кодового слова равна:

$$P_{B1} = P_{BK} \cdot K_E. \quad (3.3)$$

С учетом того, что в одном кодовом слове присутствует  $k$  информационных бит, вероятность появления ошибочного бита в декодированном информационном сообщении будет равна:

$$p_D = P_{B1} : k. \quad (3.4)$$

Для кода с параметрами  $k=4$   $n=7$  выведем формулу для вычисления вероятности появления ошибочного бита в канале, а затем обобщим полученную формулу, сделав ее применимой к кодам Хемминга с другими параметрами. Структурная схема кодового слова такого кода Хемминга изображена на рисунке 3.1. Ошибка в декодировании возникает в случае наличия в кодовом слове двух и более ошибочных бит.

1 бит	2 бит	3 бит	4 бит	5 бит	6 бит	7 бит
Проверочные биты			Информационные биты			

Рисунок 3.1. Структурная схема кодового слова

Обозначим вероятность наличия двух ошибочных бит из  $n$  как  $P_{2изn}$ . По формуле Бернулли [147, 148, 149, 150] она равна:

$$P_{2изn} = \frac{n!}{2!(n-2)!} p_B^2 (1 - p_B)^{n-2}. \quad (3.5)$$

Вероятность наличия 2 ошибочных бит из 7 равна:

$$P_{2из7} = \frac{7!}{2!5!} p_B^2 (1 - p_B)^{7-2} = 21 p^2 (1 - p_B)^5. \quad (3.6)$$

Аналогично, вероятность наличия 3 ошибочных бит из 7 равна:

$$P_{3из7} = \frac{7!}{3!4!} p_B^3 (1 - p_B)^{7-3} = 35 p^3 (1 - p_B)^4. \quad (3.7)$$

При разных значениях вероятности появления ошибочного бита в канале с использованием формул (3.6) и (3.7) получаем следующие результаты:

при  $p_B = 10^{-2}$ :  $P_{2из7} = 2 \cdot 10^{-3}$  и  $P_{3из7} = 3,36 \cdot 10^{-5}$ ;

при  $p_B = 10^{-3}$ :  $P_{2из7} = 2,1 \cdot 10^{-5}$  и  $P_{3из7} = 3,49 \cdot 10^{-8}$ ;

при  $p_B = 10^{-4}$ :  $P_{2из7} = 2,1 \cdot 10^{-7}$  и  $P_{3из7} = 3,05 \cdot 10^{-11}$ .

На основании полученных результатов можно сделать вывод о том, что вероятность наличия в слове двух ошибочных бит значительно превышает

вероятность наличия трех и более ошибок. В дальнейших расчетах будет рассматриваться только случай наличия в кодовом слове двух ошибочных бит. В таком случае формула (3.2) будет иметь следующий вид:

$$P_{BK} \approx PE_2. \quad (3.8)$$

Найдем вероятность ошибочного декодирования слова кода Хэмминга при наличии в нем двух ошибочных бит.

Примитивный декодер кода Хэмминга в случае обнаружения одиночной ошибки выдает трехбитовое слово (синдром), имеющее семь отличных от нуля значений, указывающих место ошибочного бита соответственно в одном из семи бит кодового слова. Нулевое значение синдрома указывает на отсутствие ошибок. При наличии более одного ошибочного бита синдром указывает неверное расположение ошибочного бита в кодовом слове, и вместо исправления ошибки (путем инверсии бита) в кодовом слове появляется третий ошибочный бит. При наличии двух ошибочных бит в проверочной части кодового слова (которая после декодирования отбрасывается) в процессе декодирования третий ошибочный бит появляется в информационной части слова. То есть возникновение в канале передачи данных двух ошибок в кодовом слове всегда приводит к ошибочному декодированию информации. Тогда:

$$PE_2 = P_{2из7}. \quad (3.9)$$

Из вышесказанного также следует наличие в кодовом слове на выходе процедуры декодирования трех ошибочных бита. В среднем, в информационной части слова будет содержаться  $3 \cdot (k/n)$  ошибочных бит:

$$K_E = 3 \cdot (k/n) = 3 \cdot (4/7) = 1,71. \quad (3.10)$$

Приведенные рассуждения для кода с параметрами  $k = 4$ ,  $n = 7$  справедливы и для всех других кодов Хемминга. С учетом сказанного, подставив в формулу (3.4) формулы (3.3), (3.8), (3.9) и (3.10), формула для вычисления вероятности появления ошибочного бита в декодированном информационном сообщении такова:

$$p_D = (P_{2изn} \cdot 3 \cdot (k/n)) : k = P_{2изn} \cdot (3/n). \quad (3.11)$$

В развернутом виде формулу (3.11) можно представить, как:

$$p_D = \frac{3 \cdot (n-1)!}{2! \cdot (n-2)!} p_B^2 (1 - p_B)^{n-2}. \quad (3.12)$$

Полученные с использованием формулы (3.12) результаты приведены в таблице 3.3 и на рисунке 3.2.

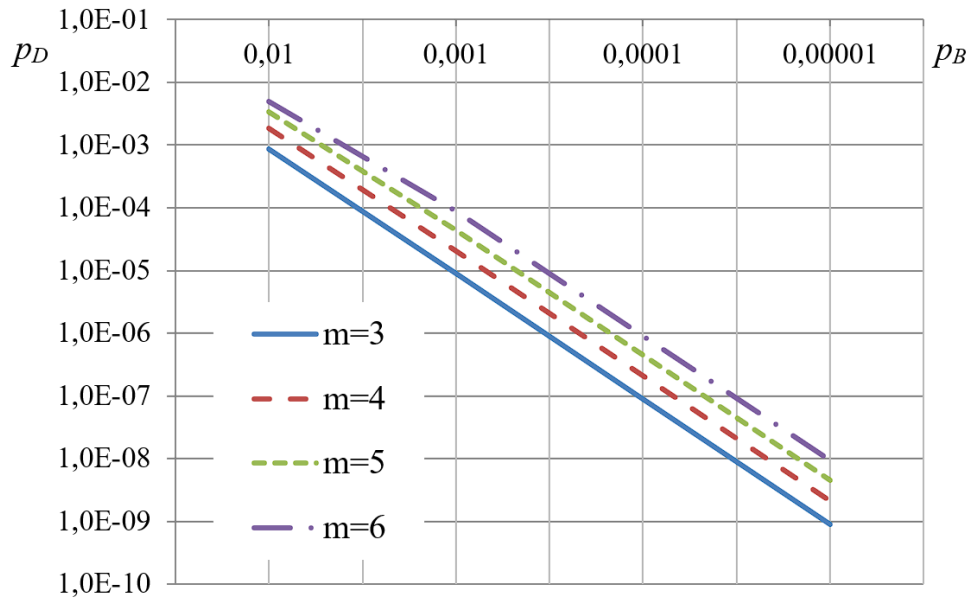


Рисунок 3.2. Сравнение эффективности кодов Хемминга

Таблица 3.3

Сравнение эффективности кодов Хемминга

№	$p_B$	Применяемый код	$p_D$	$k/n$
1.	$1 \cdot 10^{-2}$	$n=7, k=4, m=3$	$8,56 \cdot 10^{-4}$	0,57
2.		$n=15, k=11, m=4$	$1,84 \cdot 10^{-3}$	0,74
3.		$n=31, k=26, m=5$	$3,36 \cdot 10^{-3}$	0,84
4.		$n=63, k=57, m=6$	$5,04 \cdot 10^{-3}$	0,90
5.	$1 \cdot 10^{-3}$	$n=7, k=4, m=3$	$8,96 \cdot 10^{-6}$	0,57
6.		$n=15, k=11, m=4$	$2,07 \cdot 10^{-5}$	0,74
7.		$n=31, k=26, m=5$	$4,37 \cdot 10^{-5}$	0,84
8.		$n=63, k=57, m=6$	$8,75 \cdot 10^{-5}$	0,90
9.	$1 \cdot 10^{-4}$	$n=7, k=4, m=3$	$9,00 \cdot 10^{-8}$	0,57
10.		$n=15, k=11, m=4$	$2,10 \cdot 10^{-7}$	0,74
11.		$n=31, k=26, m=5$	$4,49 \cdot 10^{-7}$	0,84
12.		$n=63, k=57, m=6$	$9,24 \cdot 10^{-7}$	0,90
13.	$1 \cdot 10^{-5}$	$n=7, k=4, m=3$	$9,00 \cdot 10^{-10}$	0,57
14.		$n=15, k=11, m=4$	$2,10 \cdot 10^{-9}$	0,74
15.		$n=31, k=26, m=5$	$4,50 \cdot 10^{-9}$	0,84
16.		$n=63, k=57, m=6$	$9,29 \cdot 10^{-9}$	0,90

Наибольшей корректирующей способностью обладает код Хемминга, имеющий минимальную скорость. С ростом скорости (уменьшением избыточности) корректирующая способность кодов Хемминга снижается. Полученные результаты расчетов позволяют оценить степень ухудшения корректирующей способности кодов Хемминга при росте скорости кода. При этом скорость кода с ростом  $m$  стремится к 1, то есть, может варьироваться от 0,57 до 1 (в 1,75 раза).

### 3.2.2. Расчет корректирующей способности кода Голя

Проведем исследование корректирующей способности кода Голя.

Код Голя не способен обнаруживать неисправимые комбинации ошибок. Ошибочное декодирование кодового слова с небольшой долей приближения равно вероятности появления 4 ошибок из 23 бит:

$$P_{BK} = (23! / (4! \cdot (23 - 4)!)) \cdot p_B^4 \cdot (1 - p_B)^{23-4} = 8855 \cdot p_B^4 \cdot (1 - p_B)^{19}. \quad (3.13)$$

При ошибке декодирования кодовое слово переходит в другое, отстоящее на шесть бит. Тогда вероятность появления ошибочного бита в декодированном информационном сообщении равна:

$$p_D = (6/23) \cdot p_W = 2310 \cdot p_B^4 \cdot (1 - p_B)^{19}. \quad (3.14)$$

В результате вычисления по формуле (3.14) получены следующие результаты:

$$\text{при } p_B = 10^{-1}: p_D = 3,11 \cdot 10^{-2};$$

$$\text{при } p_B = 10^{-2}: p_D = 1,90 \cdot 10^{-5};$$

$$\text{при } p_B = 10^{-3}: p_D = 2,26 \cdot 10^{-9};$$

$$\text{при } p_B = 10^{-4}: p_D = 2,30 \cdot 10^{-13}.$$

### 3.2.3. Расчет корректирующей способности кода БЧХ

Рассчитаем вероятность появления ошибочного бита в декодированном информационном сообщении для кода для БЧХ ( $n = 31, k = 11, t = 5$ ). Ошибочное декодирование кодового слова равно вероятности появления 6 ошибок из 31 бита:

$$P_{BK} = (31! / (6! \cdot (31 - 6)!)) \cdot p_B^6 \cdot (1 - p_B)^{31-6} = 736281 \cdot p_B^6 \cdot (1 - p_B)^{25}. \quad (3.15)$$



Расстояние между кодовыми словами равно десяти, тогда при ошибке декодирования кодовое слово переходит в другое, отстоящее на десять бит. Тогда вероятность появления ошибочного бита в декодированном информационном сообщении равна:

$$p_D = (10/31) \cdot p_W = 237510 \cdot p_B^6 \cdot (1 - p_B)^{25}. \quad (3.16)$$

Результаты вычисления по данной формуле (3.16) приведены в таблице 3.4.

Таблица 3.4

Вероятность появления ошибочного бита в декодированном информационном сообщении

$p_B$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$
$p_D$	$1,71 \cdot 10^{-2}$	$1,85 \cdot 10^{-7}$	$2,32 \cdot 10^{-13}$	$2,37 \cdot 10^{-19}$

### 3.2.4. Расчет корректирующей способности кода Рида-Соломона

Проведем исследование эффективности кода Рида-Соломона при разной вероятности появления ошибочного бита в канале.

Для исследований выберем код Рида-Соломона с параметрами  $n = 9$ ,  $k = 5$ ,  $t = 2$ . Пусть размер символа кода равен одному байту, что удобно для ЭВМ. В результате кодирования каждое кодовое слово содержит пять информационных символов и четыре проверочных символа. Структура такого кодового слова, содержащего девять символов (72 бита), изображена на рисунке 3.3.

1 байт	2 байт	3 байт	4 байт	5 байт	6 байт	7 байт	8 байт	9 байт
Проверочные символы				Информационные символы				

Рисунок 3.3. Кодовое слово кода Рида-Соломона

Такой код способен исправить два любых ошибочных символа. Наличие трех ошибочных символов приводит к ошибке декодирования слова, за исключением случая, когда все три ошибочных бита находятся в проверочной части слова.

Вероятность ошибки декодирования слова приблизительно равна вероятности появления трех ошибочных бит в сообщении, при условии, что все три ошибочных бита расположены в разных байтах сообщения и хотя бы один из них находится в

информационной части слова. Разберем этот случай подробнее с точки зрения теории вероятности, исходя из заданной вероятности появления ошибочного бита в канале. Вероятность данного события  $P_{PC}$  равна сумме вероятностей следующих трех событий:

- 1) в проверочной части два «ошибочных» байта, в информационной – один;
- 2) в проверочной части один «ошибочный» байт, в информационной – два;
- 3) в проверочной части нет «ошибочных» байт, в информационной – три.

$$P_{PC} = P_4(2)P_5(1) + P_4(1)P_5(2) + P_4(0)P_5(3) = C_4^2 q^2 (1-q)^{4-2} \cdot C_5^1 q (1-q)^{5-1} + C_4^1 q (1-q)^{4-1} \cdot C_5^2 q^2 (1-q)^{5-2} + C_4^0 q^0 (1-q)^{4-0} \cdot C_5^3 q^3 (1-q)^{5-3} = \frac{4!}{2!2!1!4!} q^3 (1-q)^6 + \frac{4!}{1!3!2!3!} q^3 (1-q)^6 + \frac{5!}{2!3!} q^3 (1-q)^6 = 30q^3 (1-q)^6 + 40q^3 (1-q)^6 + 10q^3 (1-q)^6 = 80q^3 (1-q)^6, \quad (3.17)$$

где  $q$  – вероятность наличия в одном байте одного ошибочного бита. Найдем эту вероятность по формуле Бернулли:

$$q = P_8(1) = C_8^1 p_B (1 - p_B)^{8-1} = \frac{8!}{1!7!} p_B (1 - p_B)^7 = 8p_B (1 - p_B)^7. \quad (3.18)$$

Расстояние между кодовыми словами кода равно четырем байтам, то есть при ошибочном декодировании в кодовом слове будет четыре ошибочных байта. Очевидно, что в среднем в одном ошибочном байте имеется четыре ошибочных бита, тогда кодовое слово из 72 бит содержит 16 ошибочных бит. Следовательно, вероятность ошибочного бита в декодированном информационном сообщении равна:

$$p_D = (16/72) \cdot P_{PC} = 0,22 \cdot P_{PC}. \quad (3.19)$$

Результаты вычисления по формуле (3.19) приведены в таблице 3.5.

Таблица 3.5

Вероятность появления ошибочного бита в декодированном информационном сообщении

$p_B$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$
$p_D$	$5,54 \cdot 10^{-2}$	$4,44 \cdot 10^{-3}$	$8,44 \cdot 10^{-6}$	$9,11 \cdot 10^{-9}$

Проведем исследование эффективности работы кодера-декодера при наличии пакетов ошибок в принятом сообщении [151]. Рассмотрим вероятность

правильного декодирования сообщения при наличии пакетов ошибок разной длины.

Рассматриваемый код в состоянии исправить два любых искаженных символа в кодовом слове. Пакеты ошибок длиной менее 10 бит всегда затрагивают не более двух байт и успешно исправляются. При определенном расположении пакет ошибок длиной 10 бит уже может исказить более двух символов (смотри рисунок 3.4.), и если это затронет хотя бы один информационный бит, то декодер будет не в состоянии исправить ошибку. Вероятность этого события при сообщении длиной в одно кодовое слово равна 0,11. При дальнейшем увеличении длины пакета вероятность ошибки на выходе декодера будет возрастать. В случае, когда сообщение состоит из одного кодового слова, максимальная длина пакета, при которой возможно правильное декодирование, составит 32 бита. Случай, когда пакет ошибок полностью занимает всю проверочную часть кодового слова (4 байта), изображен на рисунке 3.5. Вероятность ошибки на выходе декодера в этом случае равна 0,975.

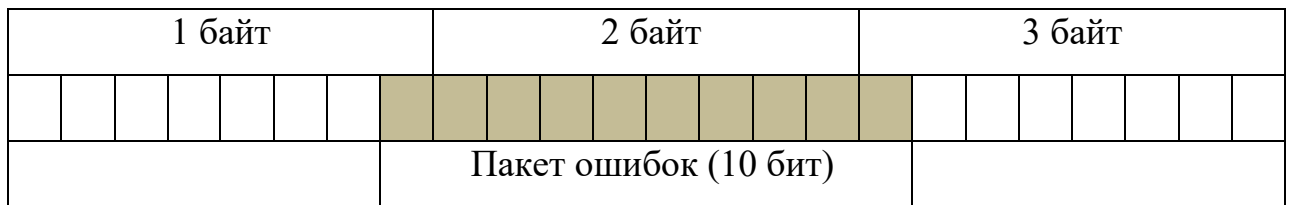


Рисунок 3.4. Воздействие пакета ошибок на 3 байта

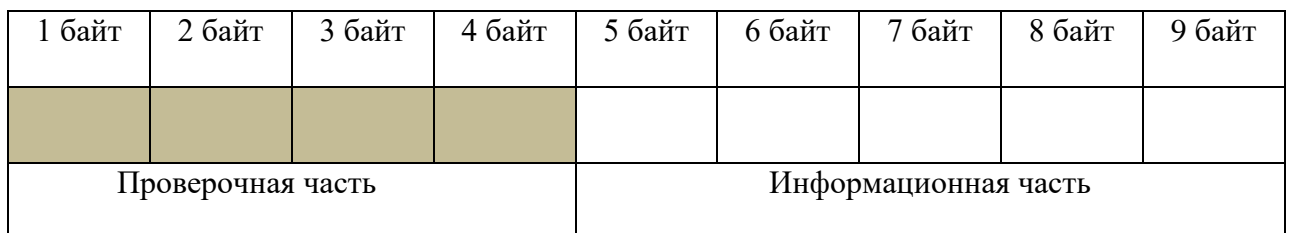


Рисунок 3.5. Пакет ошибок полностью занимает всю проверочную часть кодового слова

В случае сообщения длиной в 2 и более слов, максимальная длина пакета, при которой возможно правильное декодирование, составит 48 бит. Это случай, когда пакет ошибок полностью занимает всю проверочную часть кодового слова и 2

байта предыдущего слова (см. рис. 3.6.). Вероятность ошибки на выходе декодера при этом равна 0,988 для случая, когда сообщение состоит из 3 кодовых слов.

Рассчитаем вероятность появления пакета ошибок длиной 10 бит в случае наличия в канале передачи данных лишь независимых ошибок с вероятностью появления ошибочного бита  $p$ . Расчеты проведем для случая кодового сообщения длиной в четыре кодовых слова, то есть размером в 288 бит.

Вероятность того, что 10 бит из 10 будут ошибочными, равна:

$$w = P_{10}(10) = C_{10}^{10} p^{10} (1-p)^{10-10} = p^{10} \quad (3.20)$$

Кортеж из 10 ошибочных бит представим, как 1 условный «ошибочный бит». Тогда получаем в нашем сообщении 278 правильных бит и 1 «ошибочный». При этом необходимо найти вероятность того, что среди 279 бит будет 1 «ошибочный». Данная вероятность будет равна:

$$P_{279}(1) = C_{279}^1 w(1-w)^{279-1} = \frac{279!}{1!278!} w(1-w)^{278} = 279w(1-w)^{278}. \quad (3.21)$$

При  $p_B = 10^{-2}$ :  $w=10^{-20}$  и  $P_{279}=2,79 \cdot 10^{-18}$ .

При  $p_B = 10^{-3}$ :  $w=10^{-30}$  и  $P_{279}=2,79 \cdot 10^{-28}$ .

Полученные вероятности крайне малы, и ими можно пренебречь.

7 байт	8 байт	9 байт	1 байт	2 байт	3 байт	4 байт	5 байт	6 байт	7 байт	8 байт	9 байт
1 слово			2 слово								

Рисунок 3.6. Пакет ошибок полностью занимает всю проверочную часть кодового слова и два байта предыдущего кодового слова

Способные противостоять пакетам ошибок коды Рида-Соломона показали низкую эффективность в борьбе с независимыми ошибками, особенно при высокой вероятности появления ошибочного бита. При вероятности появления ошибочного бита в канале передачи данных равной  $1 \cdot 10^{-2}$  коды Хэмминга продемонстрировали более высокую корректирующую способность.

Одним из минусов блочных кодов является фиксированная длина блока. Данный недостаток особенно ощутим в случае изменения длины пакета передаваемых данных. Если длина пакета не делится на длину блока без остатка,

то в информационную последовательность приходится добавлять символы, не несущие информации, что приводит к снижению эффективности кодирования [85].

В параграфе произведен расчет корректирующей способности распространенных блочных кодов. Для случая двоичного блочного кода показана способность исправлять пакеты ошибок. Известным способом повышения эффективности помехоустойчивых кодов путем реализации адаптации параметров кода к изменениям характеристик канала передачи данных является перфорация. Рассмотрим применение перфорации в блочных кодах.

### **3.3. Реализация перфорации в блочных кодах**

О важности перфорации как способа повышения эффективности помехоустойчивых кодов было сказано в п. 1.6. В п. 2.4 были описаны особенности перфорации в сверточных кодах. Рассмотрим реализацию механизмов перфорации для блочных кодов [152]. Среди корректирующих кодов будут различаться механизмы перфорации для кодов, только исправляющих ошибки, и кодов, которые обнаруживают неисправимые ошибки и исправляют исправимые.

Коды Хемминга способны исправлять только один ошибочный бит. Выкалывание одного бита в кодовом слове кода Хемминга делает невозможным исправление даже одной ошибки. Рассмотренный в п. 3.2.3 вариант построения двоичного кода БЧХ способен исправлять пять любых ошибочных бит из тридцати одного. Для кода БЧХ возможна перфорация четырех бит из тридцати одного, что соответствует изменению кодовой скорости с 0,262 до 0,289, то есть на 10,5 %.

Рассмотренный в п. 3.2.4 вариант построения кода Рида-Соломона способен исправлять два любых ошибочных байта из девяти байт кодового слова. Для кода Рида-Соломона возможна перфорация одного байта из девяти, что соответствует изменению кодовой скорости с 0,555 до 0,625, то есть на 13 %.

Рассмотренный в п. 3.2.2 код Голея способен исправлять три любых ошибочных бита из двадцати трех бит кодового слова. Для кода Голея возможна

перфорация двух бит из двадцати трех, что соответствует изменению кодовой скорости с 0,52 до 0,57, то есть на 9,6 %.

На входе декодера перфорированным битам могут присваиваться случайные значения. В случае применения перфорации блочных кодов, способных не только исправлять, но и обнаруживать ошибки, то при обнаружении декодером неисправимого количества ошибок можно сделать попытку подбора значения выколотого бита. Вероятность появления ошибочного бита в информационном сообщении на выходе декодера является одной из важнейших характеристик помехоустойчивого кода. Можно сформулировать гипотезу, что применение способа подбора позволит снизить вероятность появления ошибочного бита в информационном сообщении на выходе декодера.

Далее в параграфе будет произведена оценка эффективности способа подбора путем сравнительной оценки значений вероятности битовой ошибки на выходе декодера для систематического двоичного блочного кода, исправляющего  $t$  и обнаруживающего  $t + 1$  ошибок. Опишем особенности процесса декодирования выбранного кода, которые будут учитываться в дальнейших расчетах. В случае, когда на вход декодера поступает  $t + 1$  ошибок, декодер сигнализирует о невозможности исправления ошибок, и декодирование завершается простым отбрасыванием проверочной части кодового слова. В случае, когда на вход декодера поступает число ошибок, превышающее  $t + 1$ , переданное кодовое слово переходит в соседнее, отличающееся на  $d$  бит, где  $d$  – кодовое расстояние между словами.

В качестве практического примера кода данного типа будем рассматривать расширенный код Голея с параметрами:  $n = 24$ ,  $k = 12$ ,  $d = 8$ , который способен гарантированно исправить  $t = 3$  любых ошибочных бита и сигнализировать о наличии четырех ошибочных бит.

Для сравнительной оценки эффективности способа подбора вначале произведем расчет вероятности появления ошибочного бита в декодированном информационном сообщении при отсутствии перфорации. Обозначим вероятность ошибки декодирования кодового слова как  $P_w$ . Вероятность ошибки в

информационном бите кодового слова, возникшей в случае ошибочного декодирования, обозначим как  $P_I$ . Тогда вероятность появления ошибочного бита в декодированном информационном сообщении равна:

$$p_D = P_W \cdot P_I. \quad (3.22)$$

Как было сказано ранее, ошибка декодирования кодового слова возникает при количестве ошибок равном  $t + 1$ , а вероятность появления большего числа ошибок значительно ниже. Тогда  $P_W$ , с небольшой долей приближения, вычисляется в соответствии с формулой Бернулли:

$$P_W = \frac{n!}{(t+1)!(n-t-1)!} p_B^{t+1} (1 - p_B)^{n-t-1}. \quad (3.23)$$

Найдем вероятность ошибки в информационном бите при ошибочном декодировании кодового слова. Как было сказано ранее, декодер, обнаружив неисправимое количество  $t + 1$  ошибок из  $n$  бит, просто отбрасывает проверочную часть кодового слова. Тогда  $P_I$  находится по формуле:

$$P_I = (t + 1) / n. \quad (3.24)$$

С учетом формул (3.23) и (3.24) формула для вычисления  $p_D$  имеет вид:

$$p_D = \frac{t+1}{n} \cdot \frac{n!}{(t+1)!(n-t-1)!} p_B^{t+1} (1 - p_B)^{n-t-1} = \frac{(n-1)!}{t!(n-t-1)!} p_B^{t+1} (1 - p_B)^{n-t-1}. \quad (3.25)$$

Результаты вычисления  $p_D$  по формуле (3.25) приведены в таблице 3.6.

Таблица 3.6

Результаты вычисления значений  $p_D$ 

$p_B$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
$p_D$	$1,45 \cdot 10^{-5}$	$1,74 \cdot 10^{-9}$	$1,77 \cdot 10^{-13}$	$1,77 \cdot 10^{-17}$

Далее рассмотрим, как выкалывание бит влияет на вероятность появления ошибочного бита в декодированном информационном сообщении с использованием подбора и без него.

Произведем расчет  $P_{DR}$  – вероятности появления ошибочного бита в декодированном информационном сообщении при наличии одного выколотого бита в проверочной части кодового слова и случайном выборе значения выколотого бита перед декодированием («0» или «1»). Если при случайном выборе

значение бита было угадано, то назовем этот случай «правильным выбором». Обозначим вероятность появления ошибочного бита в декодированном информационном сообщении в случае правильного выбора как  $P_{DR1}$ . Если при случайном выборе значение бита не было угадано (произошло появление дополнительной ошибки), назовем этот случай «ошибочным выбором». Обозначим вероятность появления ошибочного бита в декодированном информационном сообщении в случае ошибочного выбора как  $P_{DR2}$ . Так как вероятности правильного и ошибочного выбора одинаковы, то:

$$P_{DR} = (P_{DR1} + P_{DR2})/2. \quad (3.26)$$

Выведем формулу для нахождения  $P_{DR1}$ . Обозначим вероятность ошибки декодирования кодового слова как  $P_{WR1}$ , а вероятность ошибки в информационном бите кодового слова в случае ошибки декодирования как  $P_{IR1}$ . Тогда вероятность появления ошибочного бита в декодированном информационном сообщении равна:

$$P_{DR1} = P_{WR1} \cdot P_{IR1}. \quad (3.27)$$

$P_{WR1}$  равна вероятности появления в канале передачи данных  $t + 1$  ошибочных бит из  $n - 1$  и вычисляется аналогично формуле (3.23):

$$P_{WR1} = \frac{(n-1)!}{(t+1)!(n-t-2)!} p_B^{t+1} (1 - p_B)^{n-t-2}. \quad (3.28)$$

В случае правильного выбора значения выколото бита, из  $n - 1$  оставшихся бит мы имеем  $t + 1$  ошибок. Тогда формула для вычисления  $P_{IR1}$  будет иметь вид:

$$P_{IR1} = (t + 1)/(n - 1). \quad (3.29)$$

С учетом (3.28) и (3.29), формула для вычисления  $P_{DR1}$  имеет вид:

$$P_{DR1} = \frac{t+1}{n-1} \cdot \frac{(n-1)!}{(t+1)!(n-2-t)!} p_B^{t+1} (1 - p_B)^{n-2-t} = \frac{(n-2)!}{t!(n-2-t)!} p_B^{t+1} (1 - p_B)^{n-2-t}. \quad (3.30)$$

Выведем формулу для нахождения  $P_{DR2}$ . Обозначим вероятность ошибки декодирования кодового слова как  $P_{WR2}$ , а вероятность ошибки в информационном бите кодового слова в случае ошибки декодирования как  $P_{IR2}$ . Тогда вероятность появления ошибочного бита в декодированном информационном сообщении равна:

$$P_{DR2} = P_{WR2} \cdot P_{IR2}. \quad (3.31)$$



$P_{WR2}$  равна вероятности появления в канале передачи данных  $t$  ошибочных бит из  $n - 1$  и вычисляется аналогично формуле (3.23):

$$P_{WR2} = \frac{(n-1)!}{t!(n-1-t)!} p_B^t (1 - p_B)^{n-1-t}. \quad (3.32)$$

В случае ошибочного выбора значения выколотого бита из  $n - 1$  оставшихся бит мы имеем  $t$  ошибок. Тогда формула для вычисления  $P_{IR2}$  будет иметь вид:

$$P_{IR2} = t/(n - 1). \quad (3.33)$$

С учетом (3.32) и (3.33), формула для вычисления  $P_{DR2}$  имеет вид:

$$P_{DR2} = \frac{t}{n-1} \cdot \frac{(n-1)!}{t!(n-1-t)!} p_B^t (1 - p_B)^{n-1-t} = \frac{(n-2)!}{(t-1)!(n-1-t)!} p_B^t (1 - p_B)^{n-1-t}. \quad (3.34)$$

Результаты вычисления  $P_{DR}$  с использованием формул (3.26), (3.30) и (3.34) для расширенного кода Голя приведены в таблице 3.7.

Таблица 3.7

Результаты вычисления значений и  $P_{DR}$ 

$p_B$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
$P_{DR}$	$1,01 \cdot 10^{-4}$	$1,14 \cdot 10^{-7}$	$1,15 \cdot 10^{-10}$	$1,15 \cdot 10^{-13}$

Сравним результаты расчета вероятности появления ошибочного бита в декодированном информационном сообщении, приведенные в таблицах 3.6 и 3.7. Очевидно, что рост вероятности появления ошибочного бита в декодированном информационном сообщении вызван возросшей вероятностью ошибочного декодирования кодового слова при ошибочном выборе значения выколотого бита. Сравним значения вероятности ошибки декодирования кодового слова при правильном и ошибочном выборе значения выколотого бита, используя формулы (3.28) и (3.32) для расширенного кода Голя (см. таблицу 3.8).

Как видно из таблицы 3.8, с вероятностью более 95 % ошибочное декодирование кодового слова происходит при ошибочном выборе значения выколотого бита. Следовательно, при обнаружении  $t + 1$  ошибок после инверсии значения выколотого бита с вероятностью более 95 % будет произведено успешное декодирование.

Вероятности ошибки декодирования кодового слова

$p_B$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
$P_{WR1}$	$7,32 \cdot 10^{-5}$	$8,69 \cdot 10^{-9}$	$8,84 \cdot 10^{-13}$	$8,84 \cdot 10^{-17}$
$P_{WR2}$	$1,45 \cdot 10^{-3}$	$1,73 \cdot 10^{-6}$	$1,77 \cdot 10^{-9}$	$1,77 \cdot 10^{-12}$
$P_{WR2}/(P_{WR1}+P_{WR2})$	0,95	0,99	0,99	0,99

Произведем расчет  $P_{DS}$  – вероятности появления ошибочного бита в декодированном информационном сообщении при наличии одного выколотого бита, случайной замене выколотого бита перед декодированием, и инверсии значения выколотого бита при обнаружении декодером  $t + 1$  ошибок с последующим повторным декодированием.

Обозначим вероятность появления ошибочного бита в декодированном информационном сообщении в случае правильного выбора значения выколотого бита как  $P_{DS1}$ , а в случае ошибочного выбора значения выколотого бита как  $P_{DS2}$ . Так как вероятности правильного и ошибочного выбора одинаковы, то:

$$P_{DS} = (P_{DS1} + P_{DS2})/2. \quad (3.35)$$

Выведем формулу для нахождения  $P_{DS1}$ . При правильном выборе значения выколотого бита вероятность ошибочного декодирования кодового слова равна вероятности появления в канале передачи данных  $t + 1$  ошибочных бит из  $n - 1$  и вычисляется по формуле (3.28). В случае обнаружения  $t + 1$  ошибок декодер инвертирует выколотый бит, чем вносит дополнительную ошибку. Декодер не в состоянии обнаружить  $t + 2$  ошибок. При декодировании переданное кодовое слово переходит в другое, отличающееся на  $d$  бит. Вероятность появления ошибочного бита при декодировании кодового слова будет равна  $d/n$ . Исходя из сказанного, формула для вычисления  $P_{DS1}$  будет иметь вид:

$$P_{DS1} = \frac{d}{n} \cdot \frac{(n-1)!}{(t+1)!(n-2-t)!} p_B^{t+1} (1 - p_B)^{n-2-t}. \quad (3.36)$$

Выведем формулу для нахождения  $P_{DS2}$ . При наличии в кодовом слове после прохождения канала передачи данных  $t$  ошибок и ошибочном выборе значения выколотого бита декодер сигнализирует об обнаружении  $t + 1$  ошибок, после чего значение выколотого бита инвертируется и кодовое слово декодируется без

ошибок. При наличии в принятом кодовом слове  $t + 1$  ошибок и ошибочном выборе значения выколотого бита, декодером ошибки не обнаруживаются. Переданное кодовое слово переходит в другое, отличающееся на  $d$  бит. Тогда вероятность ошибочного декодирования кодового слова будет вычисляться по формуле (3.28), а вероятность появления ошибочного бита при декодировании кодового слова будет равна  $d/n$ . Очевидно, что в этом случае  $P_{DS2}=P_{DS1}$ , и  $P_{DS}$  вычисляется по формуле (3.36).

Результаты вычисления  $P_{DS}$  для расширенного кода Голя приведены в таблице 3.9.

Таблица 3.9

Результаты вычисления значений  $P_{DS}$ 

$p_B$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
$P_{DS}$	$2,44 \cdot 10^{-5}$	$2,89 \cdot 10^{-9}$	$2,95 \cdot 10^{-13}$	$2,96 \cdot 10^{-17}$

Сравнив значения вероятности появления ошибочного бита в декодированном информационном сообщении, приведенные в таблицах 3.6, 3.7 и 3.9, можно сделать вывод об эффективности способа подбора.

Рассмотрим, как выкалывание двух бит влияет на вероятность появления ошибочного бита в декодированном информационном сообщении с использованием подбора и без.

Обозначим как  $P_{2R}$  вероятность появления ошибочного бита в декодированном информационном сообщении при наличии двух выколотых бит и присвоении выколотым битам случайных значений («0» или «1») перед декодированием.

При случайной замене двух выколотых бит возможны четыре равновероятных варианта: удачная замена двух бит, удачная замена первого бита, удачная замена второго бита, неудачная замена двух бит. Вероятности появления ошибочного бита в декодированном информационном сообщении для первого случая обозначим как  $P_{2R1}$ , для второго и третьего случаев вероятности будут одинаковы и обозначены как  $P_{2R2}$ , для четвертого случая  $P_{2R4}$ . Тогда  $P_{2R}$  будет равно:

$$P_{2R} = (P_{2R1} + 2 \cdot P_{2R2} + P_{2R4})/4. \quad (3.37)$$

Вывод формул для расчета  $P_{2R}$  основан на применении формулы Бернулли и аналогичен произведенным ранее расчетам для случая выкалывания одного бита. Результаты вычисления  $P_{2R}$  для расширенного кода Голя приведены в таблице 3.10.

Таблица 3.10

Результаты вычисления значений  $P_{2R}$ 

$p_B$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
$P_{2R}$	$4,95 \cdot 10^{-4}$	$4,99 \cdot 10^{-6}$	$5,00 \cdot 10^{-8}$	$5,00 \cdot 10^{-10}$

Очевидно, что в большинстве случаев ошибочное декодирование кодового слова происходит при ошибочном выборе значений двух выколотых бит. Следовательно, при обнаружении  $t + 1$  ошибок, после инверсии значения выколотого бита с большой вероятностью будет произведено успешное декодирование. Далее оценим, насколько подбор значения выколотого бита снизит вероятность появления ошибочного бита в декодированном информационном сообщении.

Произведем расчет  $P_{2S}$  – вероятности появления ошибочного бита в декодированном информационном сообщении при наличии двух выколотых бит, присвоению выколотым битам перед декодированием случайных значений и инверсии значения первого выколотого бита при обнаружении декодером  $t + 1$  ошибок с последующим повторным декодированием.

При случайной замене двух выколотых бит возможны четыре равновероятных варианта замены: удачная замена двух бит, удачная замена первого бита, удачная замена второго бита, неудачная замена двух бит. Вероятности появления ошибочного бита в декодированном информационном сообщении для этих случаев обозначим как  $P_{2S1}$ ,  $P_{2S2}$ ,  $P_{2S3}$ ,  $P_{2S4}$ , соответственно. Так как все эти случаи равновероятны, то  $P_{2S}$  будет равно:

$$P_{2S} = (P_{2S1} + P_{2S2} + P_{2S3} + P_{2S4})/4. \quad (3.38)$$

Выведем формулу для нахождения  $P_{2S1}$ . В случае правильного выбора значений обоих выколотых бит вероятность ошибочного декодирования кодового

слова равна вероятности появления в канале передачи данных  $t + 1$  ошибочных бит из  $n - 2$ , которая вычисляется в соответствии с формулой Бернулли. В случае обнаружения  $t + 1$  ошибок декодер инвертирует первый выколотый бит и этим вносит дополнительную ошибку. Декодер не в состоянии обнаружить  $t + 2$  ошибок. При декодировании переданное кодовое слово переходит в другое, отличающееся на  $d$  бит. Исходя из сказанного, формула для вычисления  $P_{2S1}$  будет иметь вид:

$$P_{2S1} = \frac{d \cdot (n-2)!}{n \cdot (t+1)! (n-t-3)!} p_B^{t+1} (1 - p_B)^{n-t-3} . \quad (3.39)$$

Выведем формулу для нахождения  $P_{2S2}$ . В случае правильного выбора только значения первого выколотого бита и наличия в кодовом слове после прохождения канала передачи данных  $t$  ошибок, декодер сигнализирует об обнаружении  $t + 1$  ошибок (вероятность появления в канале передачи данных  $t$  ошибочных бит из  $n - 2$  вычисляется по формуле Бернулли). Значение первого выколотого бита инвертируется, из-за чего в кодовом слове образуется  $t + 2$  ошибок. Декодер не в состоянии обнаружить  $t + 2$  ошибок. При декодировании переданное кодовое слово переходит в другое, отличающееся на  $d$  бит. Исходя из сказанного, формула для вычисления  $P_{2S2}$  будет иметь вид:

$$P_{2S2} = \frac{d \cdot (n-2)!}{n \cdot t! (n-t-2)!} p_B^t (1 - p_B)^{n-t-2} . \quad (3.40)$$

Выведем формулу для нахождения  $P_{2S3}$ . В случае правильного выбора значения второго выколотого бита и наличия в кодовом слове после прохождения канала передачи данных  $t$  ошибок, декодер сигнализирует об обнаружении  $t + 1$  ошибок. Далее значение первого выколотого бита инвертируется, из-за чего в кодовом слове остается только  $t$  ошибок, и кодовое слово декодируется без ошибок. Ошибочное декодирование происходит при наличии в кодовом слове после прохождения канала передачи данных  $t + 1$  ошибочных бит из  $n - 2$  (вероятность появления в канале передачи данных  $t + 1$  ошибочных бит из  $n - 2$  вычисляется по формуле Бернулли). Декодер не в состоянии обнаружить  $t + 2$  ошибок. При декодировании переданное кодовое слово переходит в другое, отличающееся на  $d$  бит. Исходя из сказанного, формула для вычисления  $P_{2S3}$  совпадает с формулой (3.39).

Выведем формулу для нахождения  $P_{2S4}$ . В случае ошибочного выбора значений обоих выколотых бит и наличия в кодовом слове  $t - 1$  ошибочных бит из  $n - 2$ , декодер сигнализирует об обнаружении  $t + 1$  ошибок, после чего значение первого выколотого бита инвертируется, из-за чего в кодовом слове остается только  $t$  ошибок. В этом случае при повторном декодировании кодовое слово декодируется без ошибок. Ошибочное декодирование происходит при наличии в кодовом слове после прохождения канала передачи данных  $t$  ошибочных бит из  $n - 2$  (вероятность появления в канале передачи данных  $t$  ошибочных бит из  $n - 2$  вычисляется по формуле Бернулли). Декодер не в состоянии обнаружить  $t + 2$  ошибок. При декодировании переданное кодовое слово переходит в другое, отличающееся на  $d$  бит. Исходя из сказанного, формула для вычисления  $P_{2S4}$  совпадает с формулой (3.40).

Так как  $P_{2S1} = P_{2S3}$ , а  $P_{2S2} = P_{2S4}$ , формула (3.38) преобразуется следующим образом:

$$P_{2S} = (P_{2S1} + P_{2S2})/2. \quad (3.41)$$

Результаты вычисления  $P_{2S}$  для расширенного кода Голя приведены в таблице 3.11.

Таблица 3.11

Результаты вычисления значений  $P_{2S}$ 

$p_B$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
$P_{2S}$	$2,22 \cdot 10^{-4}$	$2,53 \cdot 10^{-7}$	$2,56 \cdot 10^{-10}$	$2,56 \cdot 10^{-13}$

Сравнение результатов вычисления  $P_{2R}$  и  $P_{2S}$ , приведенных в таблицах 3.10 и 3.11, еще раз подтверждает сделанный ранее вывод об эффективности способа подбора.

Коды, способные не только исправлять, но и обнаруживать ошибки, обладают большей избыточностью. Одним из постулатов теории помехоустойчивого кодирования является утверждение, что чем выше избыточность кода, тем выше его потенциальная корректирующая способность. Именно на этом была основана гипотеза предпринятых исследований. Гипотеза исследования подтверждена

результатами. Платой за использование подбора являются затраты времени на процедуру повторного декодирования. Повторное декодирование происходит только в случае обнаружения неисправимого количества ошибок, поэтому затраты времени на повторное декодирование в процентном отношении от общих затрат незначительны.

Сравним корректирующую способность перфорированного расширенного кода Голя и перфорированного сверточного кода, рассмотренного в п. 2.4. Сравним снижение значения вероятности битовой ошибки в декодированном информационном сообщении, полученное для расширенного кода Голя при выкалывании одного бита, с результатами, полученными для сверточного кода при перфорации каждого двадцать четвертого бита. Сравнив данные из таблицы 2.15 с результатами из таблиц 3.6, 3.7, 3.9, можно сделать следующие выводы: величина снижения корректирующей способности для сверточного кода и расширенного кода Голя, без подбора значения выколото бита, примерно одинакова. При применении подбора значения выколото бита снижение корректирующей способности более значительно для перфорированного сверточного кода.

Сравним снижение значения вероятности битовой ошибки в декодированном информационном сообщении, полученное для расширенного кода Голя при выкалывании двух бит, с результатами, полученными для сверточного кода при перфорации каждого двенадцатого бита. Сравнив данные из таблицы 2.15 с результатами из таблиц 3.6, 3.10, 3.11, можно сделать следующие выводы: величина снижения корректирующей способности для расширенного кода Голя более значительна как без подбора, так и при подборе значения выколото бит.

В параграфе произведено рассмотрение реализации перфорации блочных кодов. Для систематического двоичного блочного кода, исправляющего  $t$  и обнаруживающего  $t + 1$  предложен способ повышения корректирующей способности путем подбора значения выколото бита. При выкалывании одного бита применение метода подбора позволило снизить вероятность битовой ошибки в декодированном информационном сообщении в 46 раз (при  $p_B = 1 \cdot 10^{-3}$ ). При выкалывании двух бит применение метода подбора позволило снизить вероятность

битовой ошибки в декодированном информационном сообщении в 20 раз (при  $p_B = 1 \cdot 10^{-3}$ ). При перфорации блочных кодов изменение кодовой скорости в значительных пределах невозможно. При перфорации сверточных кодов изменение кодовой скорости возможно с 0,5 до 0,83 (на 66 %), при перфорации рассмотренных блочных кодов с 0,555 до 0,625 (на 13 %). Далее рассмотрим возможности по адаптации корректирующей способности при каскадном соединении блочных кодов.

### **3.4. Последовательное каскадное кодирование на основе внешнего кода Рида-Соломона**

Как упоминалось в п. 1.6, одним из способов повышения эффективности является соединение помехоустойчивых кодов в каскады.

Произведем оценку корректирующей способности последовательного каскадного кода. С этой целью разработан программный имитатор работы кодека на языке программирования C++ [153]. Программа обеспечивает пользователя возможностью оценить эффективность работы помехоустойчивого каскадного кода при использовании кода Рида-Соломона в качестве внешнего кода. Декодирование кода Рида-Соломона осуществляется с использованием известного алгоритма Берлекемпа-Месси [154, 155]. Программа позволяет:

- задавать тип внутреннего кода, выбирая код Хемминга или Голея;
- при отсутствии выбора внутреннего кода осуществлять кодирование-декодирование только кодом Рида-Соломона;
- задавать количество итераций кодирования-декодирования информационного сообщения;
- задавать вероятность появления ошибочного бита в закодированном сообщении, моделируя возникновение ошибок в канале передачи данных;
- выводить по итогам завершения всех итераций кодирования-декодирования общее число ошибочно декодированных байт.



Структура кодирования для случая выбора кода Хемминга ( $k = 4$ ,  $n = 7$ ) в качестве внутреннего кода приведена на рисунке 3.7.

Структура кодирования для случая выбора кода Голея в качестве внутреннего кода приведена на рисунке 3.8.

9 символов (байт) кодового слова РС																	
1 байт		2 байт		3 байт		4 байт		5 байт		6 байт		7 байт		8 байт		9 байт	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
18 информационных слов кода Хемминга																	

Рисунок 3.7. Структура последовательного соединения кода Рида-Соломона и кода Хемминга

9 символов (байт) кодового слова РС									
1 байт		2 байт		3 байт		4 байт		5 байт	
1		2		3		4		5	
6 информационных слов кода Голея									

Рисунок 3.8. Структура последовательного соединения кода Рида-Соломона и кода Голея

Результаты работы программного имитатора приведены в таблице 3.12 [156]. Сравнив результаты исследования корректирующей способности кода Рида-Соломона с расчетными данными, полученными в п. 3.2, при  $p_B < 1 \cdot 10^{-1}$  результаты практически совпадают. Существенное отклонение в результатах при  $p_B = 1 \cdot 10^{-1}$  объясняется тем, что при высокой вероятности появления ошибочного бита в канале передачи данных ошибка при декодировании кодового слова может быть вызвана наличием ошибок более чем в трех байтах, в то время как используемая формула для нахождения  $p_B$  верна при наличии трех ошибочных байт в кодовом слове.

Проанализировав результаты каскадного кодирования, можно прийти к выводу о значительном росте корректирующей способности соединения кодов. В качестве отрицательной стороны можно отметить значительное падение кодовой скорости. Для одиночного кода Рида-Соломона скорость равна 0,555, для

каскадного соединения с кодом Хемминга скорость равна 0,317, а для каскадного соединения с кодом Голя скорость равна 0,288.

Таблица 3.12

Вероятность битовой ошибки в декодированном информационном сообщении

Применяемый код	$p_B=10^{-1}$	$p_B=10^{-2}$	$p_B=10^{-3}$	$p_B=10^{-4}$
Рида-Соломона	$2,7 \cdot 10^{-1}$	$4,9 \cdot 10^{-3}$	$1,0 \cdot 10^{-5}$	$9,1 \cdot 10^{-9}$
Каскадное соединение кода Рида-Соломона и Хэмминга	$7,7 \cdot 10^{-2}$	$9,2 \cdot 10^{-7}$	$9,1 \cdot 10^{-13}$	$9,0 \cdot 10^{-19}$
Каскадное соединение кода Рида-Соломона и Голя	$4,6 \cdot 10^{-2}$	$2,8 \cdot 10^{-8}$	$2,5 \cdot 10^{-16}$	$2,4 \cdot 10^{-23}$

Полученное каскадное соединение позволяет в какой-то степени адаптировать параметры кода к характеристикам канала передачи данных. При благоприятном состоянии канала скорость передачи данных составляет 0,555, при ухудшении качества канала скорость можно снизить до 0,317, а затем и до 0,288. Недостатком такой адаптации является отсутствие плавной возможности изменения скорости и корректирующей способности кода. Имеется всего три градации.

Существуют каналы передачи данных с высокой степенью группирования ошибок [157, 158, 159]. Проведено исследование корректирующей способности кода при наличии пакетов ошибок с использованием программного имитатора.

Исследована эффективность работы кодера-декодера Рида-Соломона при наличии в канале пакетов ошибок. При размере пакета ошибок менее 10 бит декодер всегда проводит правильное декодирование. Правильное декодирование возможно с определенной вероятностью при размере пакета ошибок от 10 до 48 бит.

Исследована эффективность работы кодера-декодера последовательного каскадного соединения кода Рида-Соломона и кода Хемминга при наличии в канале пакетов ошибок. При размере пакета ошибок менее 18 бит декодер всегда проводит правильное декодирование. Правильное декодирование возможно с определенной вероятностью при размере пакета ошибок от 18 до 86 бит.

Исследована эффективность работы кодера-декодера последовательного каскадного соединения кода Рида-Соломона и кода Голея при наличии в канале пакетов ошибок. При размере пакета ошибок менее 8 бит декодер всегда проводит правильное декодирование. Правильное декодирование возможно с определенной вероятностью при размере пакета ошибок от 8 до 92 бит.

В исследованиях осуществлялось кодирование сообщения длиной 320 бит. При кодировании кодом Рида-Соломона длина кодового сообщения составила 576 бит. При кодировании кодом Рида-Соломона в каскаде с кодом Хемминга длина кодового сообщения составила 1008. При кодировании кодом Рида-Соломона в каскаде с кодом Голея длина кодового сообщения составила 1104. На рисунке 3.9. показана зависимость вероятности ошибочного декодирования сообщения  $P_{ош}$  для разных видов кодирования.

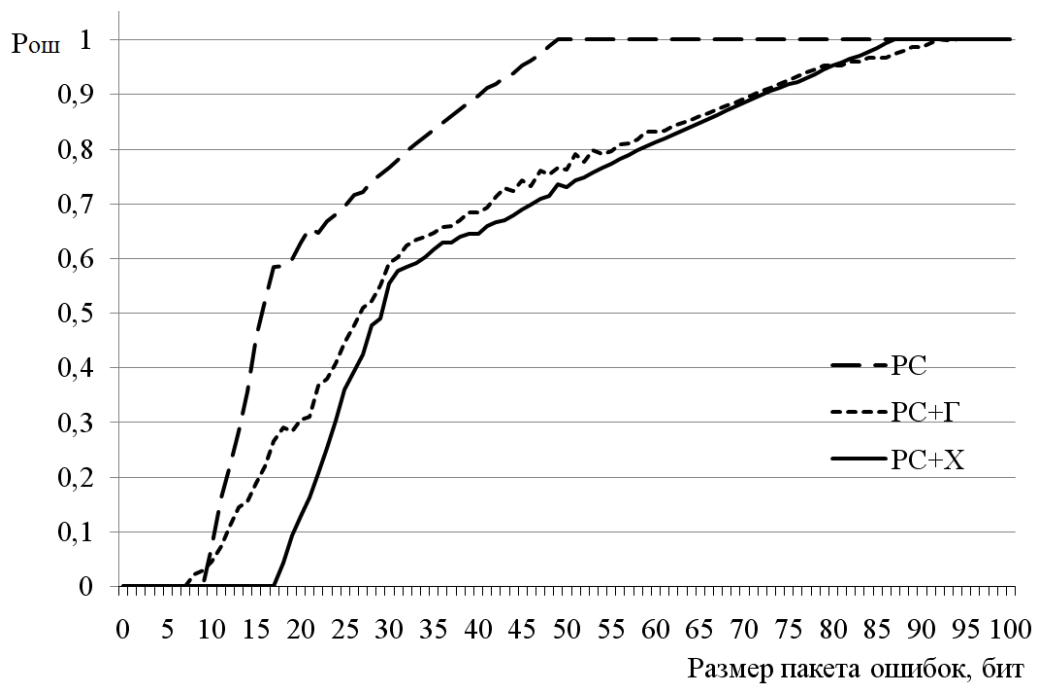


Рисунок 3.9. Зависимость вероятности приема ошибочного сообщения от величины пакета ошибок

В параграфе произведено рассмотрение возможности по адаптации корректирующей способности при каскадном соединении блочных кодов. Полученное каскадное соединение позволяет в какой-то степени адаптировать параметры кода к характеристикам канала передачи данных. При благоприятном

состоянии канала скорость передачи данных составляет 0,555, при ухудшении качества канала скорость можно снизить до 0,317, а затем и до 0,288. Недостатком такой адаптации является отсутствие плавной возможности изменения скорости и корректирующей способности кода, так как имеется всего три градации. Преимуществом использования каскадного соединения на основе внешнего кода Рида-Соломона является возможность исправления пакетов ошибок размером до 92 бит. При проведенном в главе 2 исследовании корректирующей способности сверточных кодов наблюдалось снижение вероятности битовой ошибки в декодированном информационном сообщении при декодировании мягкого решения детектора. Недостатком блочных кодов является сложность реализации декодирования мягкого решения детектора [11]. Рассмотрим реализацию мягкого декодирования для кода Рида-Соломона.

### **3.5. Мягкое декодирование кода Рида-Соломона**

Как было продемонстрировано в главе 2, декодирование мягкого решения детектора позволяет повысить корректирующую способность кода.

Повышением достоверности передачи информации в системах связи с помехоустойчивым кодированием возможно при применении мягких алгоритмов декодирования. Такие алгоритмы основаны на получении с демодулятора не бинарных сигналов в виде единиц и нулей, а в получении вероятности приема единицы или нуля. Применительно к коду Рида-Соломона, алгоритм мягкого декодирования может выглядеть так, как показано на рис. 3.10 [160, 161, 162].

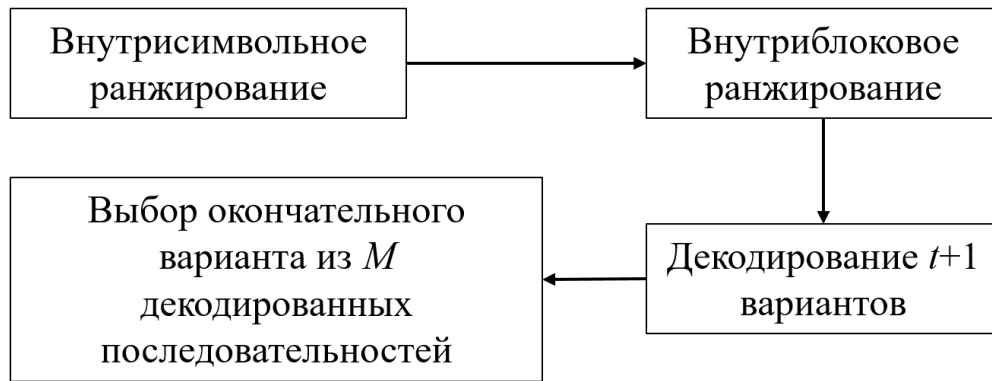


Рисунок 3.10. Алгоритм мягкого декодирования кода Рида-Соломона

На первом этапе оценивается достоверность принятия каждого символа путем перемножения вероятностей появления в нем единиц и нулей. Далее принятые символы ранжируются по убыванию достоверности их приема. Поскольку код Рида-Соломона позволяет исправлять стирания в количестве, равном используемой избыточности, декодеру предлагаются такие варианты кодовых слов, когда несколько символов с наименьшей достоверностью приема отмечены как стирания.

При этом количество вариантов кодовых слов, которые можно отправить на декодер, равно  $t + 1$ . Эти варианты различаются количеством отмеченных стираний, соответствующих символам с наихудшей достоверностью приема. В результате обработки декодером всех вариантов получится  $M$  декодированных вариантов информации, причем  $M > 1$  и  $M \leq t + 1$ . Эти варианты могут как совпадать, так и не совпадать. Существует несколько алгоритмов выбора окончательного варианта декодированной информации. Это может быть голосование по большинству как между вариантами кодовых слов в целом, так и между символами и даже битами информации. Может быть использован алгоритм вычисления метрики по принципу наибольшего правдоподобия (корреляции) в принятой кодовой последовательности. Применение алгоритмов мягкого декодирования позволяет повысить помехоустойчивость систем связи на  $1 \div 1,5$  дБ по соотношению сигнал/шум. Однако использование подобных алгоритмов имеет ряд недостатков. Во-первых, резко увеличиваются вычислительные затраты. Например, при избыточности  $2t = 20$ , имеется одиннадцать вариантов кодовых слов и декодеру необходимо провести одиннадцать ресурсоемких процедур

декодирования, а при жестком декодировании такая процедура всегда одна. Во-вторых, возрастают требования к объему оперативной памяти декодера, поскольку необходимо обрабатывать все варианты декодированной информации. В-третьих, возрастает сложность системы, поскольку необходимо проводить несколько процедур выбора и сортировки больших массивов данных. В-четвертых, не всегда удастся однозначно определить наилучший алгоритм окончательного выбора варианта декодированной информации. Для оптимизации мягкого алгоритма декодирования Рида-Соломона были проведены экспериментальные исследования. Цель исследований заключалась в определении в алгоритме мягкого декодирования, такого количества стираний, указанных декодеру, при котором достигается наивысший процент правильного распознавания информации. План экспериментов состоял в следующем:

- 1). Выбирались размерность символов  $m$ , длина информационного блока  $k$  и избыточность  $2t$ . Длина блока  $n = k + 2t$ .
- 2). Случайным образом генерировались информационные блоки.
- 3). Задаваясь соотношением сигнал/шум, генерировались мягкие решения демодулятора, таким образом, чтобы они не являлись ошибками в случае использования жестких решений.
- 4). Мягкие решения демодулятора корректировались таким образом, чтобы в случае использования жесткого декодера число ошибок было больше исправляющей способности кода Рида-Соломона при заданных  $n$  и  $k$ .
- 5). Согласно мягкому алгоритму декодирования, проводилось ранжирование символов по достоверности их приема.
- 6). Декодеру предлагалось  $t + 1$  вариантов кодовых слов с различным количеством стираний от 0 до  $2t$ .
- 7). Для каждого варианта кодового слова с различным числом стираний фиксировались правильные результаты декодирования согласно заданному блоку информации.

Результаты модельных экспериментов приведены на гистограммах на рисунках 3.11 и 3.12 при количестве ошибок больше исправляющей способности

жесткого алгоритма декодирования. Из графиков видно, что существует оптимальное количество стираний, при котором наблюдается низкий процент ошибочно декодированных блоков. Однако следует учитывать и тот факт, что оптимизированный алгоритм мягкого декодирования может допускать ошибки и при числе ошибок ниже исправляющей способности жесткого декодера. Соответствующая гистограмма приведена на рисунке 3.13. В этом случае можно предложить комбинированный алгоритм декодирования, при котором на первой стадии осуществляется жесткое декодирования, а в случае невозможности исправления ошибок использовать мягкое декодирование при фиксированном оптимальном количестве стираний.

В результате анализа экспериментальных данных можно сделать вывод, о целесообразности оптимизации алгоритма мягкого декодирования. Оптимизация заключается в определении такого количества символов с наихудшей достоверностью приема, при котором достигается наивысший процент правильного распознавания информации. Таким образом, декодер будет обрабатывать не  $t + 1$  вариантов кодовых слов, а всего один вариант, в котором отмечено число стираний, соответствующих наиболее достоверному распознаванию информации. Это позволит значительно снизить как вычислительные затраты, так и аппаратные, поскольку снижается как объем вычислений, так и объем требуемой оперативной памяти. Кроме того, снижается временная задержка при декодировании за счет значительного уменьшения многократных процедур декодирования.

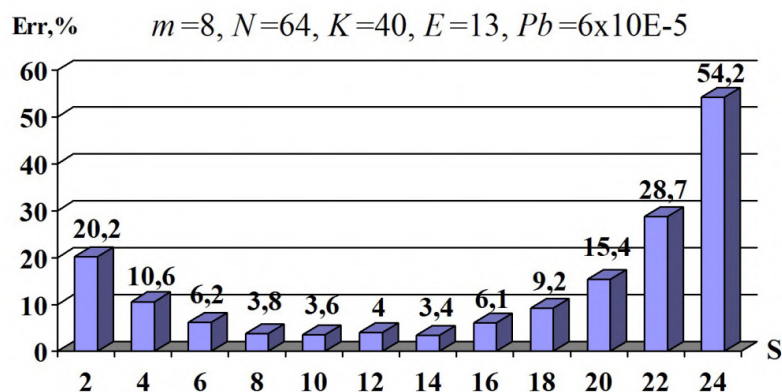


Рисунок 3.11. Результаты модельных экспериментов

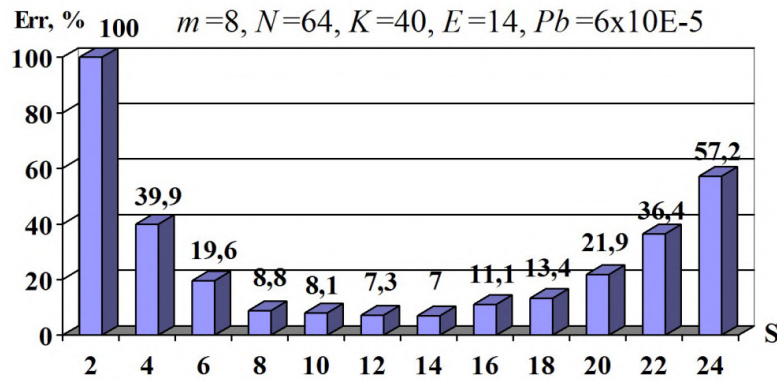


Рисунок 3.12. Результаты модельных экспериментов

Аппаратно жесткие кодеки представляют собой бинарный демодулятор, специализированные микросхемы кодирования и декодирования, микросхемы оперативной памяти и микросхемы управления, чаще всего представляющие собой микросхемы ПЛИС. Оптимизированный алгоритм мягкого декодирования потребует лишь аппаратной перестройки демодулятора, который будет выдавать не логические единицы и нули, а вероятности их появления в двоичном коде. Достаточно шестнадцати градаций вероятности принятия логической единицы, для организации алгоритма мягкого декодирования, т.е. для демодулятора достаточно четырехзначного двоичного кода. В остальном аппаратная перестройка будет заключаться в изменении алгоритма управления, т.е. в иной прошивке микросхемы ПЛИС, что не скажется на экономических затратах по производству кодеков.

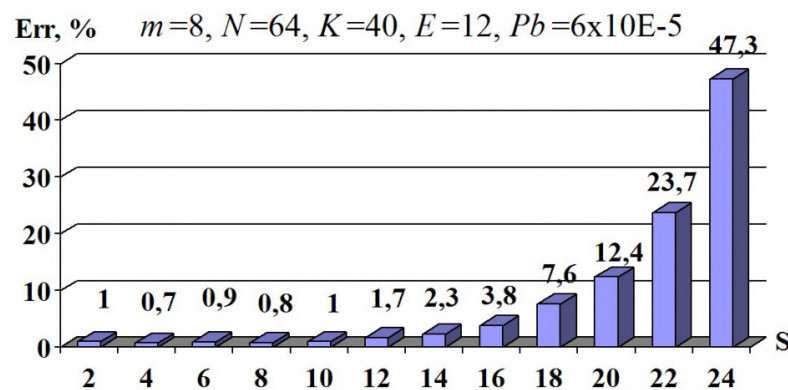


Рисунок 3.13 – Результаты модельных экспериментов

Таким образом, повышение достоверности передачи информации возможно за счет применения оптимизированных алгоритмов мягкого декодирования путем



незначительного изменения демодуляторов в приемных устройствах и видоизмененном алгоритме обработки данных жестких декодеров. Применение подобного оптимизированного мягкого декодера повысит достоверность передачи информации по соотношению сигнал/шум в пределах от 0,5 до 1 дБ. Далее рассмотрим возможности по мягкому декодированию двоичных блочных кодов.

### **3.6. Мягкое декодирование блочных кодов с использованием синдромной решетки**

В настоящее время большинство блочных кодов декодируется при подаче на вход декодера жесткого решения о значении принятого символа. Декодирование при использовании мягкого решения позволяет улучшить корректирующую способность кода, но, как правило, требует более сложных и требовательных к вычислительным ресурсам алгоритмов декодирования [131]. Алгоритм Витерби хорошо известен и применяется при декодировании сверточных кодов при поступлении от детектора как жестких, так и мягких решений о значении принятого символа и был подробно рассмотрен в п. 2.1. Применение алгоритма Витерби для декодирования блочных кодов возможно при использовании синдромной решетки блочного кода.

В параграфе будет рассмотрено декодирование линейного блочного кода с использованием синдромной решетки в соответствии с алгоритмом Витерби, а также проведено сравнение эффективности декодирования жесткого и мягкого решения детектора [163]. В качестве практического примера будет рассмотрено декодирование уже упоминавшегося в данной главе систематического кода Хемминга ( $n = 7, k = 4$ ).

Для выбранного кода Хемминга кодовое слово формируется при умножении информационных бит на генерирующую матрицу  $G$ , имеющую следующий вид [164]:

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3.42)$$

Коды Хемминга обычно декодируются с использованием проверочной матрицы  $H$ , которая для выбранного кода имеет вид [164]:

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}. \quad (3.43)$$

Наиболее распространен метод декодирования кода Хемминга, в процессе которого происходит умножение принятого искаженного кодового слова на матрицу  $H^T$ . Исходя из значения, полученного таким образом вектора (названного синдромом), определяется номер ошибочного бита. Данный метод декодирования позволяет исправить ошибку только в одном бите кодового слова.

Декодирования кода Хемминга можно также осуществить с использованием синдромной решетки. Структура синдромной решетки однозначно определяется полным набором кодовых слов или проверочной матрицей [10]. Максимальное число уровней синдромной решетки (состояний) равно  $\min\{2^k, 2^{n-k}\}$ . Синдромная решетка для выбранного кода Хемминга изображена на рисунке 3.14. В отличие от сверточного кода структура синдромной решетки кода Хемминга нерегулярна. Одна часть синдромной решетки относится к проверочным битам (три шага слева), другая часть (четыре шага справа) – к информационным битам.

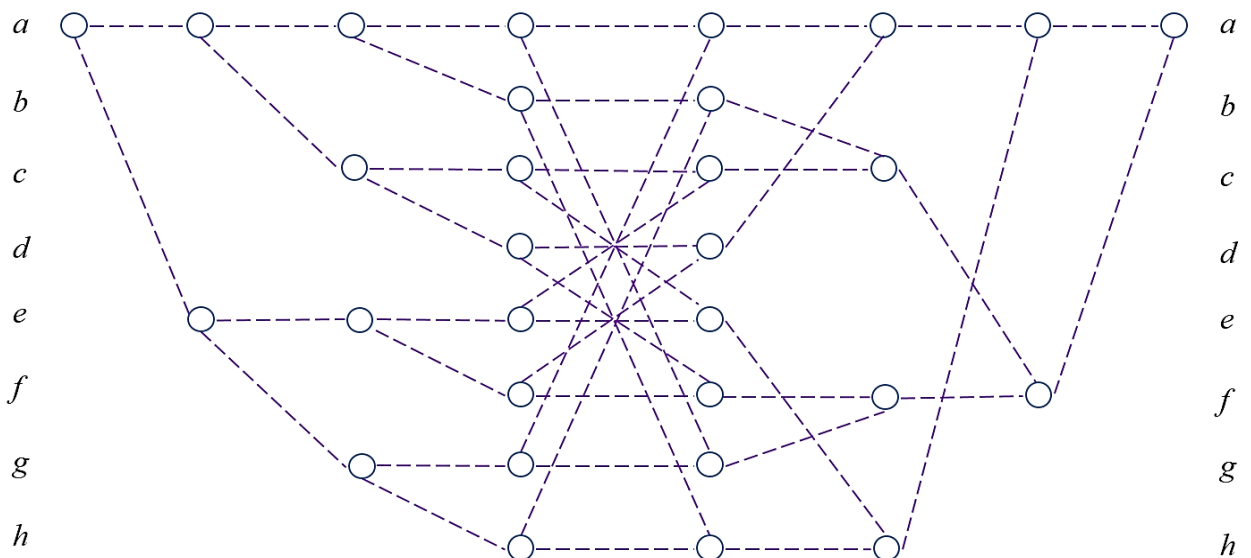


Рисунок 3.14. Структура синдромной решетки кода Хемминга

Процесс декодирования по синдромной решетке итеративен. На каждом шаге декодирования для каждой ветви решетки вычисляется метрика – число, характеризующее степень отличия значения ветви от значения принятого символа. При движении слева направо из ветвей образуется путь. Сумма метрик ветвей, входящих в путь, образует метрику пути. Если два пути входят в один узел решетки, то из них остается путь, имеющий меньшую метрику (выживший путь), а другой путь отбрасывается. После последнего шага декодирования остается один финальный выживший путь, однозначно определяющий декодированную последовательность. Таким образом, в соответствии с алгоритмом Витерби, из всех разрешенных последовательностей выбирается наиболее близкая к принятой кодовой последовательности.

При передаче сообщений биполярными сигналами ветвь, соединяющая узлы с одинаковыми состояниями, соответствует значению кодового бита «-1», а ветвь, соединяющая узлы с разными состояниями, соответствует значению кодового бита «1». Метрика ветви вычисляется как разность значения ветви и символа кодовой последовательности. На примере фрагмента синдромной решетки (см. рис. 3.15) рассмотрим расчет метрики ветви и вычисление метрики пути. На рисунке использованы следующие обозначения:  $\alpha$ ,  $\beta$ ,  $\delta$  – метрики путей, исходящих из узлов,  $\alpha'$ ,  $\beta'$  – метрики путей, входящих в узел, «-1» и «1» – значения ветвей,  $x$  – значение принятого кодового бита. На очередной итерации происходит приращение метрики пути на значение метрики ветви:  $\alpha' = \alpha + |x - 1|$ ,  $\beta' = \beta + |x + 1|$ . Если два пути с метриками  $\alpha'$  и  $\beta'$  входят в один узел решетки, то из них остается выживший путь, имеющий метрику  $\delta = \alpha'$  при  $\alpha' < \beta'$ , иначе  $\delta = \beta'$ .

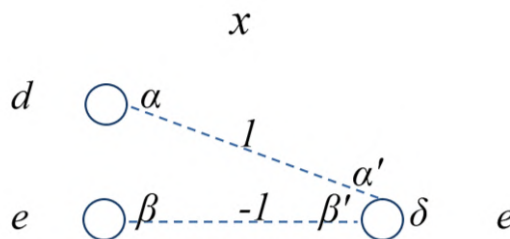


Рисунок 3.15. Фрагмент синдромной решетки кода Хемминга

Рассмотрим процесс декодирования конкретной кодовой последовательности. Пусть мы имеем информационную последовательность 0100, в процессе кодирования (при умножении вектора информационной последовательности на матрицу  $G$ ) формируется кодовая последовательность: 0110100. При передаче с использованием биполярных сигналов передаваемая последовательность будет иметь вид:  $-1\ 1\ 1\ -1\ 1\ -1\ -1$ . Декодирование с использованием синдромной решетки в случае принятия жесткого решения позволяет исправить один ошибочный бит, а в случае принятия мягкого решения позволяет исправить множественные ошибки. Подтвердим сказанное на примерах.

Рассмотрим декодирование принятой последовательности при принятии детектором жесткого решения о значении символа. Пусть жесткий выход детектора содержит одну ошибку (ошибочный символ подчеркнут) и имеет вид:  $-1\ 1\ 1\ -1\ \underline{-1}$   $-1\ -1$ . На рисунке 3.16 показан процесс декодирования. Входная последовательность приведена курсивом в верхней части рисунка. Остальные обозначения введены ранее. Финальный выживший путь изображен сплошной линией, и соответствующая ему последовательность полностью соответствует переданному кодовому слову, то есть не содержит ошибок.

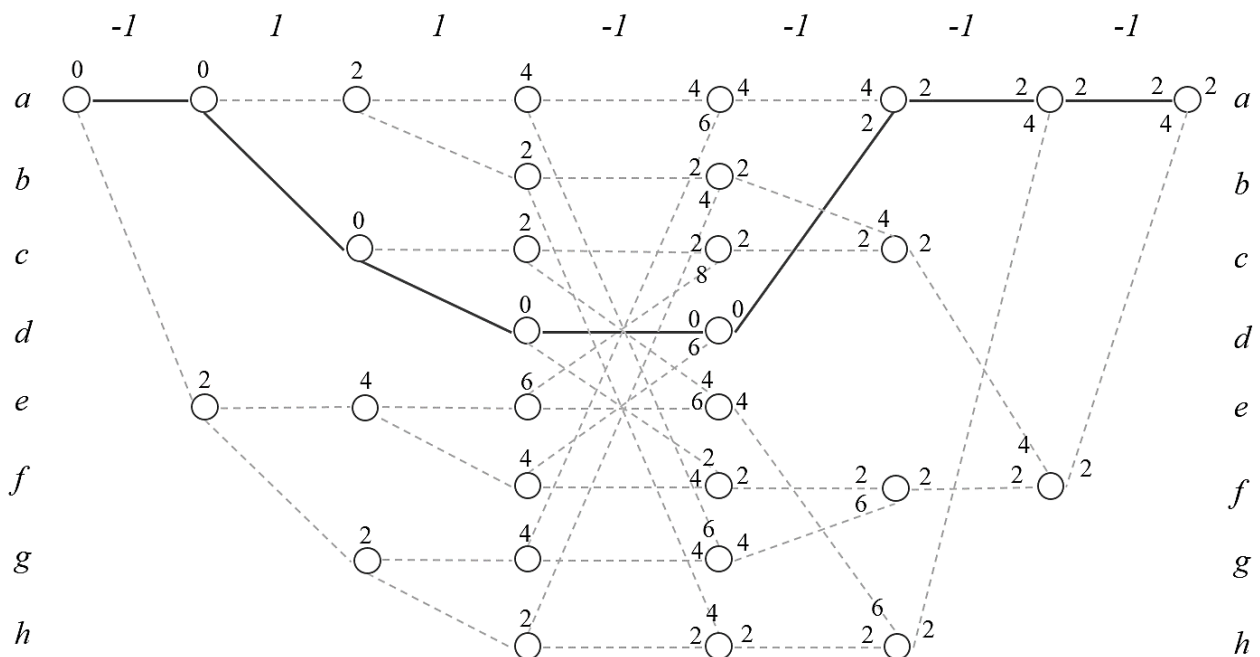


Рисунок 3.16. Декодирование жесткого решения

Рассмотрим случай, когда жесткий выход детектора содержит две ошибки и имеет вид:  $-1 \ 1 \ 1 \ -1 \ \underline{-1} \ \underline{1} \ -1$ . На рисунке 3.17 показан процесс декодирования. Финальный выживший путь изображен сплошной линией, и соответствующая ему последовательность  $\underline{1} \ 1 \ 1 \ -1 \ \underline{-1} \ \underline{1} \ -1$  отличается от переданного кодового слова в трех битах (подчеркнуты), при этом информационная часть последовательности содержит две ошибки.

Рассмотрим декодирование принятой последовательности при принятии детектором мягкого решения о значении символа. Пусть мягкий выход детектора, нормированный от  $-1$  до  $1$ , имеет вид:  $-0,6 \ 0,8 \ 0,4 \ \underline{0,2} \ \underline{-0,1} \ \underline{0,1} \ -0,5$  (жесткое решение в данном случае содержало бы три ошибочных бита). На рисунке 3.18 показан процесс декодирования мягкого выхода детектора. Финальный выживший путь определяет последовательность  $-1 \ 1 \ 1 \ -1 \ 1 \ -1 \ -1$ , полностью соответствующую переданному кодовому слову. Все ошибки исправлены.

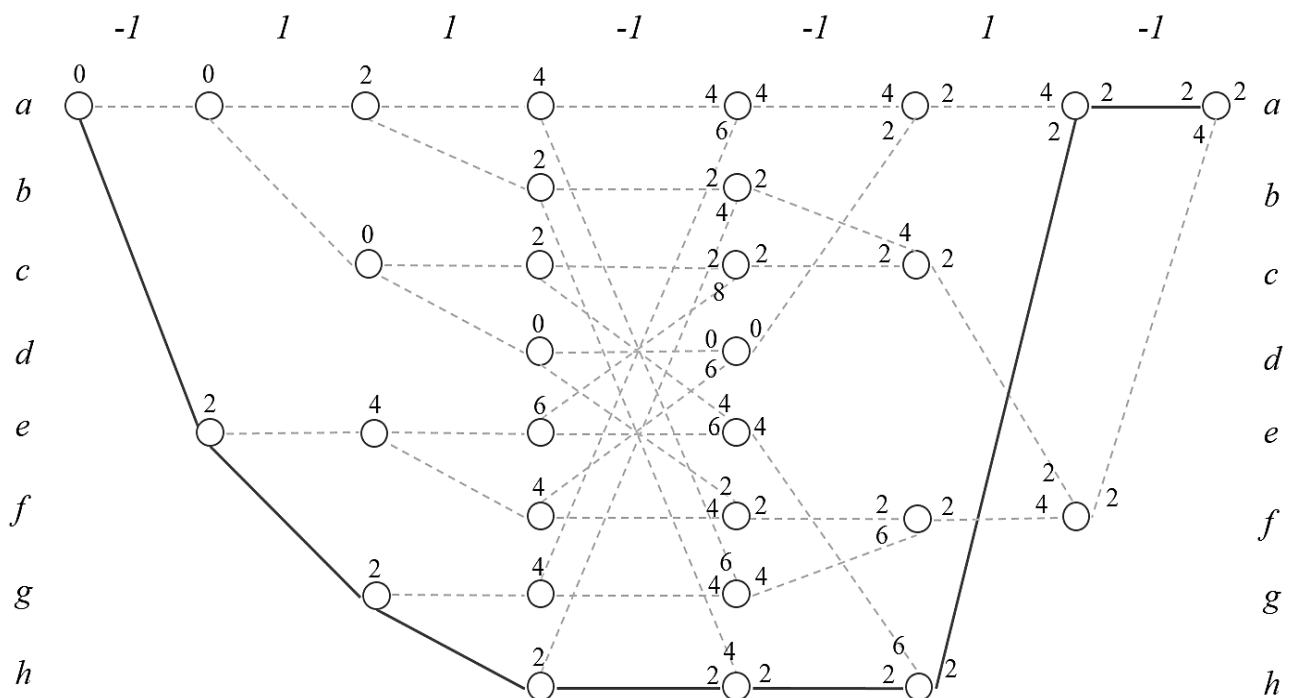


Рисунок 3.17. Декодирование жесткого решения

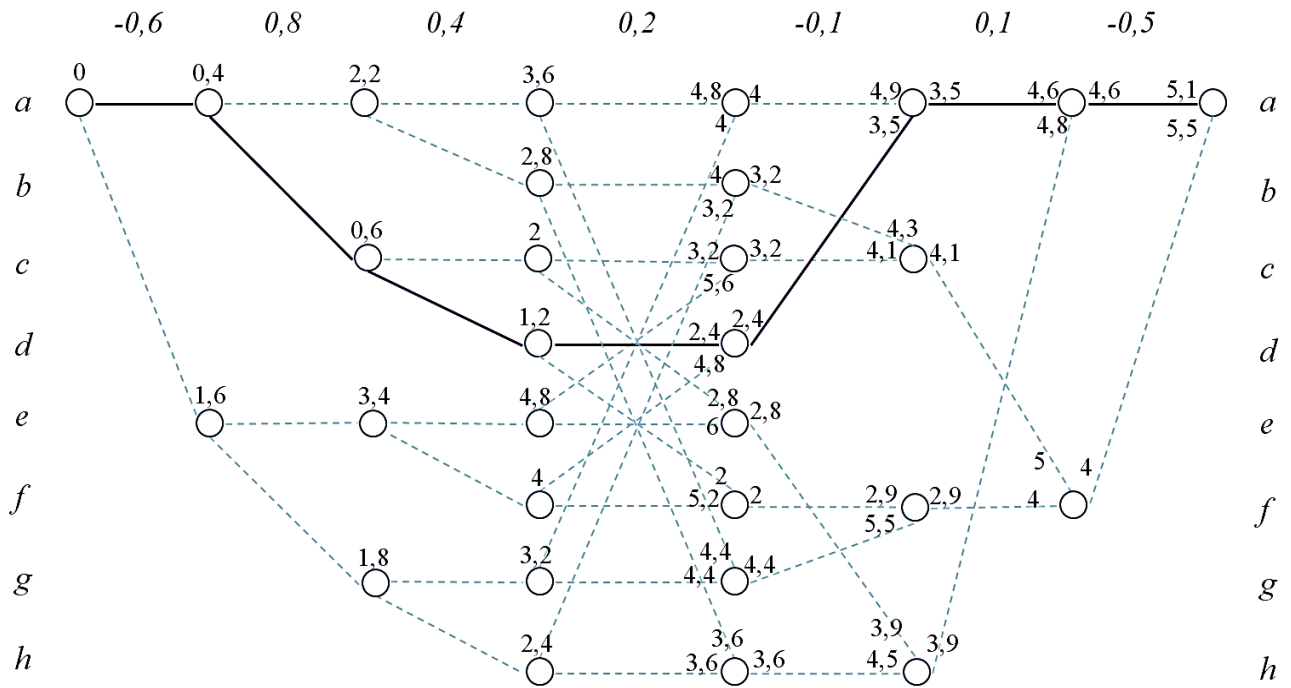


Рисунок 3.18. Декодирование мягкого решения

С целью исследования корректирующей способности кода Хемминга на языке программирования Python создан программный имитатор. Программный имитатор реализует следующее:

- генерацию информационной последовательности;
- кодирование кодом Хемминга;
- введение выкалываний в кодовую последовательность (см. п. 3.7);
- моделирование жесткого или мягкого выхода детектора приемника при заданной вероятностью появления ошибочного бита в канале  $p_B$ ;
- декодирование последовательности с использованием синдромной решетки;
- подсчет вероятности битовой ошибки в декодированной информационной последовательности.

Результаты работы программы при декодировании жесткого и мягкого решения детектора и разных значениях вероятности ошибочного бита в канале передачи данных приведены в таблице 3.13.

Подводя итоги результатов декодирования жесткого и мягкого решения детектора, можно резюмировать следующее. Декодирование жесткого решения соответствует расчетам, приведенным в п. 3.2. Декодирование мягкого решения с

использованием синдромной решетки существенно улучшает корректирующую способность кода, позволяя исправлять множественные ошибки в принятом кодовом слове. Недостатком декодирования с использованием синдромной решетки является необходимость хранения в памяти структуры решетки.

Таблица 3.13

Вероятность появления ошибочного бита в декодированном информационном сообщении при декодировании

Вид декодирования	$p_B = 1 \cdot 10^{-1}$	$p_B = 1 \cdot 10^{-2}$	$p_B = 1 \cdot 10^{-3}$	$p_B = 1 \cdot 10^{-4}$
Декодирование жесткого решения	$6,7 \cdot 10^{-2}$	$8,56 \cdot 10^{-4}$	$8,92 \cdot 10^{-6}$	$9,05 \cdot 10^{-8}$
Декодирование мягкого решения	$4,1 \cdot 10^{-2}$	$1,7 \cdot 10^{-4}$	$6,3 \cdot 10^{-7}$	$1,3 \cdot 10^{-9}$

В параграфе рассмотрена возможность мягкого декодирования блочного кода с использованием синдромной решетки. По итогам проведенных исследований зафиксировано снижение вероятности битовой ошибки в декодированном информационном сообщении при декодировании мягкого решения детектора – в 70 раз при  $p_B = 1 \cdot 10^{-4}$ . Далее рассмотрим мягкое декодирование с применением синдромной решетки перфорированного блочного кода.

### 3.7. Декодирование мягкого решения перфорированного кода Хемминга

В п. 3.3 проведено исследование корректирующей способности перфорированного блочного кода при декодировании жесткого решения детектора. Перфорация для кода Хемминга, способного при декодировании жесткого решения исправить только один ошибочный бит не целесообразна. Как было рассмотрено в п. 3.6, при декодировании мягкого решения детектора становится возможным исправление множественных ошибок в кодовом слове. Используя описанный вариант декодирования мягкого решения детектора, приведенный в п. 3.6, исследуем корректирующую способность перфорированного блочного кода при декодировании мягкого решения детектора для кода Хемминга.

Рассмотрим возможность декодирования перфорированного кодового слова с использованием синдромной решетки. В качестве кодовой последовательности используем последовательность, рассмотренную в п. 3.6:  $-1\ 1\ 1\ -1\ 1\ -1\ -1$ . Значение выколотого бита определим как нулевое.

Вначале разберем возможность декодирования кодовой последовательности при выкалывании одного информационного бита и двух ошибках. Пусть мягкий выход детектора при этом имеет вид:  $-0,6\ 0,8\ 0,4\ \underline{0,2}\ 0\ \underline{0,1}\ -0,5$ . Процесс декодирования схематично показан на рисунке 3.19. По результатам декодирования ошибки исправлены, декодирование произведено успешно.

Теперь рассмотрим возможность декодирования кодовой последовательности при выкалывании одного проверочного бита и двух ошибках. Пусть мягкий выход детектора при этом имеет вид:  $-0,6\ 0\ 0,4\ \underline{0,2}\ 0,8\ \underline{0,1}\ -0,5$ . Процесс декодирования схематично показан на рисунке 3.20. По результатам декодирования ошибки исправлены, декодирование произведено успешно.

Для исследования корректирующих возможностей перфорированного кода Хемминга при декодировании мягкого решения детектора будет использован программный имитатор, описанный в п. 3.6. Исследования показали, что лучшая корректирующая способность достигается при выкалывании проверочных бит кодового слова. Результаты работы программного имитатора при перфорации проверочных бит кодового слова и разных значениях вероятности появления ошибочного бита в канале передачи данных приведены в таблице 3.14. Проанализировав полученные результаты, можно сделать вывод о целесообразности перфорации не более чем двух бит в кодовом слове.

При декодировании мягкого решения стало возможным введение перфорации в код Хемминга. Корректировка кодовой скорости реализуется в пределах от 0,57 до 0,8 (на 40 %), что значительно превышает возможности по корректировке скорости при декодировании жесткого решения детектора (см. п. 3.3). Подробно рассмотрев особенности декодирования жестких и мягких решений детектора для блочных кодов, применим полученные знания с целью повышения эффективности последовательных каскадных кодов, созданных на основе внешнего блочного кода.



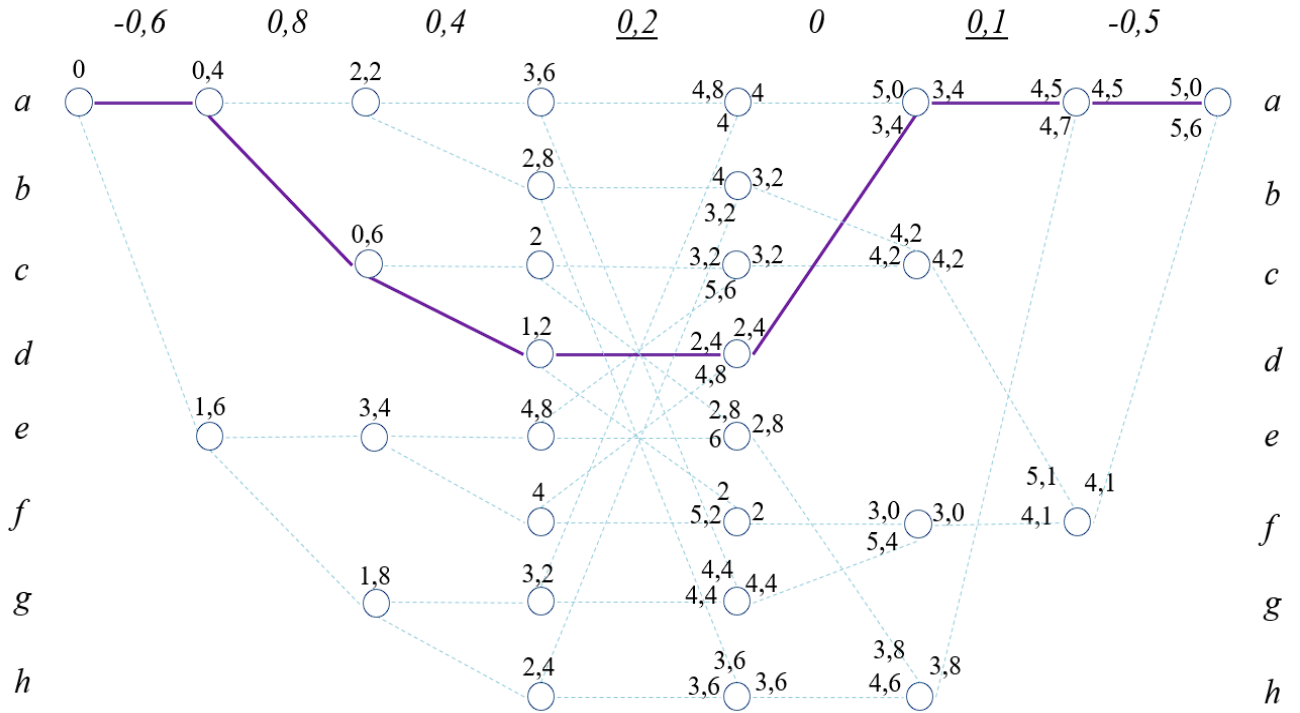


Рисунок 3.19. Декодирование мягкого решения

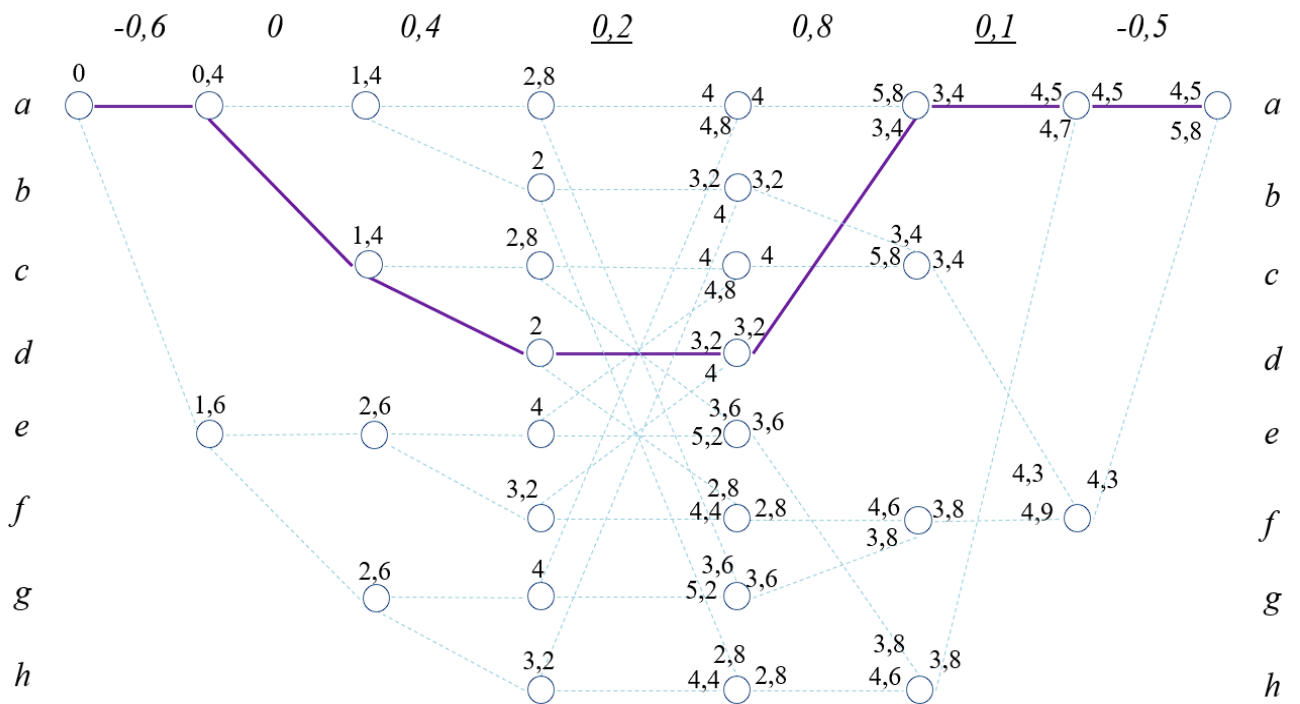


Рисунок 3.20. Декодирование мягкого решения

Вероятность битовой ошибки в декодированном информационном сообщении при перфорации кодового слова кода Хемминга

Число выколотых бит	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
0	$4,1 \cdot 10^{-2}$	$1,7 \cdot 10^{-4}$	$6,3 \cdot 10^{-7}$	$1,3 \cdot 10^{-9}$
1	$6,2 \cdot 10^{-2}$	$1,4 \cdot 10^{-3}$	$5,9 \cdot 10^{-5}$	$3,3 \cdot 10^{-6}$
2	$8,3 \cdot 10^{-2}$	$5,0 \cdot 10^{-3}$	$3,9 \cdot 10^{-4}$	$3,5 \cdot 10^{-5}$
3	$1,1 \cdot 10^{-1}$	$1,05 \cdot 10^{-2}$	$1,0 \cdot 10^{-3}$	$0,9 \cdot 10^{-4}$

### 3.8. Повышение адаптивных возможностей последовательного каскадного кода на основе внешнего блочного кода

В п. 2.6 был рассмотрен модифицированный метод последовательного каскадного кодирования, позволяющий регулировать и корректирующую способность ПКК путем изменения доли кодовых бит внешнего кода, кодируемых внутренним кодом. Таким образом было реализовано повышение эффективности последовательного каскадного кода на основе внешнего сверточного кода при применении избирательного кодирования внутренним кодом. Используя принципы и допущения, описанные в п. 2.6, в данном параграфе рассмотрим возможность реализации модифицированного метода последовательного каскадного кодирования, путем построения ГПКК на основе внешнего блочного кода при его декодировании согласно алгоритму Витерби [165]. По итогам декодирования ГПКК осуществим оценку корректирующей способности кода путем оценки значения вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$ . Исследования будут проводиться при разном соотношении количества бит, кодированных и не кодированных внутренним кодом. В качестве внешнего блочного кода будем использовать код Хемминга, возможность декодирования кодового слова которого с использованием алгоритма

Витерби рассмотрена в п. 3.6. В качестве внутреннего кода будем использовать код Голя.

Проведем исследования, дающие количественную информацию о значении  $p_D$  ГПКК в зависимости от следующих параметров:

- вероятности ошибки в канале передачи данных  $p_B$ ;
- вероятности ошибки на выходе декодера внутреннего кода  $p_V$ ;
- количества символов с выхода внутреннего кода, присутствующих во входной последовательности внешнего кода.

С целью реализации заявленных исследований на языке программирования Python создан программный имитатор, моделирующий работу ГПКК, с учетом принятых допущений [166]. Программный имитатор содержит в своем составе следующие основные компоненты:

- генератор информационной последовательности;
- кодер внешнего кода Хемминга;
- блок моделирования выхода детектора приемника с заданной вероятностью появления ошибочного бита  $p_B$ ;
- блок моделирования выхода декодера внутреннего кода с вероятностью появления ошибочного бита  $p_V$ ;
- декодер внешнего кода Хемминга.

Генерация информационной последовательности, генерация вектора ошибок и генерация мягкого выхода детектора подробно описаны в п. 2.2.

Результаты исследования ГПКК приведены в таблице 3.15. Количество бит кодового слова кодированных внутренним кодом обозначено в таблице как  $S_H$ . Проанализировав результаты, можно сделать следующие выводы. Получена возможность изменения корректирующей способности путем осуществления кодирования внутренним кодом только определенной части бит. Диапазон таких изменений увеличивается при снижении вероятности ошибочного бита в декодируемой внешним кодом последовательности.

Тип внутреннего кода целесообразно выбирать, исходя из прогнозируемых характеристик канала. В каналах, где периодически фиксируются случаи высокой

степени группирования ошибок, можно рекомендовать применение в качестве внутреннего кода – кода Рида-Соломона.

Таблица 3.15

Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Голея

$C_H$	$p_B=1 \cdot 10^{-1},$ $p_V=3,1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-2},$ $p_V=1,9 \cdot 10^{-5}$	$p_B=1 \cdot 10^{-3},$ $p_V=2,3 \cdot 10^{-9}$	$p_B=5 \cdot 10^{-4},$ $p_V=1,4 \cdot 10^{-10}$
3	$2,9 \cdot 10^{-2}$	$1,7 \cdot 10^{-4}$	$1,7 \cdot 10^{-6}$	$4,7 \cdot 10^{-7}$
2	$4,1 \cdot 10^{-2}$	$3,5 \cdot 10^{-4}$	$3,5 \cdot 10^{-6}$	$8,1 \cdot 10^{-7}$
1	$5,3 \cdot 10^{-2}$	$6,7 \cdot 10^{-4}$	$7,0 \cdot 10^{-6}$	$1,52 \cdot 10^{-6}$

Для улучшения восприятия, данные из таблицы 3.15 представлены в графическом виде на рисунке 3.21.

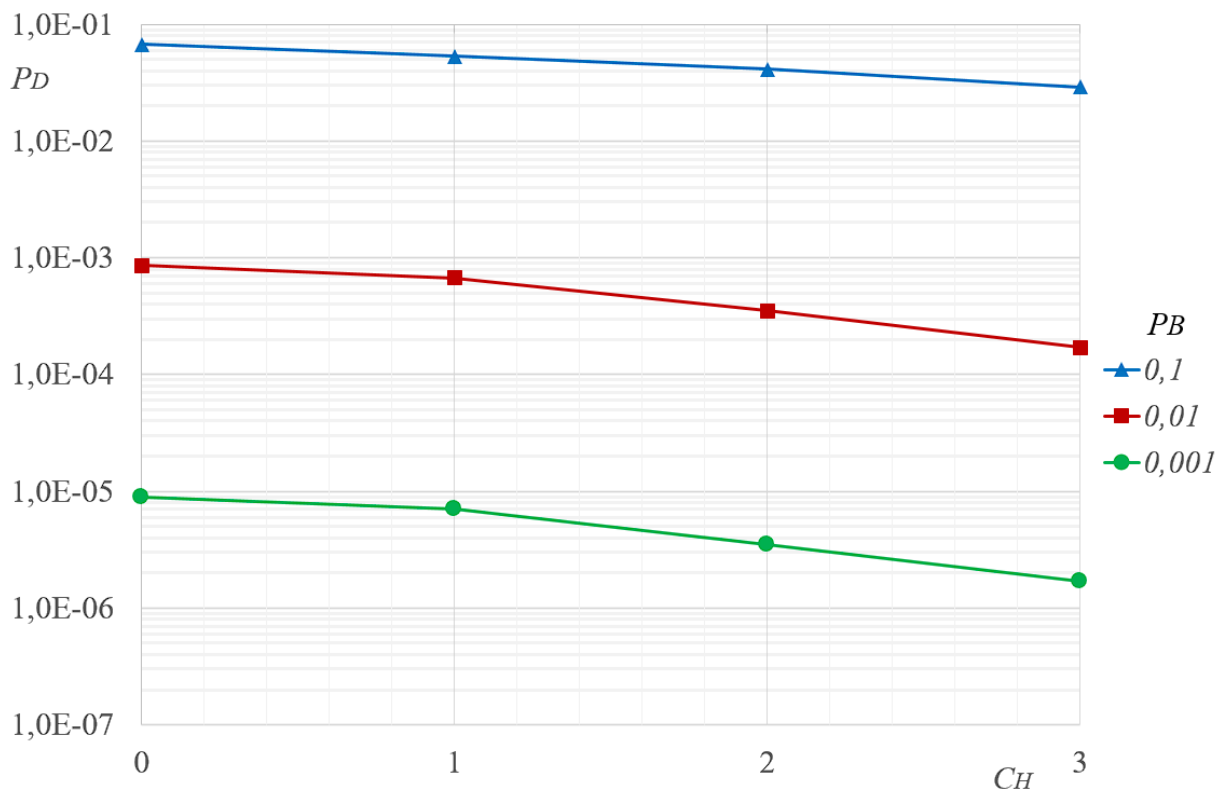


Рисунок 3.21. Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Голея

В параграфе рассмотрен модифицированный метод последовательного кодирования, обеспечивший возможность повышения адаптивности последовательного каскадного кода, созданного на основе внешнего блочного

кода. Далее рассмотрим, возможность повышения корректирующей способности, основываясь на знаниях построения ГПКК.

### **3.9. Повышение корректирующей способности ГПКК при декодировании по максимуму правдоподобия с применением коэффициентов к значениям декодируемых бит**

В п. 2.7 была экспериментально подтверждена эффективность учета большей степени достоверности символов, дополнительно закодированных внутренним кодом путем умножения их значений на коэффициенты  $K > 1$  или путем умножения менее достоверных значений символов на  $K < 1$  при декодировании согласно принципу максимального правдоподобия. Рассмотрим аналогичную возможность при построении ГПКК на основе внешнего кода Хемминга [165].

На практическом примере покажем качественную возможность повышения эффективности декодирования при использовании для кодовых бит описанных коэффициентов. Покажем, как применение коэффициентов к рассмотренной последовательности, содержащей две ошибки, приведет к повышению корректирующей способности. Продемонстрируем процесс декодирования кода согласно алгоритму Витерби для конкретной кодовой последовательности. Для примера используем кодовое слово кода Хемминга, формирование которого описано в п. 3.6, имеющее вид:  $-111-11-1-1$ . Известно, что при декодировании кода Хемминга в случае принятия детектором жесткого решения о значении принятого символа возможно исправление только одного ошибочного бита [10, 12]. Пусть последовательность на входе декодера содержит две ошибки и имеет вид:  $-111\underline{1}1\underline{1}-1$ . На рисунке 3.22 схематично показан процесс декодирования. Путь по решетке, соответствующий переданной кодовой последовательности, изображен жирной сплошной линией. Путь, имеющий минимальную метрику, изображен жирной пунктирной линией, и

соответствующая ему последовательность  $-11-1111-1$  отличается от переданного кодового слова, что ведет к ошибочному декодированию.

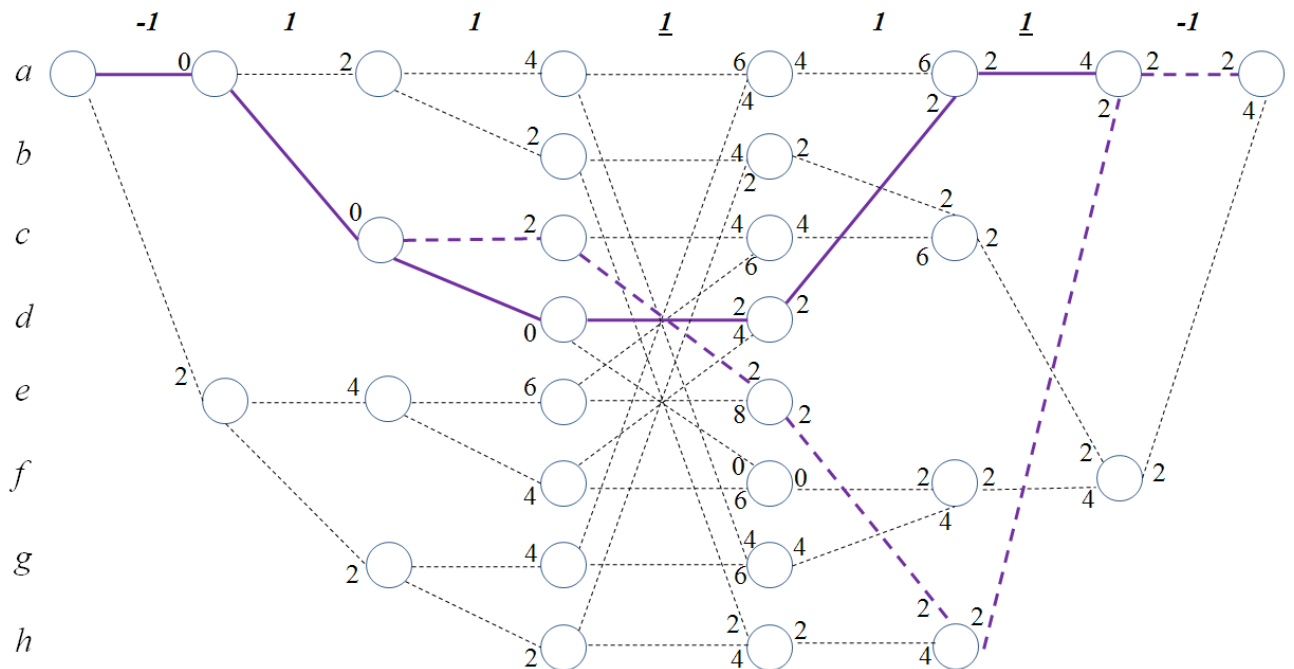


Рисунок 3.22. Декодирование при наличии двух ошибок.

Продemonстрируем, как наличие символов с выхода внутреннего декодера во входной последовательности внешнего декодера, и применение коэффициентов, повлияют на корректирующую способность. Пусть первые три бита кодового слова поступают с выхода декодера внутреннего кода, тогда для менее достоверных бит применим умножение их значений на понижающий коэффициент  $K = 0,4$ . Рассмотренная выше кодовая последовательность, содержащая две ошибки в информационных битах  $-1\ 1\ 1\ \underline{1}\ 1\ \underline{1}-1$ , с учетом коэффициентов будет иметь следующий вид:  $-1\ 1\ 1\ \underline{0,4}\ 0,4\ \underline{0,4}-0,4$ . Процесс декодирования схематично показан на рисунке 3.23. Ошибки в двух битах кодовой последовательности исправлены.

Рассмотрев представленный пример декодирования, можно сделать вывод об определенных качественных изменениях в результатах декодирования ГПКК при изменении числа бит, кодированных внутренним кодом, и использовании коэффициентов.

Произведем исследования корректирующей способности ГПКК, аналогичные проведенным исследованиям в п. 3.8, при умножении значений, закодированных внутренним кодом бит, на коэффициенты. Исследования будут проводиться при разном соотношении количества бит, кодированных и не кодированных внутренним кодом, и разном значении коэффициента  $K$ . В отличие от рассмотренных примеров декодирования в программном имитаторе вместо умножения на понижающие коэффициенты значений символов, не подвергшихся кодированию внутренним кодом, было реализовано умножение на коэффициент  $K > 1$  значений символов, закодированных внутренним кодом. В качестве внутреннего кода будем использовать код Голя.

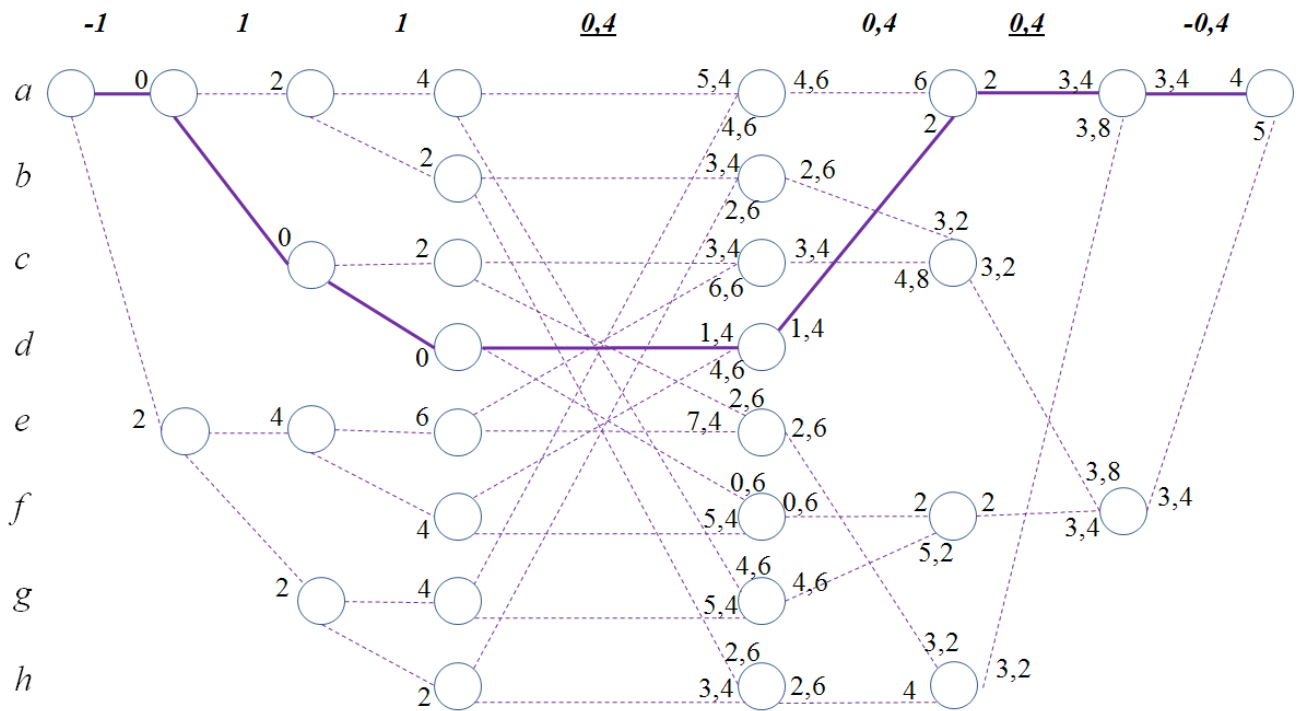


Рисунок 3.23. Успешное декодирование при наличии двух ошибок

Результаты исследования ГПКК приведены в таблице 3.16 [166]. В таблице в скобках приведены конкретные значения  $K$ , для которых получены наилучшие результаты корректирующей способности. Количество бит кодового слова кодированных внутренним кодом обозначено в таблице как  $S_H$ . Проанализировав результаты, можно сделать следующие выводы. Применение повышающего коэффициента к значениям символов, закодированных внутренним кодом,

приводит к росту корректирующей способности. При кодировании трех из семи бит кодового слова внутренним кодом Голя вероятность битовой ошибки снижается в 2,5 раза с  $4,7 \cdot 10^{-7}$  до  $1,9 \cdot 10^{-7}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ ). Эффект от применения коэффициентов увеличивается при снижении вероятности ошибочного бита в декодируемой последовательности, а также при увеличении разницы между значениями  $p_B$  и  $p_V$ .

Таблица 3.16

Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Голя

$C_H$	$p_B=1 \cdot 10^{-1},$ $p_V=3,1 \cdot 10^{-2}$		$p_B=1 \cdot 10^{-2},$ $p_V=1,9 \cdot 10^{-5}$		$p_B=1 \cdot 10^{-3},$ $p_V=2,3 \cdot 10^{-9}$		$p_B=5 \cdot 10^{-4},$ $p_V=1,4 \cdot 10^{-10}$	
	$K=1$	$K>1$	$K=1$	$K>1$	$K=1$	$K>1$	$K=1$	$K>1$
3	$2,9 \cdot 10^{-2}$	$2,7 \cdot 10^{-2}$ ( $K=1,3$ )	$1,7 \cdot 10^{-4}$	$9,7 \cdot 10^{-5}$ ( $K=4$ )	$1,7 \cdot 10^{-6}$	$6,8 \cdot 10^{-7}$ ( $K=5$ )	$4,7 \cdot 10^{-7}$	$1,9 \cdot 10^{-7}$ ( $K=5$ )
2	$4,1 \cdot 10^{-2}$	$3,9 \cdot 10^{-2}$ ( $K=1,2$ )	$3,5 \cdot 10^{-4}$	$2,5 \cdot 10^{-4}$ ( $K=5$ )	$3,5 \cdot 10^{-6}$	$2,2 \cdot 10^{-6}$ ( $K=5$ )	$8,1 \cdot 10^{-7}$	$4,92 \cdot 10^{-7}$ ( $K=5$ )
1	$5,3 \cdot 10^{-2}$	$5,28 \cdot 10^{-2}$ ( $K=1,2$ )	$6,7 \cdot 10^{-4}$	$5,0 \cdot 10^{-4}$ ( $K=1,2$ )	$7,0 \cdot 10^{-6}$	$3,7 \cdot 10^{-6}$ ( $K=5$ )	$1,52 \cdot 10^{-6}$	$1,28 \cdot 10^{-6}$ ( $K=5$ )

Для улучшения восприятия, данные из таблицы 3.16 представлена в графическом виде на рисунке 3.24. Пунктиром приведены графики для  $K = 1$ .

В параграфе рассмотрены изменения метода декодирования по максимуму правдоподобия при применении к значениям декодируемых бит коэффициентов, пропорциональных их достоверности. В результате чего наблюдался рост корректирующей способности ГПКК. При кодировании трех из семи бит кодового слова внутренним кодом Голя вероятность битовой ошибки снижается в 2,5 раза с  $4,7 \cdot 10^{-7}$  до  $1,9 \cdot 10^{-7}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ ), что в 368 раз меньше снижения вероятности битовой ошибки при построении ГПКК на основе внешнего сверточного кода. Такое преимущество ГПКК на основе внешнего сверточного кода, очевидно, обусловлено лучшей взаимосвязью бит кодового слова сверточного кода.



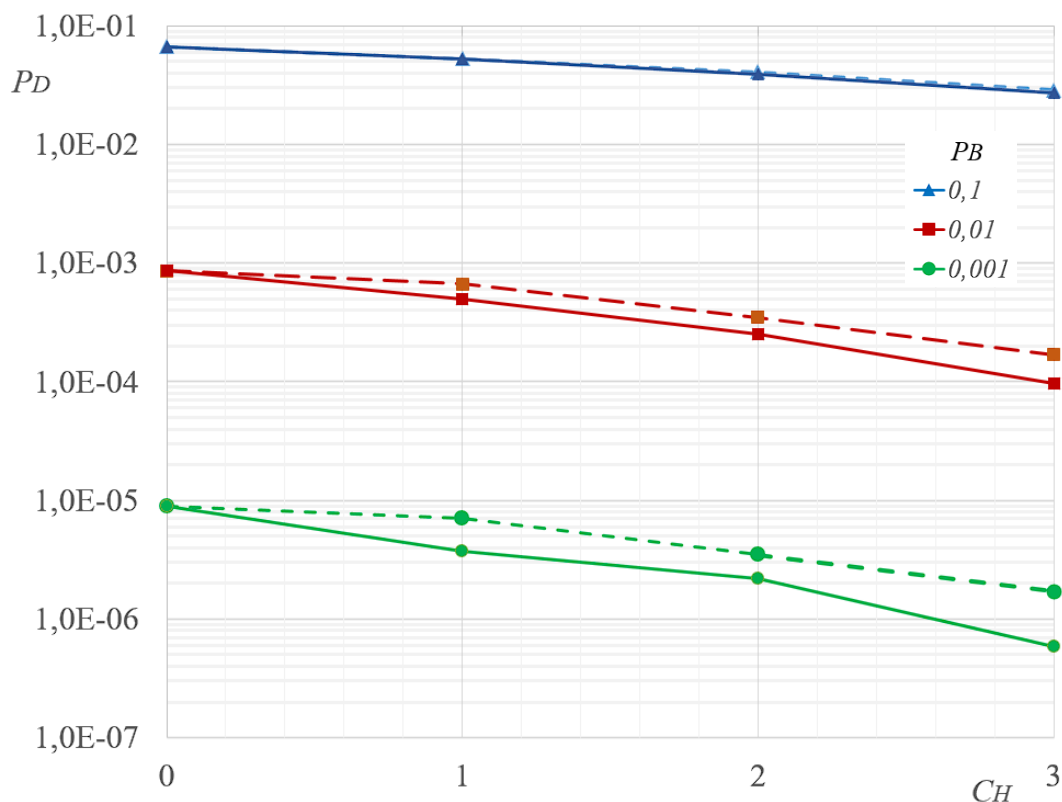


Рисунок 3.24. Вероятность появления ошибочного бита в декодированном информационном сообщении при использовании внутреннего кода Голя

### Выводы по главе 3

В главе рассмотрен ряд наиболее распространенных блочных кодов, упоминаемых в разных разделах диссертационного исследования. Осуществлен расчет корректирующей способности. Для случая недвоичного блочного кода произведена оценка способности к исправлению пакетов ошибок.

Рассмотрены возможности перфорации блочных кодов. При перфорации блочных кодов реализовать изменение кодовой скорости в значительных пределах затруднительно. При перфорации сверточных кодов (см. п. 2.5) изменение кодовой скорости возможно с 0,5 до 0,83 (на 66 %), при перфорации блочных кодов с 0,555 до 0,625 (на 13 %). Далее определены возможности по адаптации корректирующей способности при каскадном соединении блочных кодов.

Произведено изучение возможности по адаптации корректирующей способности при каскадном соединении блочных кодов. Рассмотренное каскадное соединение позволяет в какой-то степени адаптировать параметры кода к характеристикам канала передачи данных путем выбора того или иного внутреннего кода. При благоприятном состоянии канала скорость передачи данных составляет 0,555, при ухудшении качества канала скорость можно снизить до 0,317, а затем и до 0,288. Недостатком такой адаптации является отсутствие плавной возможности изменения скорости и корректирующей способности кода, так как имеется всего три градации. Преимуществом использования каскадного соединения на основе внешнего кода Рида-Соломона является возможность исправления пакетов ошибок размером до 92 бит. При проведенном в главе 2 исследовании корректирующей способности сверточных кодов, наблюдалось снижение вероятности битовой ошибки в декодированном информационном сообщении при декодировании мягкого решения. Недостатком блочных кодов является сложность реализации декодеров мягкого решения детектора. Рассмотрен вариант построения мягкого декодера для кода Рида-Соломона, обеспечивающего повышение достоверности передачи информации за счет применения оптимизированных алгоритмов мягкого декодирования и незначительного изменения демодуляторов в приемных устройствах. Рассмотренный вариант значительно усложняет и замедляет процесс декодирования, обеспечивая энергетический выигрыш в пределах от 0,5 до 1 дБ.

Исследована возможность мягкого декодирования блочного кода в соответствии с принципом максимального правдоподобия при использовании синдромной решетки. По итогам проведенных исследований зафиксировано снижение вероятности битовой ошибки в декодированном информационном сообщении при декодировании мягкого решения детектора – в 70 раз, при  $p_B = 1 \cdot 10^{-4}$ .

При применении описанного варианта декодирования мягкого решения стала возможной перфорация кода Хемминга. Корректировка кодовой скорости реализуется в пределах от 0,57 до 0,8 (на 40 %), что значительно превышает

возможности по корректировке скорости при декодировании жесткого решения детектора.

Предложен модифицированный метод последовательного кодирования, позволяющий изменять параметры последовательного каскадного кода, путем осуществления кодирования внутренним кодом только определенной части бит с выхода кодера внешнего кода. В качестве внешнего кода использован блочный код. Плавное изменение кодовой скорости возможно на 48 %.

Описана модификация метода декодирования по максимуму правдоподобия при применении к значениям символов коэффициентов, пропорциональных их достоверности. В результате чего наблюдался рост корректирующей способности ГПКК. При использовании кода Голя в качестве внутреннего кода наблюдается снижение вероятности битовой ошибки в 2,5 раза с  $4,7 \cdot 10^{-7}$  до  $1,9 \cdot 10^{-7}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ ).

Главы 2 и 3 были посвящены вопросам повышения эффективности последовательных каскадных кодов, состоящих из соединения разнообразных блочных и сверточных кодов. Развитием концепции каскадного кодирования стали турбокоды, в которых используются итеративные алгоритмы декодирования. В следующей главе будут исследованы вопросы возможного применения итеративных алгоритмов декодирования турбокодов для декодирования блочных и сверточных кодов.

## Глава 4. ДЕКОДИРОВАНИЕ БЛОЧНЫХ И СВЕРТОЧНЫХ КОДОВ С ПРИМЕНЕНИЕМ ДЕКОДЕРА ТУРБОКОДА

Данная глава посвящена вопросам повышения эффективности применения систематических сверточных и систематических блочных кодов при декодировании в соответствии с критерием максимума апостериорной вероятности. В начале главы рассматриваются принципы построения турбокодов, вводятся термины и обозначения, принятые при рассмотрении процессов кодирования и декодирования турбокодов. Исследуется возможность повышения эффективности турбокодов путем применения перфорации. Далее рассматривается построение комбинированных кодов путем реализации декодирования блочных и сверточных кодов с применением декодера турбокода и возможности повышения корректирующей способности предложенных комбинированных кодов.

### **4.1. Вычисление надежности информационных символов в перфорированных сверточных кодах в соответствии с алгоритмом SOVA**

Турбокод является развитием системы каскадного кодирования путем применения итеративного декодирования. Рассмотрим принципы построения турбокодов, изложенные в многочисленных источниках [9, 167, 168, 169, 170]. Основных принципов всего два:

- построение кодов с кодовыми словами, обладающими квазислучайными свойствами;

- построение декодеров, реализующих алгоритмы итеративного декодирования, в ходе которого осуществляется передача мягкого решения с выхода одного декодера на вход другого с повторением данной процедуры до тех пор, пока не будет достигнута необходимая точность принятия решения.

При высокой вероятности возникновения ошибки наиболее эффективными среди помехоустойчивых кодов считаются турбокоды. Турбокоды за время их

использования зарекомендовали себя как одни из наиболее востребованных в современных телекоммуникационных системах помехоустойчивых кодов [171, 172, 173, 174, 175]. Они показали свою эффективность в разнообразных стандартах: CDMA-2000, INTELSAT, IEEE 802.16, WiMax, DVB-RCS, UMTS, 3GPP LTE, TIA-1008, DMR.

Одним из важных этапов работы декодера турбокода является вычисление надежности информационных символов. Существует модифицированный алгоритм Витерби с мягким выходом, вычисляющий надежности информационных символов (далее – SOVA, от английского «soft-output Viterbi algorithm») [12, 176, 177, 178]. Надежность информационного символа  $u_i$  вычисляется как логарифм отношений правдоподобия:

$$\Lambda(u_i) = \log \frac{Pr(u_i=1|r)}{Pr(u_i=0|r)}, \quad (4.1)$$

где  $r$  – принятая последовательность. Путем несложных преобразований [12], формулу (4.1) можно привести к следующему виду:

$$\Lambda(u_i) = M_i(0) - M_i(1). \quad (4.2)$$

Для  $i$  секции решетки надежность информационного символа  $u_i$  равна разности между минимальной метрикой путей  $M_i(0)$  для случая  $u_i=0$ , и минимальной метрикой путей  $M_i(1)$  для случая  $u_i=1$ . Суть алгоритма SOVA заключается в вычислении надежности информационного символа по формуле (4.1). Знак «+» или «-» значения надежности информационного символа, рассчитанного по формуле (4.2), соответствует принятию жесткого решения о значении информационного символа, а абсолютная величина характеризует степень надежности жесткого решения. При этом знак «+» соответствует принятию жесткого решения о декодировании «1», а «-» о декодировании «0».

Алгоритм SOVA может использоваться при бинарной или многоуровневой модуляции для декодирования блоков данных или непрерывной последовательности (в режиме скользящего окна), самостоятельно или как компонент декодера турбокода. При относительно низкой сложности, эффективность алгоритма SOVA асимптотически приближается к эффективности

оптимальных алгоритмов декодирования по максимуму апостериорной вероятности при снижении вероятности появления ошибочного бита в канале передачи данных.

Рассмотрим вычисление надежности информационных символов для случая простейшего несистематического сверточного кода, кодер которого изображен на рисунке 2.1. Для других видов сверточных кодов процесс вычисления надежности информационных символов будет аналогичен рассмотренному ниже.

На рисунке 4.1 слева представлен фрагмент решетки, соответствующей одной итерации декодирования. Из каждого узла решетки линиями показаны две разрешенные ветви, соответствующие информационным символам «1» или «0». Выходящие из узла верхние ветви соответствуют информационному символу «0» и показаны на рисунке 4.1 в центре. Нижние ветви соответствуют информационному символу «1» и показаны на рисунке 4.1 справа.

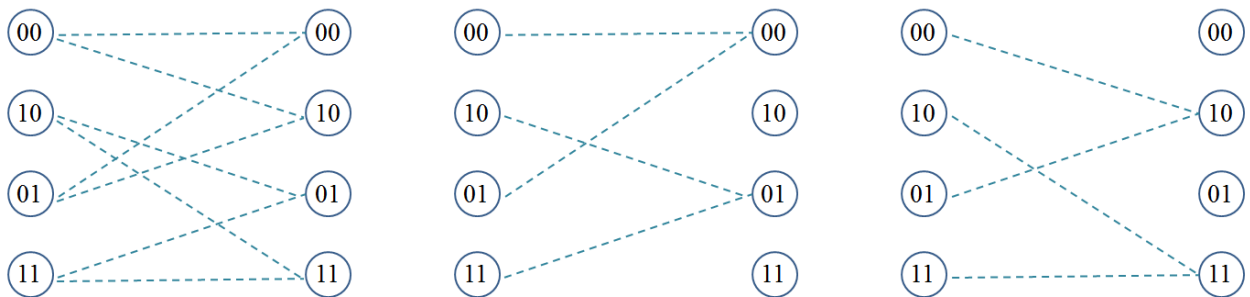


Рисунок 4.1. Разрешенные переходы между узлами решетки (ветви)

Все ветви являются частью разнообразных разрешенных путей. Для вычисления надежности информационного символа, соответствующего данной итерации, необходимо найти минимальную метрику для путей, содержащих ветви соответствующие информационному символу «0» (то есть  $M_i(0)$ ), и найти минимальную метрику для путей, содержащих ветви соответствующие информационному символу «1» (то есть  $M_i(1)$ ), и подставить найденные значения в формулу (4.2). Работа алгоритма SOVA для случая выдачи детектором мягкого решения отличается только правилом вычисления метрик.

Пусть кодером закодирована информационная последовательность 110100, в результате чего получена кодовая последовательность 11 01 01 00 10 11.

Рассмотрим вычисление надежности информационных символов для случая выдачи детектором жесткого решения. Пусть детектирована последовательность 11 01 00 00 10 11, то есть имеется один ошибочный бит (подчеркнут). Прямой проход декодирования с вычислением Хемминговых метрик разрешенных путей схематично показан рисунке 4.2. Обозначения на рисунке 4.2 введены в п. 2.1.

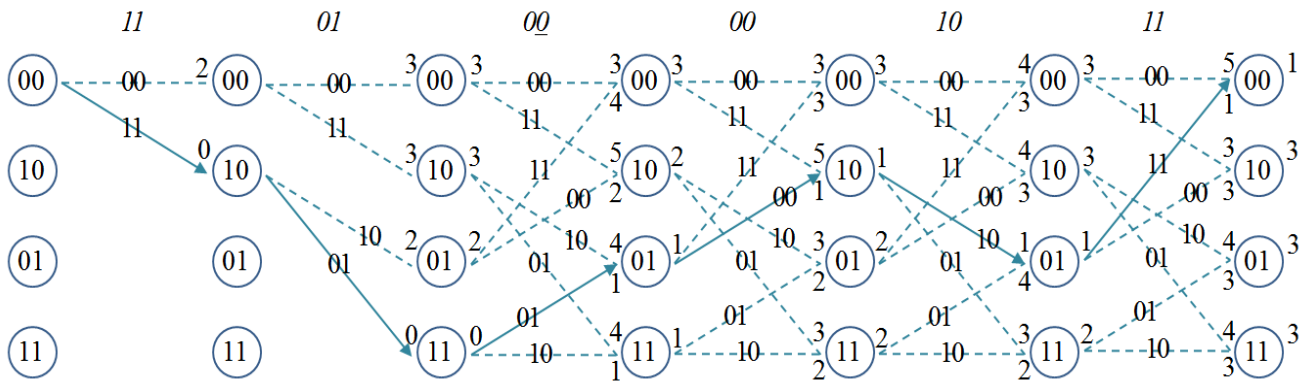


Рисунок 4.2. Прямой проход

При декодировании конечной последовательности, закодированной сверточным кодом, наименее защищенными от ошибок являются последние биты последовательности. Для повышения защищенности последних бит к информационной последовательности добавляются терминирующие биты [10, 12, 179], обнуляющие значения ячеек памяти кодера (для рассматриваемой информационной последовательности это два последних бита). Известное конечное состояние ячеек памяти упрощает процесс декодирования, позволяя исключить из рассмотрения ряд невозможных путей в конце решетки. Рисунок 4.3 повторяет рисунок 4.2, при этом в нем в конце решетки исключены невозможные пути, а на каждой ветви указано значение ее метрики (вместо значений генерируемых бит). Обратный проход осуществляется аналогично прямому проходу и схематично показан на рисунке 4.4. Обратный проход начинается с правой части решетки, а значит числа справа от узла решетки – значения метрик путей, входящих в узел, число слева от узла решетки – значение метрики пути, выходящего из узла.

На рисунке 4.5 схематично показаны результаты обоих проходов. Значения метрик прямого и обратного путей указаны над узлами в формате «метрика прямого пути/метрика обратного пути».

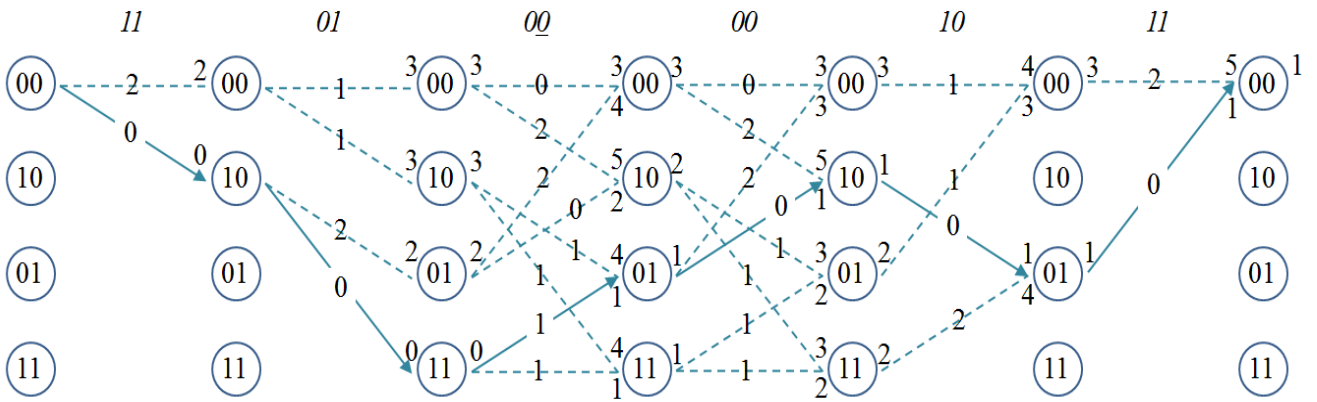


Рисунок 4.3. Прямой проход. Метрики ветвей

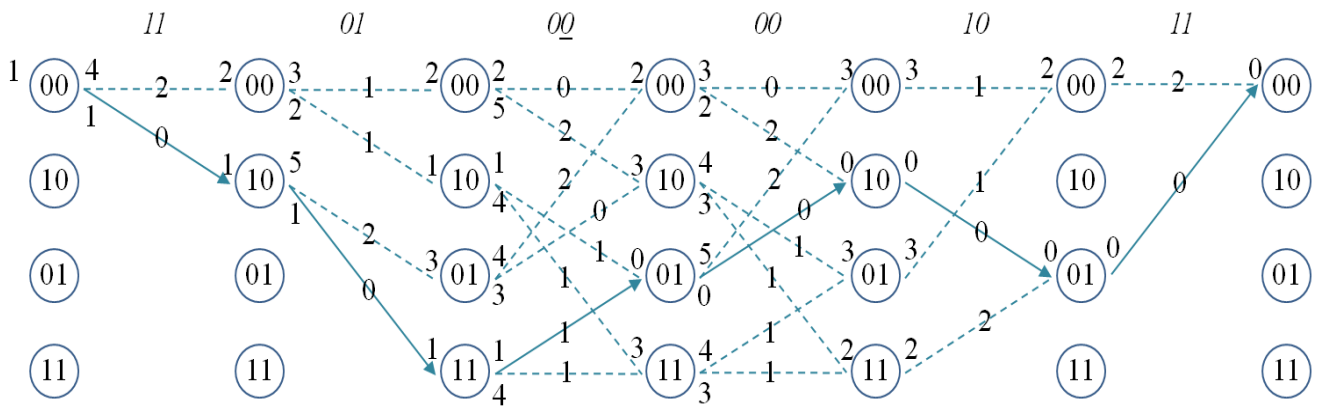


Рисунок 4.4. Обратный проход

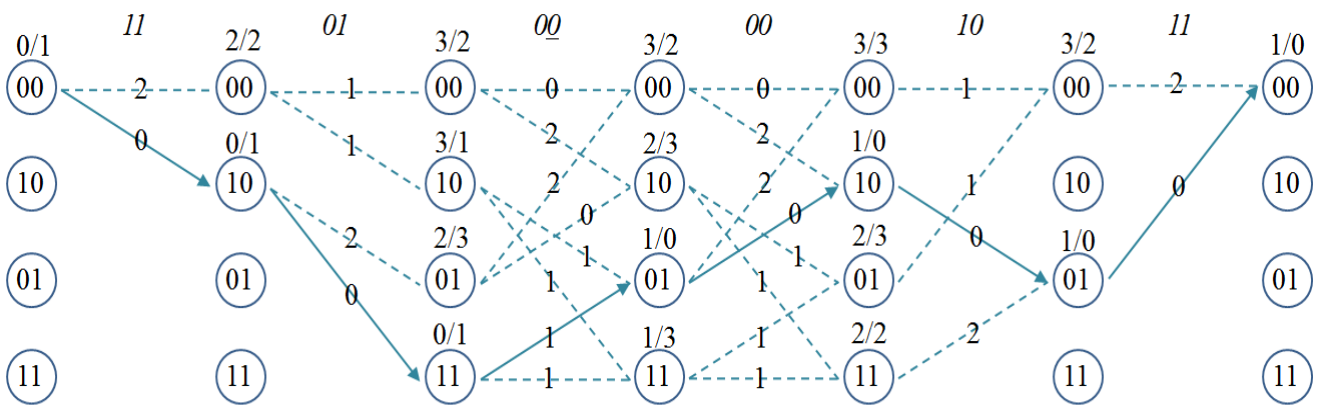


Рисунок 4.5. Прямой и обратный проходы

В ходе прямого прохода определяется финальный выживший путь, имеющий наименьшую метрику. Если для данной итерации финальный выживший путь



соответствует информационному биту «0», то значение  $M_i(0)$  равно метрике финального выжившего пути и остается найти только  $M_i(1)$ . Аналогично, если для данной итерации финальный выживший путь соответствует информационному биту «1», то значение  $M_i(1)$  равно метрике финального выжившего пути и остается найти только  $M_i(0)$ .

Подробно рассмотрим расчет надежности третьего информационного символа. На третьей итерации декодирования финальный выживший путь проходит от узла «11» к узлу «01» и соответствует значению информационного символа «0» (см. рис. 4.6 слева). Метрика финального выжившего пути равна единице, тогда  $M_3(0) = 1$ . Рассмотрим нахождение минимальной метрики пути, соответствующему информационному символу «1». Такие пути содержат ветви, показанные на рисунке 4.6 в центре. Метрика пути, содержащего ту или иную ветвь, равна сумме трех слагаемых: метрике прямого пути от начала решетки до данной ветви ( $MF$ ), метрике самой ветви и метрике обратного пути от конца решетки до данной ветви ( $MB$ ). На рисунке 4.6 (справа) приведен расчет минимальной метрики пути, включающего ветвь от узла «01» к узлу «10». Аналогично рассмотренному можно рассчитать метрики трех других путей, соответствующих информационному символу «1». Данные для расчетов и результаты расчетов метрик путей приведены в таблице 4.1. Определив из таблицы минимальное значение метрики пути, получим  $M_3(1) = 4$ .

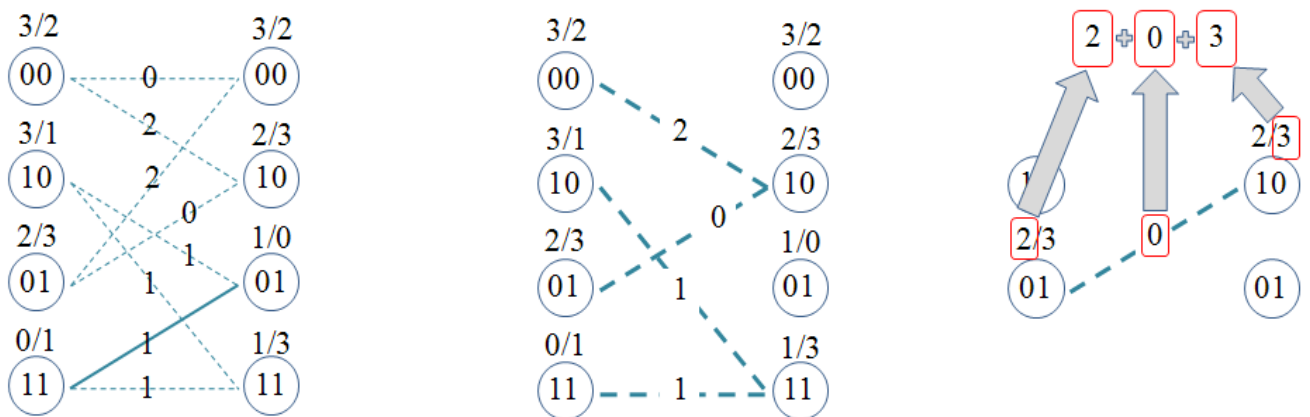


Рисунок 4.6. Расчет надежности информационного символа

Расчет метрик путей

Ветви, входящие в путь	$MF$	Метрика ветви	$MB$	Метрика пути
«00» – «10»	3	2	3	8
«10» – «11»	3	1	3	7
«01» – «10»	2	0	3	5
«11» – «11»	0	1	3	4

Подставив значения  $M_3(0)$  и  $M_3(1)$  в формулу (4.2), рассчитаем надежность третьего декодированного информационного символа:

$$A(u_3) = M_3(0) - M_3(1) = 1 - 4 = -3. \quad (4.3)$$

Рассчитанные аналогично значения надежности остальных символов приведены в таблице 4.2. Расчет надежности двух последних символов (терминирующих символов) упрощается, так как отсутствуют ветви, соответствующие информационному символу «1». Терминирующие символы в данном случае обладают абсолютной надежностью. Следует также отметить, что в рассмотренном примере с жесткими решениями детектора, были получены мягкие решения относительно значений информационных бит.

Актуальной задачей является адаптация корректирующей способности помехоустойчивого кода к изменениям характеристик канала передачи данных. Это может быть реализовано, например, путем перфорации кода.

Таблица 4.2

Результаты расчета значений символов

$i$	1	2	3	4	5	6
$A(u_i)$	3	4	-3	4	$-\infty$	$-\infty$
Решение о значении информационного бита	1	1	0	1	0	0

Рассмотрим вычисление надежности информационных символов для случая перфорированной кодовой последовательности. Пусть в рассмотренной ранее кодовой последовательности выколот каждый четвертый бит, а детектированная последовательность соответствует ранее рассмотренной. Тогда, с учетом

перфорации, на вход декодера поступает последовательность 11 0X00 0X 10 1X, где символом  $X$  обозначены перфорированные биты, не принимающие участие в расчете метрики. На рисунке 4.7 схематично показан прямой проход декодирования перфорированной последовательности. Условные обозначения аналогичны использованным для схемы на рисунке 4.3. Финальный выживший путь соответствует найденному ранее для принятой последовательности без перфорации. На рисунке 4.8 показаны результаты обоих проходов (условные обозначения аналогичны использованным в схеме на рисунке 4.5). С использованием полученных в результате прямого и обратного прохода метрик можно произвести расчет надежности информационных символов. Результаты расчета приведены в таблице 4.3.

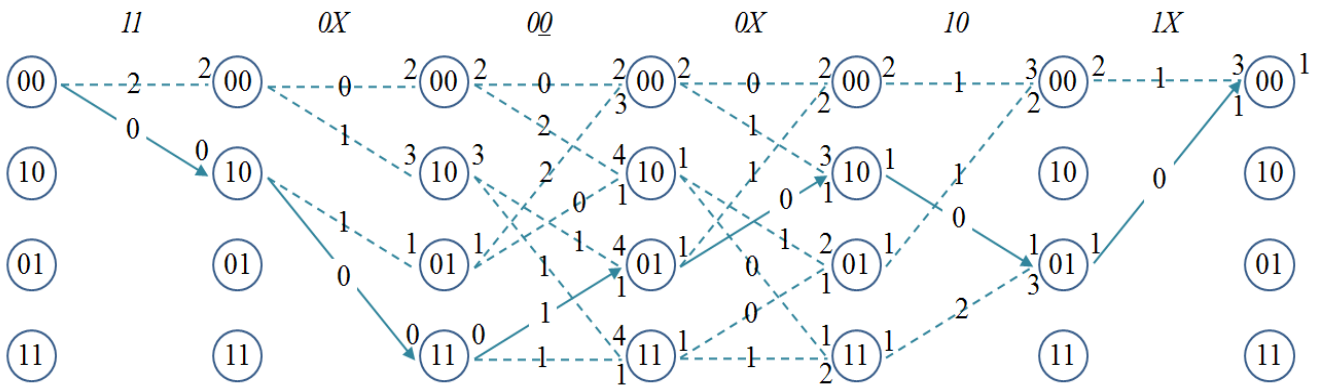


Рисунок 4.7. Прямой проход декодирования перфорированной последовательности

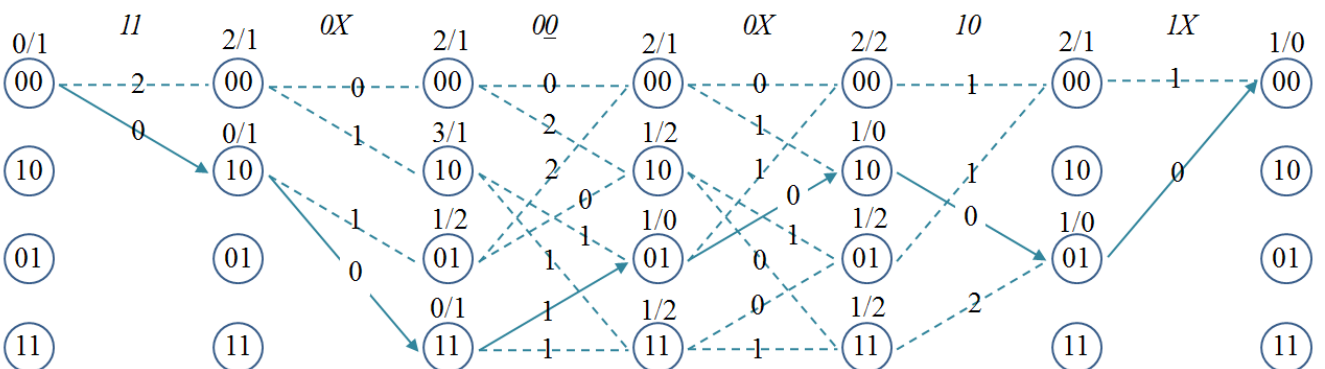


Рисунок 4.8. Прямой и обратный проходы декодирования перфорированной последовательности

Результаты расчета значений символов

$i$	1	2	3	4	5	6
$L(u_i)$	2	2	-2	2	$-\infty$	$-\infty$
Решение о значении информационного бита	1	1	0	1	0	0

Жесткие значения информационных бит, полученные для кодовых последовательностей с перфорацией и без перфорации, одинаковы, а надежность значений для случая перфорированной последовательности ниже, что согласуется со снижением корректирующей способности возникшей из-за перфорации кода.

Вычисление надежности информационных символов является необходимым этапом работы декодера турбокода. Надежности информационных символов могут быть получены при использовании модифицированного алгоритма Витерби с мягким выходом, при этом алгоритм может работать как с жесткими, так и с мягкими решениями детектора. Алгоритм без существенных изменений работает с перфорированными кодовыми последовательностями, при этом наблюдается ожидаемое падение надежности жестких решений.

В параграфе рассмотрены принципы построения турбокодов и процесс декодирования в соответствии с алгоритмом SOVA. Показан процесс декодирования перфорированного турбокода по синдромной решетке. Наиболее распространенным алгоритмом декодирования турбокодов является алгоритм, реализующий декодирование согласно с принципом максимума апостериорной вероятности информационного бита (далее – MAP, от английского «maximum a posteriori») и его модификации. Алгоритм MAP является развитием алгоритма SOVA. Далее рассмотрим процесс декодирования турбокода с применением алгоритма MAP. Произведем оценку корректирующей способности турбокода, в том числе, при наличии перфорации.

## **4.2. Оценка корректирующей способности перфорированного турбокода при применении алгоритма декодирования MAP**

### **4.2.1. Оценка целесообразности введения перфорации в разные части кодового слова турбокода**

В п. 4.1 рассмотрена возможность построения турбокодов на основе перфорированных сверточных кодов. Как видно из структурной схемы  $N$  – размерного кодера турбокода (см. рис. 4.9), поток передаваемых данных состоит из последовательности информационных бит и  $N$  последовательностей проверочных бит [10, 11, 12]. В данном параграфе будет произведено исследование целесообразности перфорации той или иной части кодового слова по итогам рассмотрения результатов декодирования кодовой последовательности, сформированной кодером турбокода, состоящим из двух кодеров систематического сверточного кода (далее – ССК) и устройства перемежения (см. рис. 4.10). Декодирование будет произведено в соответствии с принципом максимума апостериорной вероятности информационного бита. Процесс декодирования в соответствии с алгоритмом MAP подробно изложен в разнообразных источниках [180, 181, 182] и является развитием рассмотренного в п. 4.1 алгоритма SOVA. Реализация процедуры декодирования турбокода основана на вычислении надежности информационных символов. В параграфе будет рассмотрено вычисление надежности информационных символов на выходе декодера перфорированного турбокода после первой полной итерации декодирования для трех вариантов перфорации: последовательности информационных бит, последовательности проверочных бит первого кода, последовательности проверочных бит второго кода [183].

Коротко опишем процесс создание кодовой последовательности турбокода. За один такт кодирования на выходе турбокода формируются три кодовых бита. Первый кодовый бит – это информационный бит, второй кодовый бит – проверочный бит с выхода первого кодера, третий кодовый бит – проверочный бит

с выхода второго кодера (на вход второго кодера подаются информационные биты после перемежения).

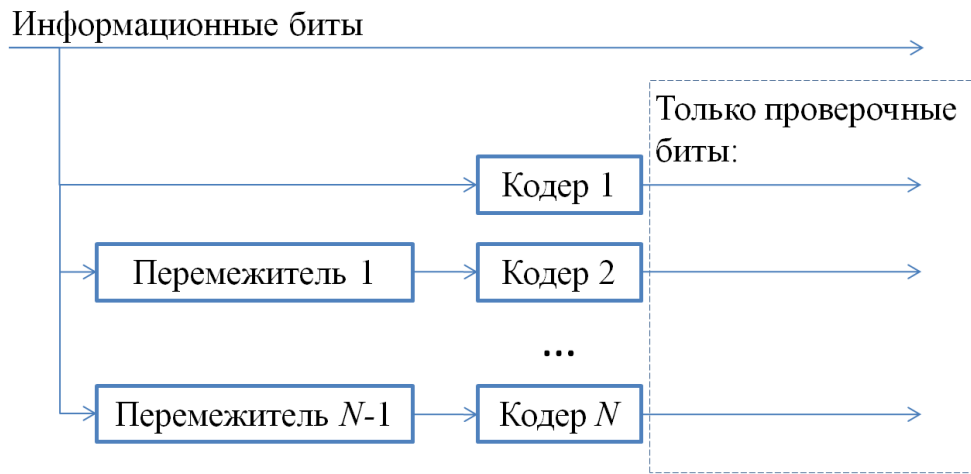


Рисунок 4.9. Структурная схема  $N$  – размерного кодера турбокода

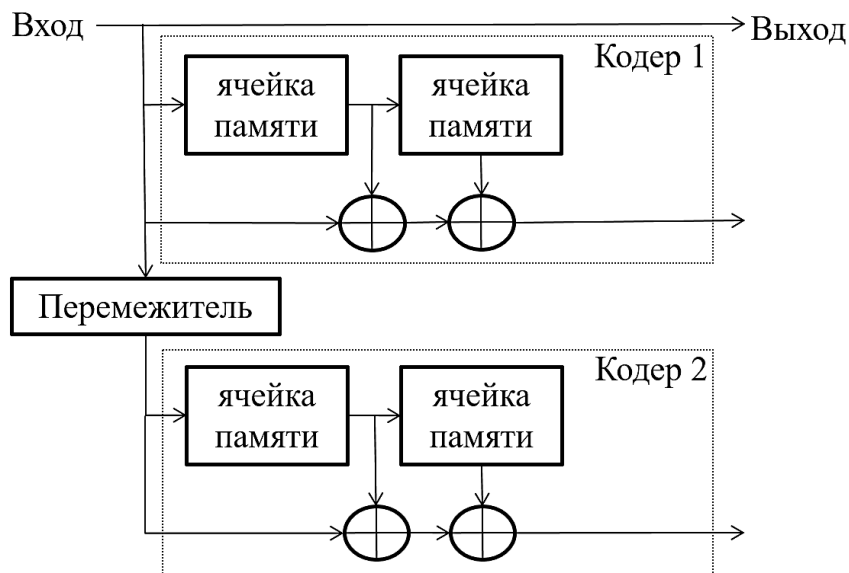


Рисунок 4.10. Структурная схема кодера турбокода на основе сверточного кодера

Пусть 1100 информационная последовательность. Процесс кодирования первым кодером схематично показан на рисунке 4.11. На рисунке состояние узлов решетки 00, 10, 01, 11 для удобства обозначены буквами  $a, b, c, d$  соответственно. Ветви, формирующие кодовую последовательность, показаны сплошными линиями. В результате формируется последовательность проверочных бит первого кодера: 1001. Известно, что процесс декодирования сверточного кода облегчается, если известно финальное состояние ячеек памяти, как правило, соответствующее их обнулению (состояние  $a$ ). Для этого в конце информационных бит добавляют,

так называемую, «терминирующую» последовательность. Будем считать, что при декодировании последовательности, сформированной первым кодером, финальное состояние известно.

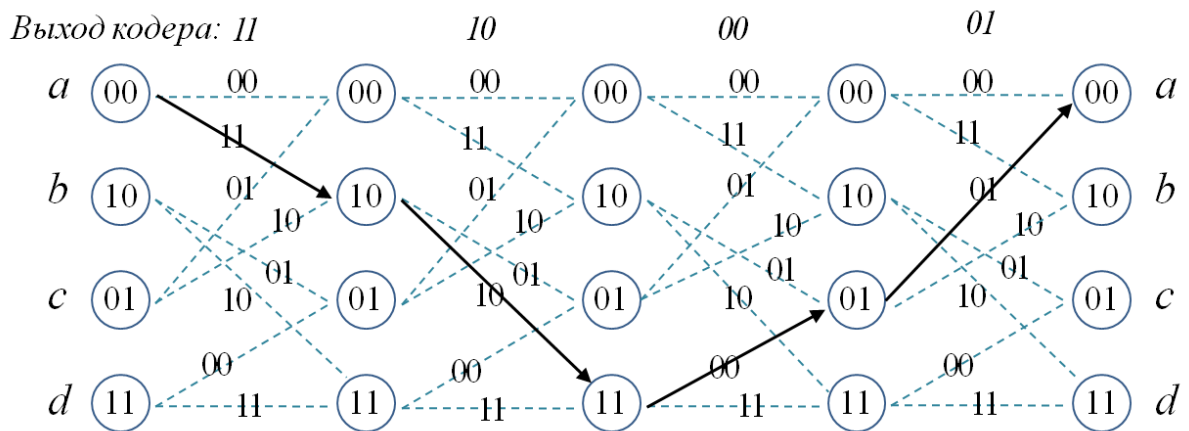


Рисунок 4.11. Процесс кодирования первым кодером

Имея информационную последовательность 1100 и последовательность проверочных бит первого кодера 1001 получаем на 2/3 сформированную последовательность выхода кодера турбокода: 11 10 00 01. Для окончательного формирования выходной последовательности кодера турбокода необходимо к каждой паре бит добавить проверочные биты с выхода второго кодера.

На вход второго кодера поступает информационная последовательность после перемежения. Пусть при перемежении первый по счету бит меняется с третьим, а второй с четвертым, тогда на вход кодера поступает последовательность 0011. Процесс кодирования вторым кодером схематично показан на рисунке 4.12. В результате формируется последовательность проверочных бит второго кодера 0010.

Заполнив каждую третью позицию выходной последовательности кодера турбокода проверочными битами второго кодера, имеем окончательно сформированную последовательность выхода кодера турбокода: 110 100 001 010. Пусть передача бит происходит биполярными сигналами, тогда передаваемая последовательность будет иметь вид: 11-1 1-1-1 -1-11 -11-1.

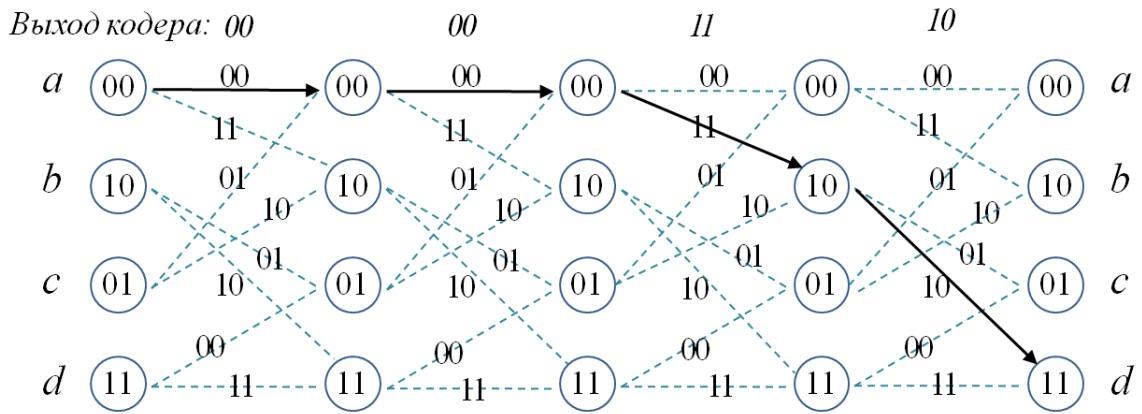


Рисунок 4.12. Процесс кодирования вторым кодером

Алгоритм декодирования MAP наиболее эффективен при использовании мягкого выхода детектора. Очевидно, что сравнение результатов декодирования при перфорации информационной или одной из последовательности проверочных бит будут отличаться при выкалывании символа, имеющего значение 0,1 или 0,9. Для исключения такого рода искажения результатов исследования будем использовать жесткое решение детектора о значении принятого символа («1» или «-1»), а выколотым битам присваивать на входе декодера значение «0». Структурная схема декодера турбокода изображена на рисунке 4.13. В каждый момент времени  $k$  (кроме  $k = 5$ ) на вход декодера турбокода поступают детектированные значения информационного бита  $x_k$  и проверочных бит первого и второго кодера  $y_{k1}$  и  $y_{k2}$ , соответственно.

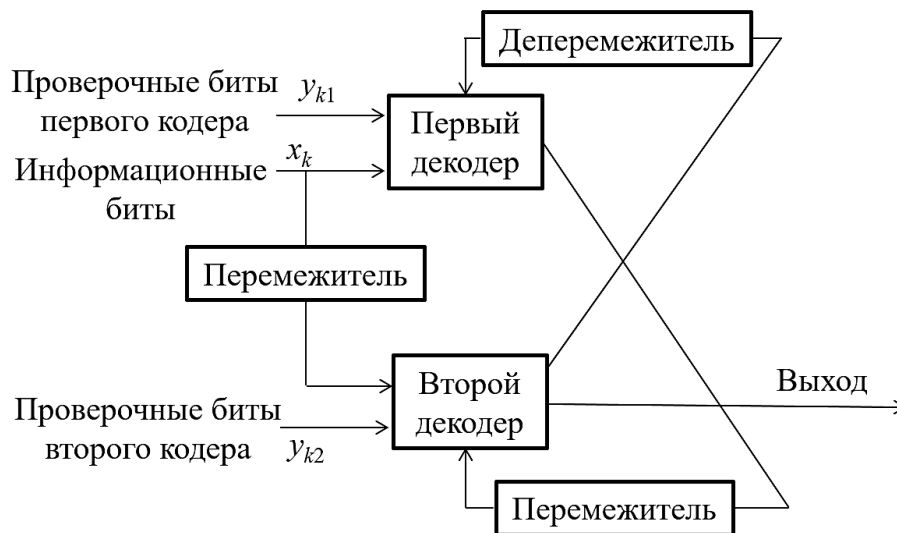


Рисунок 4.13. Структурная схема декодера турбокода



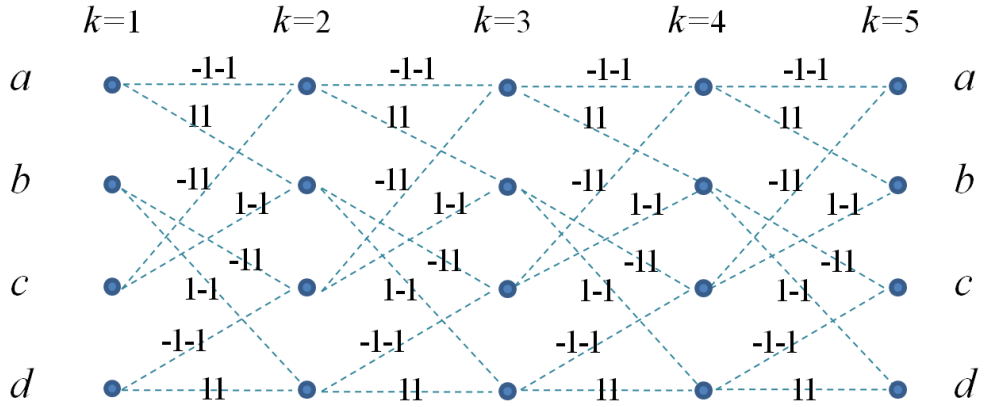


Рисунок 4.14. Ветви декодера

Коротко рассмотрим алгоритм декодирования MAP, используя формулы (4.4) – (4.9) из источника [12], который начинается с работы первого декодера. На вход первого декодера в каждый момент времени  $k$  (кроме  $k = 5$ ) поступает детектированные значения информационного бита  $x_k$  и проверочного бита первого кодера  $y_{k1}$ . Декодирование начинается с расчета метрик ветвей решетки декодера (см. рис. 4.14). Формула для вычисления метрик ветвей имеет следующий вид:

$$\delta_k^{i,m} = 0,5 \cdot \exp(x_k \cdot u_k^i + y_{k1} \cdot v_k^{i,m}), \quad (4.4)$$

где параметр  $i$  принимает значение 0 или 1, в зависимости от того, приходу какого бита соответствует данная ветвь (для ветвей, идущих вверх от узла  $i = 0$ , для ветвей идущих вниз от узла  $i = 1$ ),  $u_k$  и  $v_k$  – значения бит, генерируемых данной ветвью,  $m$  – параметр, обозначающий состояние узла решетки (принимающий значения:  $a, b, c, d$ ).

После расчета значений метрик ветвей для всех  $m$  и  $k$  (кроме  $k = 5$ ) можно приступить к следующему этапу декодирования, в ходе которого будет проведен расчет метрик прямого прохода по решетке декодера. Формула для вычисления метрик прямого прохода имеет следующий вид:

$$\alpha_{k+1}^m = \sum_{j=0}^1 \delta_k^{j,b(j,m)} \cdot \alpha_k^{b(j,m)}. \quad (4.5)$$

Начальное состояние пути по решетке (при  $k = 1$ ) нам известно:  $m = a$ . Тогда при  $k = 1$  и  $m = a$  метрика прямого прохода равна 1, а все остальные метрики прямого прохода (для  $m = b, c, d$ ) при  $k = 1$  равны нулю. Далее вычисляем все

метрики прямого прохода по формуле (4.5) для  $k = 2, 3, 4$ . Для последнего состояния  $k = 5$  метрика прямого прохода не вычисляется.

После расчета метрик прямого прохода необходимо произвести расчет метрик обратного прохода по решетке декодера из состояния  $k = 5$  в состояние  $k = 1$ . Формула для вычисления метрик обратного прохода имеет следующий вид:

$$\beta_k^m = \sum_{j=0}^1 \delta_k^{j,m} \cdot \beta_{k+1}^{f(j,m)}. \quad (4.6)$$

Как говорилось ранее, из-за применения терминирующей последовательности кодирование первым кодером завершается в известном состоянии  $m = a$ . Тогда при  $k = 5$  и  $m = a$  метрика обратного прохода равна 1, а все остальные метрики обратного прохода (для  $m = b, c, d$ ) при  $k = 5$  равны нулю. Далее вычисляем все метрики прямого прохода по формуле (4.6) для  $k = 4, 3, 2$ . Для состояния  $k = 1$  метрика обратного прохода не вычисляется.

Имея значения всех трех видов метрик, можно провести расчет логарифмической функции правдоподобия для всех информационных бит по следующей формуле:

$$L_e(d_k) = \ln \left( \frac{\sum_m \alpha_k^m \delta_k^{1,m} \beta_{k+1}^{f(1,m)}}{\sum_m \alpha_k^m \delta_k^{0,m} \beta_{k+1}^{f(0,m)}} \right). \quad (4.7)$$

Далее, аналогичные вычисления по формулам (4.4) – (4.7) проводятся для второго декодера с учетом ниже приведенных особенностей. На вход второго декодера в каждый момент времени  $k$  (кроме  $k = 5$ ) поступают детектированные и перемеженные значения информационного бита и детектированные значения проверочного бита второго кодера  $u_{k2}$ . Для второго декодера при вычислении метрик обратного прохода необходимо иметь в виду, что конечное состояние на решетке неизвестно, поэтому при  $k = 5$  все метрики обратного прохода равны единице.

После расчета логарифмической функции правдоподобия для всех информационных бит на выходе второго декодера и перемежения результатов расчета, первая полная итерация декодирования считается завершенной. Становится возможным вычисление значения мягкого выхода декодера турбокода:

$$L(d_k) = L_c(d_k) + L'_e(d_k) + L_e(d_k), \quad (4.8)$$

где  $L_c(d_k)$  – канальное измерение значения символа,  $L'_e(d_k)$  – априорное значение логарифмической функции правдоподобия (в нашем случае по результатам прохождения первого декодера),  $L_e(d_k)$  – значение логарифмической функции правдоподобия по результатам декодирования (в нашем случае по результатам прохождения второго декодера).

Для достижения необходимой надежности решения может быть произведено несколько итераций, причем каждая последующая должна быть проведена с учетом логарифмической функции правдоподобия для всех информационных бит, полученной на предыдущей итерации. Для сравнения результатов декодирования при различных вариантах введения перфорации ограничимся результатами одной полной итерации. В конечном счете, после проведения необходимого количества итераций, принимается жесткое решение о значении в соответствии со следующим критерием:

$$\begin{cases} d_k = 1, \text{ если } L(d_k) > 0 \\ d_k = 0, \text{ если } L(d_k) < 0 \end{cases} \quad (4.9)$$

По итогам рассмотрения процесса декодирования можно сделать предположение относительно наибольшей целесообразности перфорации той или иной последовательности, исходя из следующих рассуждений. Информационные биты участвуют в процессе декодирования в обоих декодерах, и их перфорация отрицательно повлияет на достоверность результатов декодирования обоих декодеров. Как говорилось ранее, из-за применения терминирующей последовательности кодирование первым кодером завершается в известном состоянии, в связи с чем декодирование данной последовательности дает более достоверные результаты, чем последовательности, сформированной вторым кодером. Отсюда следует предположение о наибольшей целесообразности перфорации последовательности, сформированной из проверочных бит второго кодера (как наименее ценной).

Рассмотрим декодирование детектированной без ошибок последовательности 11-1 1-1-1 -1-11 -11-1 для трех вариантов введения перфорации, определенных ранее.

При перфорации всех информационных бит последовательность на входе декодера турбокода имеет вид 01-1 0-1-1 0-11 01-1. Тогда на вход первого декодера поступает последовательность 01 0-1 0-1 01, а на вход второго декодера 0-1 0-1 01 0-1.

Произведем расчет выхода первого декодера. Результаты расчета метрик ветвей по формуле (4.4) приведены в таблице 4.4. Результаты расчета метрик прямого прохода по формуле (4.5) приведены в таблице 4.5. Результаты расчета метрик обратного прохода по формуле (4.6) приведены в таблице 4.6.

По итогам работы первого декодера по формуле (4.7) получены следующие результаты расчета логарифмической функции правдоподобия:  $L_e(d_1) = 3,875$ ;  $L_e(d_2) = 3,875$ ;  $L_e(d_3) = -\infty$ ;  $L_e(d_4) = -\infty$ . Выход первого декодера сформирован.

Таблица 4.4

Значения метрик ветвей

$k$	1		2		3		4	
$i$	0	1	0	1	0	1	0	1
$m=a$	0,184	1,359	1,359	0,184	1,359	0,184	0,184	1,359
$m=b$	1,359	0,184	0,184	1,359	0,184	1,359	1,359	0,184
$m=c$	1,359	0,184	0,184	1,359	0,184	1,359	1,359	0,184
$m=d$	0,184	1,359	1,359	0,184	1,359	0,184	0,184	1,359

Таблица 4.5

Значения метрик прямого прохода

$k$	1	2	3	4
$m=a$	1	0,184	0,25	0,386
$m=b$	0	1,359	0,034	0,386
$m=c$	0	0	0,25	2,517
$m=d$	0	0	1,847	0,386

Значения метрик обратного прохода

$k$	2	3	4	5
$m = a$	0,386	0,25	0,184	1
$m = b$	2,517	0,25	0	0
$m = c$	0,386	0,034	1,359	0
$m = d$	0,386	1,847	0	0

Выход второго декодера рассчитывается аналогично с учетом выше сказанных замечаний. По итогам работы второго декодера получены следующие результаты расчета логарифмической функции правдоподобия:  $L(d_1) = 1,325$ ;  $L(d_2) = 0$ ;  $L(d_3) = 0$ ;  $L(d_4) = 0$ .

Результаты расчета по формуле (4.8) мягкого выхода декодера турбокода по итогам одной полной итерации приведены в таблице 4.7. Жесткий выход турбокода в соответствии с критерием (4.9) не содержит ошибок.

Таблица 4.7

Результаты расчета мягкого выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	0	3,876	1,325	5,201	1
2	0	3,876	0	3,876	1
3	0	$-\infty$	0	$-\infty$	0
4	0	$-\infty$	0	$-\infty$	0

Таблица 4.8

Результаты расчета мягкого выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	1	2	4,897	7,897	1
2	1	2	3,843	6,843	1
3	-1	$-\infty$	-6,794	$-\infty$	0
4	-1	$-\infty$	-5,567	$-\infty$	0

При перфорации проверочных бит первого кодера входная последовательность декодера турбокода выглядит следующим образом: 10–1 10–1 –101 –10–1. Тогда на вход первого декодера поступает последовательность 10 10 –10 –10, а на вход второго декодера –1–1 –1–1 11 1–1. Результаты расчета мягкого выхода декодера турбокода приведены в таблице 4.8. Жесткий выход турбокода в соответствии с критерием (4.9) не содержит ошибок.

При перфорации проверочных бит второго кодера входная последовательность декодера турбокода выглядит следующим образом: 110 1–10 –1–10 –110. Тогда на вход первого декодера поступает последовательность 11 1–1 –1–1 –11, а на вход второго декодера –10 –10 10 10. Результаты расчета мягкого выхода декодера турбокода приведены в таблице 4.9. Жесткий выход турбокода в соответствии с критерием (4.9) не содержит ошибок.

Таблица 4.9

Результаты расчета мягкого выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	1	7,307	2	10,307	1
2	1	7,307	2	10,307	1
3	–1	–∞	–2	–∞	0
4	–1	–∞	–2	–∞	0

Сравнительные результаты расчета мягкого выхода декодера турбокода для разных видов перфорации приведены в таблице 4.10. Наибольшей достоверностью обладают результаты декодирования последовательности с перфорацией проверочных бит второго кодера. Наименьшей достоверностью обладают результаты декодирования последовательности с перфорацией информационных бит.

Рассмотрим результаты декодирования принятой перфорированной последовательности при наличии в ней ошибочных бит для обозначенных ранее трех вариантов введения перфорации. Пусть каждая из последовательностей

(составляющих выход кодера турбокода), не подвергшихся перфорации содержит по одному ошибочному биту.

При перфорации всех информационных бит и наличии ошибок в каждой из последовательностей проверочных бит, последовательность на входе декодера турбокода имеет вид  $01\underline{1} 0-1-1 0-11 0\underline{1}-1$ , где ошибочные биты подчеркнуты. Тогда на вход первого декодера поступает последовательность  $01 0-1 0-1 0\underline{1}$ , а на вход второго декодера  $0\underline{1} 0-1 01 0-1$ .

При перфорации проверочных бит первого кодера и наличии ошибок в информационной последовательности и последовательности проверочных бит второго кодера, последовательность на входе декодера турбокода имеет вид  $10\underline{1} 10-1 \underline{1}01 -10-1$ , где ошибочные биты подчеркнуты. Тогда на вход первого декодера поступает последовательность  $10 10 \underline{1}0 -10$ , а на вход второго декодера  $\underline{1}1 -1-1 11 1-1$ .

Таблица 4.10

Значения выхода декодера турбокода при всех вариантах перфорации

№ бита	Виды перфорации			значение переданного бита
	перфорация информационных бит	перфорация проверочных бит первого кодера	перфорация проверочных бит второго кодера	
1	5,201	7,897	10,307	1
2	3,876	6,843	10,307	1
3	$-\infty$	$-\infty$	$-\infty$	0
4	$-\infty$	$-\infty$	$-\infty$	0

При перфорации проверочных бит второго кодера и наличии ошибок в информационной последовательности и последовательности проверочных бит первого кодера, последовательность на входе декодера турбокода имеет вид  $110 1-10 \underline{1}-10 -1\underline{1}0$ , где ошибочные биты подчеркнуты. Тогда на вход первого декодера поступает последовательность  $11 1-1 \underline{1}-1 -1\underline{1}$ , а на вход второго декодера  $\underline{1}0 -10 10 10$ .

Сравнительные результаты расчета мягкого выхода декодера турбокода для разных видов перфорации при наличии ошибок в неперфорированных последовательностях приведены в таблице 4.11. Как и в случаях отсутствия ошибок, результаты сравнения подтверждают выдвинутое предположение о наибольшей целесообразности введения перфорации в последовательность проверочных бит второго кодера. Также еще раз подтверждена более высокая чувствительность декодера турбокода к достоверности значений информационных бит.

По итогам рассмотрения процесса декодирования можно сделать предположение относительно наибольшей целесообразности перфорации той или иной последовательности, исходя из следующих рассуждений. Информационные биты участвуют в процессе декодирования в обоих декодерах, и их перфорация отрицательно повлияет на достоверность результатов декодирования обоих декодеров. Как говорилось ранее, из-за применения терминирующей последовательности кодирование первым кодером завершается в известном состоянии, в связи с чем декодирование данной последовательности дает более достоверные результаты, чем последовательности, сформированной вторым кодером. Отсюда следует предположение о наибольшей целесообразности перфорации последовательности, сформированной из проверочных бит второго кодера (как наименее ценной).

Таблица 4.11

Значения выхода декодера турбокода при всех вариантах перфорации

№ бита	Виды перфорации			значение переданного бита
	перфорация информационных бит	перфорация проверочных бит первого кодера	перфорация проверочных бит второго кодера	
1	1,449	5,273	7	1
2	-1,073	4,023	6,3	1
3	$-\infty$	$-\infty$	$-\infty$	0
4	$-\infty$	$-\infty$	$-\infty$	0



Рассмотрим результаты декодирования ранее сформированной последовательности 11-1 1-1-1 -1-11 -11-1, детектированной без ошибок для трех вариантов введения перфорации, определенных ранее. Для удобства, кодовая последовательность и выходы детектора для трех вариантов перфорации приведены в таблице 4.12.

При перфорации всех информационных бит последовательность на входе декодера турбокода имеет вид 01-1 0-1-1 0-11 01-1. Тогда на вход первого декодера поступает последовательность 01 0-1 0-1 01, а на вход второго декодера 0-1 0-1 01 0-1. Для удобства, значения входов декодеров для данного случая приведены в таблице 4.13.

Таблица 4.12

Кодовая последовательность и выходы детекторов

Номера бит	0	1	2	3	4	5	6	7	8	9	10	11
	x	y <sub>1</sub>	y <sub>2</sub>	x	y <sub>1</sub>	y <sub>2</sub>	x	y <sub>1</sub>	y <sub>2</sub>	x	y <sub>1</sub>	y <sub>2</sub>
Последовательность на выходе кодера турбокода	1	1	-1	1	-1	-1	-1	-1	1	-1	1	-1
Выход детектора для случая перфорации информационных бит	0	1	-1	0	-1	-1	0	-1	1	0	1	-1
Выход детектора для случая перфорации проверочных бит первого кода	1	0	-1	1	0	-1	-1	0	1	-1	0	-1
Выход детектора для случая перфорации проверочных бит второго кода	1	1	0	1	-1	0	-1	-1	0	-1	1	0
Выход детектора для случая перфорации проверочных бит	1	0	0	1	-1	-1	-1	0	0	-1	1	-1

Входы декодеров при перфорации информационных бит

			$k=$	1	2	3	4
Вход первого декодера	$x_k$	№	0	3	6	9	
		значение	0	0	0	0	
	$y_k$	№	1	4	7	10	
		значение	1	-1	-1	1	
Вход второго декодера	$x_k$	№	6	9	0	3	
		значение	0	0	0	0	
	$y_k$	№	2	5	8	11	
		значение	-1	-1	1	-1	

Произведем расчет выхода первого декодера, начав с расчета метрик ветвей по формуле (4.4).

Для  $k = 1$  метрики ветвей, идущих от состояния  $k = 1$  в  $k = 2$  будут равны:

Для ветвей, идущих вверх (соответствующих приходу 0)  $i = 0$ :

из состояния  $a$ :

$$\delta_1^{0,m=a} = 0,5 \cdot \exp(x_1 \cdot u_1^0 + y_1 \cdot v_1^{0,m=a}) = 0,5 \cdot \exp(0 \cdot (-1) + 1 \cdot (-1)) = 0,184,$$

из состояния  $b$ :

$$\delta_1^{0,m=b} = 0,5 \cdot \exp(x_1 \cdot u_1^0 + y_1 \cdot v_1^{0,m=b}) = 0,5 \cdot \exp(0 \cdot (-1) + 1 \cdot 1) = 1,359,$$

из состояния  $c$ :

$$\delta_1^{0,m=c} = 0,5 \cdot \exp(x_1 \cdot u_1^0 + y_1 \cdot v_1^{0,m=c}) = 0,5 \cdot \exp(0 \cdot (-1) + 1 \cdot 1) = 1,359,$$

из состояния  $d$ :

$$\delta_1^{0,m=d} = 0,5 \cdot \exp(x_1 \cdot u_1^0 + y_1 \cdot v_1^{0,m=d}) = 0,5 \cdot \exp(0 \cdot (-1) + 1 \cdot (-1)) = 0,184,$$

Для ветвей, идущих вниз (соответствующих приходу 1)  $i = 1$ :

из состояния  $a$ :

$$\delta_1^{1,m=a} = 0,5 \cdot \exp(x_1 \cdot u_1^1 + y_1 \cdot v_1^{1,m=a}) = 0,5 \cdot \exp(0 \cdot 1 + 1 \cdot 1) = 1,359,$$

из состояния  $b$ :

$$\delta_1^{1,m=b} = 0,5 \cdot \exp(x_1 \cdot u_1^1 + y_1 \cdot v_1^{1,m=b}) = 0,5 \cdot \exp(0 \cdot 1 + 1 \cdot (-1)) = 0,184,$$

из состояния  $c$ :

$$\delta_1^{1,m=c} = 0,5 \cdot \exp(x_1 \cdot u_1^1 + y_1 \cdot v_1^{1,m=c}) = 0,5 \cdot \exp(0 \cdot 1 + 1 \cdot (-1)) = 0,184,$$

из состояния  $d$ :

$$\delta_1^{1,m=d} = 0,5 \cdot \exp(x_1 \cdot u_1^1 + y_1 \cdot v_1^{1,m=d}) = 0,5 \cdot \exp(0 \cdot 1 + 1 \cdot 1) = 1,359,$$

Аналогично рассчитаем метрики ветвей для  $k = 2, 3, 4$ . Результаты расчета приведены в таблице 4.14.

Рассчитаем метрики прямого прохода. Как говорилось ранее, при  $k = 1$  для состояния  $m = a$  метрика прямого прохода равна 1, а для всех остальных состояний метрика прямого прохода равна 0. Рассчитаем метрики прямого прохода при  $k = 2$  по формуле (4.5).

Метрика прямого прохода для состояния  $m = a$  определяется метриками входящих в это состояние ветвей, перемноженных на значения метрик прямого прохода, для состояний, из которых эти ветви исходят. С учетом того, что в состояние  $m = a$  входит верхняя ветвь ( $i = 0$ ) из состояний  $a$  и верхняя ветвь ( $i = 0$ ) из состояния  $c$ :

$$\alpha_2^{m=a} = \delta_1^{0,m=a} \cdot \alpha_1^{m=a} + \delta_1^{0,m=c} \cdot \alpha_1^{m=c} = 0,184 \cdot 1 + 1,359 \cdot 0 = 0,184.$$

Таблица 4.14

Значения метрик ветвей

$k$	1		2		3		4	
$i$	0	1	0	1	0	1	0	1
$m = a$	0,184	1,359	1,359	0,184	1,359	0,184	0,184	1,359
$m = b$	1,359	0,184	0,184	1,359	0,184	1,359	1,359	0,184
$m = c$	1,359	0,184	0,184	1,359	0,184	1,359	1,359	0,184
$m = d$	0,184	1,359	1,359	0,184	1,359	0,184	0,184	1,359

Метрика прямого прохода для состояния  $m = b$  определяется метриками входящих в это состояние ветвей, перемноженных на значения метрик прямого прохода, для состояний, из которых эти ветви исходят. С учетом того, что в состояние  $m = b$  входит нижняя ветвь ( $i = 1$ ) из состояний  $a$  и нижняя ветвь ( $i = 1$ ) из состояния  $c$ :

$$\alpha_2^{m=b} = \delta_1^{1,m=a} \cdot \alpha_1^{m=a} + \delta_1^{1,m=c} \cdot \alpha_1^{m=c} = 1,359 \cdot 1 + 0,184 \cdot 0 = 1,359.$$

Метрика прямого прохода для состояния  $m = c$  определяется метриками входящих в это состояние ветвей, перемноженных на значения метрик прямого прохода, для состояний, из которых эти ветви исходят. С учетом того, что в состояние  $m = c$  входит верхняя ветвь ( $i = 0$ ) из состояний  $b$  и верхняя ветвь ( $i = 0$ ) из состояния  $d$ :

$$\alpha_2^{m=c} = \delta_1^{0,m=b} \cdot \alpha_1^{m=b} + \delta_1^{0,m=d} \cdot \alpha_1^{m=d} = 1,359 \cdot 0 + 0,184 \cdot 0 = 0.$$

Метрика прямого прохода для состояния  $m = d$  определяется метриками входящих в это состояние ветвей, перемноженных на значения метрик прямого прохода, для состояний, из которых эти ветви исходят. С учетом того, что в состояние  $m = d$  входит нижняя ветвь ( $i = 1$ ) из состояний  $b$  и нижняя ветвь ( $i = 1$ ) из состояния  $d$ :

$$\alpha_2^{m=d} = \delta_1^{1,m=b} \cdot \alpha_1^{m=b} + \delta_1^{1,m=d} \cdot \alpha_1^{m=d} = 0,184 \cdot 0 + 1,359 \cdot 0 = 0.$$

Аналогично вычисляются метрики прямого прохода для  $k = 3, 4$ . Результаты расчета приведены в таблице 4.15.

Таблица 4.15

Значения метрик прямого прохода

$k$	1	2	3	4
$m = a$	1	0,184	0,25	0,386
$m = b$	0	1,359	0,034	0,386
$m = c$	0	0	0,25	2,517
$m = d$	0	0	1,847	0,386

Рассчитаем метрики обратного прохода. Как говорилось ранее, при  $k = 5$  для состояния  $m = a$  метрика обратного прохода равна 1, а для всех остальных состояний метрика обратного прохода равна 0. Рассчитаем метрики обратного прохода при  $k = 4$  по формуле (4.6).

Метрика обратного прохода для состояния  $m = a$  определяется метриками входящих в это состояние при обратном проходе ветвей, перемноженных на значения метрик обратного прохода, для состояний, из которых эти ветви исходят.

С учетом того, что при обратном проходе справа налево в состояние  $m = a$  входит верхняя ветвь ( $i = 0$ ) из состояний  $a$  и верхняя ветвь ( $i = 0$ ) из состояния  $b$ :

$$\beta_4^{m=a} = \delta_4^{0,m=a} \cdot \beta_5^{m=a} + \delta_4^{0,m=b} \cdot \beta_5^{m=b} = 0,184 \cdot 1 + 1,359 \cdot 0 = 0,184.$$

С учетом того, что при обратном проходе справа налево в состояние  $m = b$  входит верхняя ветвь ( $i = 0$ ) из состояний  $c$  и верхняя ветвь ( $i = 0$ ) из состояния  $d$ :

$$\beta_4^{m=b} = \delta_4^{0,m=c} \cdot \beta_5^{m=c} + \delta_4^{0,m=d} \cdot \beta_5^{m=d} = 1,359 \cdot 0 + 0,184 \cdot 0 = 0.$$

С учетом того, что при обратном проходе справа налево в состояние  $m = c$  входит нижняя ветвь ( $i = 1$ ) из состояний  $a$  и нижняя ветвь ( $i = 1$ ) из состояния  $b$ :

$$\beta_4^{m=c} = \delta_4^{1,m=a} \cdot \beta_5^{m=a} + \delta_4^{1,m=b} \cdot \beta_5^{m=b} = 1,359 \cdot 1 + 0,184 \cdot 0 = 1,359.$$

С учетом того, что при обратном проходе справа налево в состояние  $m = d$  входит нижняя ветвь ( $i = 1$ ) из состояний  $c$  и нижняя ветвь ( $i = 1$ ) из состояния  $d$ :

$$\beta_4^{m=d} = \delta_4^{1,m=c} \cdot \beta_5^{m=c} + \delta_4^{1,m=d} \cdot \beta_5^{m=d} = 0,184 \cdot 0 + 1,359 \cdot 0 = 0.$$

Аналогично вычисляются метрики обратного прохода для  $k = 3, 2$ . Результаты расчета приведены в таблице 4.16.

Таблица 4.16

Значения метрик обратного прохода

$k$	2	3	4	5
$m = a$	0,386	0,25	0,184	1
$m = b$	2,517	0,25	0	0
$m = c$	0,386	0,034	1,359	0
$m = d$	0,386	1,847	0	0

Рассчитав значения всех трех видов метрик, можно провести расчет логарифмической функции правдоподобия для всех информационных бит по формуле (4.7). Рассмотрим эту формулу более подробно. В верхней части дроби для всех ветвей, соответствующих приходу единичного информационного бита, просуммировано произведение метрики ветви, метрики прямого прохода для состояния  $m$ , соответствующего началу ветви и метрики обратного прохода для состояния (чаще всего не  $m$ ), соответствующего концу ветви. В нижней части дроби для всех ветвей, соответствующих приходу нулевого информационного бита,

просуммировано произведение метрики ветви, метрики прямого прохода для состояния  $m$ , соответствующего началу ветви и метрики обратного прохода, для состояния (чаще всего не  $m$ ), соответствующего концу ветви. Для рассматриваемого примера формула (4.7) может быть преобразована в вид:

$$L_e(d_k) = \ln \left( \frac{\alpha_k^{m=a} \delta_k^{1,m=a} \beta_{k+1}^{m=b} + \alpha_k^{m=b} \delta_k^{1,m=b} \beta_{k+1}^{m=d} + \alpha_k^{m=c} \delta_k^{1,m=c} \beta_{k+1}^{m=b} + \alpha_k^{m=d} \delta_k^{1,m=d} \beta_{k+1}^{m=d}}{\alpha_k^{m=a} \delta_k^{0,m=a} \beta_{k+1}^{m=a} + \alpha_k^{m=b} \delta_k^{0,m=b} \beta_{k+1}^{m=c} + \alpha_k^{m=c} \delta_k^{0,m=c} \beta_{k+1}^{m=a} + \alpha_k^{m=d} \delta_k^{0,m=d} \beta_{k+1}^{m=c}} \right), \quad (4.10)$$

Тогда для  $k = 1$  согласно формуле (4.10):

$$L_e(d_1) = \ln \left( \frac{1 \cdot 1,359 \cdot 2,517 + 0 \cdot 0,184 \cdot 0,386 + 0 \cdot 0,184 \cdot 2,517 + 0 \cdot 1,359 \cdot 0,386}{1 \cdot 0,184 \cdot 0,386 + 0 \cdot 1,359 \cdot 0,386 + 0 \cdot 1,359 \cdot 0,386 + 0 \cdot 0,184 \cdot 0,368} \right) = 3,875.$$

Для  $k = 2$  согласно формуле (4.10):

$$L_e(d_2) = \ln \left( \frac{0,184 \cdot 0,184 \cdot 0,25 + 1,359 \cdot 1,359 \cdot 1,847 + 0 \cdot 1,359 \cdot 0,25 + 0 \cdot 0,184 \cdot 1,848}{0,184 \cdot 1,359 \cdot 0,25 + 1,359 \cdot 0,184 \cdot 0,034 + 0 \cdot 0,184 \cdot 0,25 + 0 \cdot 1,359 \cdot 0,034} \right) = 3,875.$$

Для  $k = 3$  согласно формуле (4.10):

$$L_e(d_3) = \ln \left( \frac{0,25 \cdot 0,184 \cdot 0 + 0,034 \cdot 1,359 \cdot 0 + 0,25 \cdot 1,359 \cdot 0 + 1,847 \cdot 0,184 \cdot 0}{0,25 \cdot 1,359 \cdot 0,184 + 0,034 \cdot 0,184 \cdot 1,359 + 0,25 \cdot 0,184 \cdot 0,184 + 1,847 \cdot 1,359 \cdot 1,359} \right) = -\infty.$$

Для  $k = 4$  согласно формуле (4.10):

$$L_e(d_4) = \ln \left( \frac{0,386 \cdot 1,359 \cdot 0 + 0,386 \cdot 0,184 \cdot 0 + 2,517 \cdot 0,184 \cdot 0 + 0,386 \cdot 1,359 \cdot 0}{0,386 \cdot 0,184 \cdot 1 + 0,386 \cdot 1,359 \cdot 0 + 2,517 \cdot 1,359 \cdot 1 + 0,386 \cdot 0,184 \cdot 0} \right) = -\infty.$$

Выход первого декодера сформирован.

Выход второго декодера рассчитывается аналогично с учетом выше сказанных замечаний. По итогам работы второго декодера получены следующие результаты расчета логарифмической функции правдоподобия для всех информационных бит:  $L(d_1) = 1,325$ ;  $L(d_2) = 0$ ;  $L(d_3) = 0$ ;  $L(d_4) = 0$ .

Теперь имеются все необходимые данные для расчета по формуле (4.8) мягкого выхода декодера турбокода по итогам одной полной итерации декодирования. Для удобства восприятия результаты расчета приведены в таблице 4.17. Жесткий выход турбокода в соответствии с критерием (4.9) не содержит ошибок.

При перфорации проверочных бит первого кодера входная последовательность декодера турбокода выглядит следующим образом: 10–1 10–1 –101 –10–1. Исходя из входной последовательности декодера турбокода, на вход первого декодера поступает последовательность 10 10 –10 –10, а на вход второго

декодера  $-1-1 -1-1 11 1-1$ . Для удобства, значения входов декодеров для данного случая приведены в таблице 4.18.

Таблица 4.17

Результаты расчета мягкого выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	0	3,876	1,325	5,201	1
2	0	3,876	0	3,876	1
3	0	$-\infty$	0	$-\infty$	0
4	0	$-\infty$	0	$-\infty$	0

Результаты расчета мягкого выхода декодера турбокода приведены в таблице 4.19. Жесткий выход турбокода в соответствии с критерием (4.9) не содержит ошибок.

Таблица 4.18

Входы декодеров при перфорации проверочных бит первого кодера

		$k=$		1	2	3	4
Вход первого декодера	$x_k$	№		0	3	6	9
		значение		1	1	-1	-1
	$y_k$	№		1	4	7	10
		значение		0	0	0	0
Вход второго декодера	$x_k$	№		6	9	0	3
		значение		-1	-1	1	1
	$y_k$	№		2	5	8	11
		значение		-1	-1	1	-1

Таблица 4.19

Результаты расчета мягкого выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	1	2	4,897	7,897	1
2	1	2	3,843	6,843	1
3	-1	$-\infty$	-6,794	$-\infty$	0
4	-1	$-\infty$	-5,567	$-\infty$	0

При перфорации проверочных бит второго кодера входная последовательность декодера турбокода выглядит следующим образом: 110 1–10 –1–10 –110. Исходя из входной последовательности декодера турбокода, на вход первого декодера поступает последовательность 11 1–1 –1–1 –11, а на вход второго декодера –10 –10 10 10. Для удобства, значения входов декодеров для данного случая приведены в таблице 4.20.

Результаты расчета мягкого выхода декодера турбокода приведены в таблице 4.21. Жесткий выход турбокода, в соответствии с критерием (4.9), не содержит ошибок.

Сравнительные результаты расчета мягкого выхода декодера турбокода для разных видов перфорации приведены в таблице 4.22. Наибольшей достоверностью обладают результаты декодирования последовательности с перфорацией проверочных бит второго кодера. Наименьшей достоверностью обладают результаты декодирования последовательности с перфорацией информационных бит. Результаты сравнения подтверждают выдвинутое предположение о наибольшей целесообразности введения перфорации в последовательность проверочных бит второго кодера, как наименее ценную из всех трех последовательностей.

Таблица 4.20

Входы декодеров при перфорации проверочных бит второго кодера

			$k=$	1	2	3	4
Вход первого декодера	$x_k$	№	0	3	6	9	
		значение	1	1	–1	–1	
	$y_k$	№	1	4	7	10	
		значение	1	–1	–1	1	
Вход второго декодера	$x_k$	№	6	9	0	3	
		значение	–1	–1	1	1	
	$y_k$	№	2	5	8	11	
		значение	0	0	0	0	



Таблица 4.21

Результаты расчета мягкого выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	1	7,307	2	10,307	1
2	1	7,307	2	10,307	1
3	-1	$-\infty$	-2	$-\infty$	0
4	-1	$-\infty$	-2	$-\infty$	0

Таблица 4.22

Значения выхода декодера турбокода при всех вариантах перфорации

№ бита	Виды перфорации			значение переданного бита
	перфорация информационных бит	перфорация проверочных бит первого кодера	перфорация проверочных бит второго кодера	
1	5,201	7,897	10,307	1
2	3,876	6,843	10,307	1
3	$-\infty$	$-\infty$	$-\infty$	0
4	$-\infty$	$-\infty$	$-\infty$	0

Рассмотрим результаты декодирования принятой перфорированной последовательности, содержащей ошибочные биты. Пусть каждая из последовательностей, не подвергшихся перфорации, содержит по одному ошибочному биту. Кодовая последовательность и выходы детектора для разных видов перфорации приведены в таблице 4.23. Ошибочные биты выделены подчеркиванием.

При перфорации всех информационных бит и наличии ошибок в каждой из последовательностей проверочных бит, последовательность на входе декодера турбокода имеет вид  $01\underline{1} 0-1-1 0-11 0-\underline{1}-1$ , где ошибочные биты подчеркнуты. Тогда на вход первого декодера поступает последовательность  $01 0-1 0-1 0-\underline{1}$ , а на вход второго декодера  $0\underline{1} 0-1 01 0-1$ . Для удобства, значения входов декодеров для данного случая приведены в таблице 4.24.

Кодовая последовательность и выходы детекторов

	0	1	2	3	4	5	6	7	8	9	10	11
	$x$	$y_1$	$y_2$	$x$	$y_1$	$y_2$	$x$	$y_1$	$y_2$	$x$	$y_1$	$y_2$
Кодовая последовательность	1	1	-1	1	-1	-1	-1	-1	1	-1	1	-1
Выход детектора с ошибками и перфорацией информационных бит	0	1	<u>1</u>	0	-1	-1	0	-1	1	0	<u>-1</u>	-1
Выход детектора с ошибками и перфорацией проверочных бит первого кода	1	0	<u>1</u>	1	0	-1	<u>1</u>	0	1	-1	0	-1
Выход детектора с ошибками и перфорацией проверочных бит второго кода	1	1	0	1	-1	0	<u>1</u>	-1	0	-1	<u>-1</u>	0

Результаты расчета мягкого выхода декодера турбокода приведены в таблице 4.25. В соответствии с полученным мягким выходом, по критерию (4.9) второй бит информационной последовательности декодирован с ошибкой.

Таблица 4.24

Входы декодеров при перфорации информационных бит

			$k=$	1	2	3	4
Вход первого декодера	$x_k$	№	0	3	6	9	
		значение	0	0	<u>0</u>	0	
	$y_k$	№	1	4	7	10	
		значение	1	-1	-1	<u>-1</u>	
Вход второго декодера	$x_k$	№	6	9	0	3	
		значение	<u>0</u>	0	0	0	
	$y_k$	№	2	5	8	11	
		значение	<u>1</u>	-1	1	-1	

Результаты расчета выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	0	0,124	1,325	1,449	1
2	0	-0,124	-0,949	-1,073	0
3	0	$-\infty$	2	$-\infty$	0
4	0	$-\infty$	1,325	$-\infty$	0

При перфорации проверочных бит первого кодера и наличии ошибок в информационной последовательности и последовательности проверочных бит второго кодера, последовательность на входе декодера турбокода имеет вид  $10\underline{1}10-1\underline{1}01-10-1$ , где ошибочные биты подчеркнуты. Тогда на вход первого декодера поступает последовательность  $1010\underline{1}0-10$ , а на вход второго декодера  $\underline{11}-1-111-1$ . Для удобства, значения входов декодеров для данного случая приведены в таблице 4.26.

Таблица 4.26

Входы декодеров при перфорации проверочных бит первого кодера

		$k=$	1	2	3	4
Вход первого декодера	$x_k$	№	0	3	6	9
		значение	1	1	<u>1</u>	-1
	$y_k$	№	1	4	7	10
		значение	0	0	0	<u>0</u>
Вход второго декодера	$x_k$	№	6	9	0	3
		значение	<u>1</u>	-1	1	1
	$y_k$	№	2	5	8	11
		значение	<u>1</u>	-1	1	-1

Результаты расчета мягкого выхода декодера турбокода приведены в таблице 4.27. Жесткий выход турбокода в соответствии с критерием (4.9) не содержит ошибок.

При перфорации проверочных бит второго кодера и наличии ошибок в информационной последовательности и последовательности проверочных бит первого кодера, последовательность на входе декодера турбокода имеет вид  $110$

1-10 1-10 -110, где ошибочные биты подчеркнуты. Тогда на вход первого декодера поступает последовательность 11 1-1 1-1 -11, а на вход второго декодера 10 -10 10 10. Для удобства, значения входов декодеров для данного случая приведены в таблице 4.28.

Таблица 4.27

Результаты расчета мягкого выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	1	2	2,273	5,273	1
2	1	2	1,023	4,023	1
3	1	$-\infty$	1,206	$-\infty$	0
4	-1	$-\infty$	-0,077	$-\infty$	0

Результаты расчета мягкого выхода декодера турбокода приведены в таблице 4.29. Жесткий выход турбокода в соответствии с критерием (4.9) не содержит ошибок.

Таблица 4.28

Входы декодеров при перфорации проверочных бит второго кодера

		$k=$	1	2	3	4
Вход первого декодера	$x_k$	№	0	3	6	9
		значение	1	1	<u>1</u>	-1
	$y_k$	№	1	4	7	10
		значение	1	-1	-1	<u>-1</u>
Вход второго декодера	$x_k$	№	6	9	0	3
		значение	<u>1</u>	-1	1	1
	$y_k$	№	2	5	8	11
		значение	<u>0</u>	0	0	0

Сравнительные результаты расчета мягкого выхода декодера турбокода для разных видов перфорации при наличии ошибок в неперфорированных последовательностях приведены в таблице 4.30. Наибольшей достоверностью обладают результаты декодирования последовательности с перфорацией проверочных бит второго кодера. Наименьшей достоверностью обладают

результаты декодирования последовательности с перфорацией информационных бит (содержащие ошибку во втором информационном бите). Как и в случаях отсутствия ошибок, результаты сравнения подтверждают выдвинутое предположение о наибольшей целесообразности введения перфорации в последовательность проверочных бит второго кодера, как наименее ценную из всех трех последовательностей.

Таблица 4.29

Результаты расчета мягкого выхода декодера турбокода

$k$	$L_c(d_k)$	$L'_e(d_k)$	$L_e(d_k)$	$L(d_k)$	$d_k$
1	1	4	2	7	1
2	1	3,3	2	6,3	1
3	1	$-\infty$	2	$-\infty$	0
4	-1	$-\infty$	-2	$-\infty$	0

Таблица 4.30

Значения выхода декодера турбокода при всех вариантах перфорации

№ бита	Виды перфорации			значение переданного бита
	перфорация информационных бит	перфорация проверочных бит первого кодера	перфорация проверочных бит второго кодера	
1	1,449	5,273	7	1
2	-1,073	4,023	6,3	1
3	$-\infty$	$-\infty$	$-\infty$	0
4	$-\infty$	$-\infty$	$-\infty$	0

#### 4.2.2. Оценка корректирующей способности турбокода при введении перфорации

В п. 4.2.1 по итогам рассмотрения примеров декодирования перфорированных кодовых слов турбокода определено, что наиболее целесообразным является введение перфорации в проверочные биты кодового слова турбокода. В данном разделе предлагается произвести оценку корректирующей способности турбокода, кодирование и декодирование кодового слова которого описано в п. 4.2.1, в

зависимости от количества выколотых бит из числа проверочных бит кодового слова турбокода.

Проведем исследования успешности декодирования кодового слова турбокода на основе систематического сверточного кода (далее – ТССК) при разном количестве и расположении ошибок [183]. На основе анализа результатов исследования осуществим вывод формулы для расчета вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$ . С целью проведения исследований в MS Excel произведена разработка реализации описанного в п. 4.2.1 декодера одного кодового слова ТССК, осуществляющего три итерации декодирования жесткого решения детектора, заданного пользователем кодового слова.

Как показали исследования, декодер турбокода при использовании алгоритма MAP способен исправлять две ошибки в кодовом слове вне зависимости от их расположения. Рассмотрим возможности декодера по исправлению трех ошибок при их произвольном расположении. Выделим три случая расположения ошибочных бит в кодовом слове турбокода:

– 1 случай – все три ошибки располагаются в информационных битах (в этом случае существует 4 варианта расположения ошибок);

– 2 случай – ошибочные биты располагаются как среди информационных бит, так и среди проверочных бит (136 вариантов расположения ошибок);

– 3 случай – все три ошибки располагаются в проверочных битах (56 вариантов расположения ошибок).

Рассмотрев результаты декодирования, приведенные в таблице 4.31, можно прийти к следующим заключениям. Успешное декодирование кодового слова происходит в 163 вариантах из 196 возможных, то есть в 83.2 % случаев. Наибольший процент успешного декодирования наблюдается для случая отсутствия ошибок в информационных битах. Еще раз подтверждается суждение, что наиболее негативный эффект на декодирование кодового слова ТССК оказывают ошибки, возникшие в информационных битах.

Количество вариантов исправления трех ошибок в кодовом слове

Случаи расположения ошибок			Всего вариантов исправления
1 случай	2 случай	3 случай	
3 (75 %)	110 (81 %)	50 (89 %)	163 (83.2 %)

Произведем оценку корректирующей способности ТССК в канале с независимыми ошибками. Для этого рассчитаем значение вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$  и сравним со значениями  $p_D$  некоторых известных кодов.

Так как вероятность наличия трех ошибочных бит в кодовом слове значительно превышает вероятность возникновения большего числа ошибок, то с незначительной долей приближения можно считать, что ошибочное декодирование кодового слова вызвано наличием трех ошибок. Вероятность появления трех ошибочных бит из двенадцати можно найти по формуле Бернулли:

$$P_{\text{Зиз12}} = (12! / (3! \cdot (12 - 3)!)) \cdot p_B^3 \cdot (1 - p_B)^{12-3} = 220 \cdot p_B^3 \cdot (1 - p_B)^9, \quad (4.11)$$

где  $p_B$  вероятность появления ошибочного бита в канале передачи данных.

С учетом того, что при наличии трех ошибок в кодовом слове ошибочное декодирование осуществляется в 16,8 % случаев, вероятность ошибочного декодирования кодового слова вычисляется по следующей формуле:

$$p_W = 0,168 \cdot P_{\text{Зиз12}} = 37 \cdot p_B^3 \cdot (1 - p_B)^9. \quad (4.12)$$

Как показали результаты исследований, из-за ошибочного декодирования кодового слова ошибка возникает в среднем в 1.2 битах из двух информационных бит. Тогда вероятность появления ошибочного бита в декодированном информационном сообщении равна:

$$p_D = (1,2/2) \cdot p_W = 22,2 \cdot p_B^3 \cdot (1 - p_B)^9. \quad (4.13)$$

С целью реализации заявленных исследований корректирующей способности ТССК, на языке программирования Python создан программный имитатор [184]. Программа обеспечивает пользователя возможностью оценить эффективность работы турбокода, построенного на основе систематического сверточного кода при

искусственном введении ошибок в массив информации и последующем декодированием.

Программа позволяет:

- задавать количество итераций кодирования-декодирования массива размерностью 1000 бит;
- задавать вероятность появления ошибочного бита в массиве, моделируя появление ошибок в канале передачи данных;
- задавать вид входных данных для декодирования: квантованный на восемь уровней сигнал (мягкий) или бинарный сигнал (жесткий);
- получать по итогам завершения всех итераций кодирования-декодирования следующую информацию: число бит информации, число кодовых бит, число сгенерированных ошибок, число ошибок после декодирования, вероятность ошибочного бита.

С использованием описанного программного имитатора произведен расчет вероятности битовой ошибки для описанного ТССК с двумя терминирующими битами. Результаты работы программного имитатора в сравнении с расчетами по формуле (4.13) приведены в таблице 4.32. Также в таблице приведены результаты работы программы при отсутствии терминирующих бит в кодовом слове. Все результаты приведены для случая декодирования жесткого решения детектора. Рассмотрев данные таблицы 4.32 можно сделать следующие выводы. Наличие терминирующих бит повышает корректирующую способность турбокода. Расчеты по формуле (4.13) для значений  $p_B \leq 1 \cdot 10^{-2}$  совпадают с результатами работы программного имитатора.

Для короткой информационной последовательности наличие терминирующих бит существенно снижает скорость передачи информации. Далее проведем исследования корректирующей способности перфорированного ТССК при отсутствии в кодовом слове терминирующих бит. Результаты работа программного имитатора при декодировании жесткого и мягкого решения детектора приведены в таблицах 4.33 и 4.34 соответственно. Анализ результатов показал возможность



плавной регулировки избыточности и корректирующей способности турбокода путем изменения числа выколотых бит.

Таблица 4.32

Вероятность появления ошибочного бита в декодированном информационном сообщении при декодировании жесткого решения детектора

Применяемый код	$p_B=10^{-1}$	$p_B=10^{-2}$	$p_B=10^{-3}$	$p_B=10^{-4}$
ТССК с двумя терминирующими битами (по формуле 4.13)	$8,60 \cdot 10^{-3}$	$2,02 \cdot 10^{-5}$	$2,20 \cdot 10^{-8}$	$2,22 \cdot 10^{-11}$
ТССК с двумя терминирующими битами (результаты работы программного имитатора)	$1,72 \cdot 10^{-2}$	$2,04 \cdot 10^{-5}$	$2,19 \cdot 10^{-8}$	$2,22 \cdot 10^{-11}$
ТССК без терминирующих бит (результаты работы программного имитатора)	$3,09 \cdot 10^{-2}$	$1,31 \cdot 10^{-4}$	$8,50 \cdot 10^{-7}$	$4,69 \cdot 10^{-9}$

Таблица 4.33

Вероятность появления ошибочного бита в декодированном информационном сообщении при декодировании жесткого решения детектора

Число выкол	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
0	$3,1 \cdot 10^{-2}$	$1,3 \cdot 10^{-4}$	$8,5 \cdot 10^{-7}$	$4,7 \cdot 10^{-9}$
1	$3,9 \cdot 10^{-2}$	$3,49 \cdot 10^{-4}$	$3,4 \cdot 10^{-6}$	$2,4 \cdot 10^{-8}$
2	$4,9 \cdot 10^{-2}$	$5,46 \cdot 10^{-4}$	$6,1 \cdot 10^{-6}$	$6,0 \cdot 10^{-8}$
3	$5,6 \cdot 10^{-2}$	$6,8 \cdot 10^{-4}$	$7,8 \cdot 10^{-6}$	$8,0 \cdot 10^{-8}$
4	$5,6 \cdot 10^{-2}$	$6,8 \cdot 10^{-4}$	$8,5 \cdot 10^{-6}$	$1,01 \cdot 10^{-7}$
5	$8,8 \cdot 10^{-2}$	$7,8 \cdot 10^{-3}$	$7,37 \cdot 10^{-4}$	$7,6 \cdot 10^{-5}$

В параграфе рассмотрен распространенный алгоритм декодирования турбокодов MAP, часто упоминаемый в последующих параграфах. Произведена оценка корректирующей способности турбокодов при введении перфорации в разные части кодового слова турбокода. Наименьшей достоверностью обладают результаты декодирования последовательности с перфорацией информационных бит кодового слова турбокода. Сделан вывод о высокой чувствительности декодера

турбокода к достоверности значений информационных бит. Введение перфорации позволяет плавно изменять корректирующую способность турбокода и кодовую скорость. При декодировании жесткого решение детектора изменение кодовой скорости возможно от 0,33 до 0,57, то есть на 72 %. При декодировании мягкого решение детектора изменение кодовой скорости возможно от 0,33 до 0,8, то есть на 142 %. Далее рассмотрим возможности по декодированию блочных и сверточных кодов рассмотренным в данном параграфе декодером турбокода, работающим по принципу максимизации апостериорной информации.

Таблица 4.34

Вероятность появления ошибочного бита в декодированном информационном сообщении при декодировании мягкого решения детектора

Число выкал	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
0	$9,55 \cdot 10^{-3}$	$2,81 \cdot 10^{-6}$	$3,50 \cdot 10^{-9}$	$2,91 \cdot 10^{-12}$
1	$1,36 \cdot 10^{-2}$	$8,8 \cdot 10^{-6}$	$1,09 \cdot 10^{-8}$	$2,5 \cdot 10^{-11}$
2	$1,96 \cdot 10^{-2}$	$2,9 \cdot 10^{-5}$	$1,01 \cdot 10^{-7}$	$2,4 \cdot 10^{-10}$
3	$2,64 \cdot 10^{-2}$	$1,18 \cdot 10^{-4}$	$7,0 \cdot 10^{-7}$	$1,2 \cdot 10^{-9}$
4	$3,4 \cdot 10^{-2}$	$4,67 \cdot 10^{-4}$	$1,67 \cdot 10^{-5}$	$9,0 \cdot 10^{-7}$
5	$4,4 \cdot 10^{-2}$	$6,6 \cdot 10^{-4}$	$1,75 \cdot 10^{-5}$	$9,6 \cdot 10^{-7}$
6	$6,8 \cdot 10^{-2}$	$2,6 \cdot 10^{-3}$	$1,47 \cdot 10^{-4}$	$9,8 \cdot 10^{-6}$
7	$8,3 \cdot 10^{-2}$	$5,5 \cdot 10^{-3}$	$4,5 \cdot 10^{-4}$	$3,5 \cdot 10^{-5}$
8	$9,9 \cdot 10^{-2}$	$1,0 \cdot 10^{-2}$	$9,9 \cdot 10^{-4}$	$1,0 \cdot 10^{-4}$

### 4.3. Исследование возможности декодирования блочного и сверточного кода с использованием декодера турбокода

Кодовое слово турбокода состоит из блока информационных бит и  $N$  блоков проверочных бит. Если к кодовому слову систематического блочного кода (далее – СБК) или ССК добавить еще один или несколько блоков информационных бит, то в приемном устройстве будет возможно формирование нескольких вариантов

кодového слова, при декодировании которых может быть использован декодер турбокода. Назовем такой комбинированный код кодом с повтором информационных бит (далее – КПИБ). Структурная схема кодера КПИБ представлена на рисунке 4.15. Кодовое слово КПИБ состоит из блока проверочных бит и  $N$  блоков информационных бит. Кодер КПИБ не в состоянии сформировать кодовые слова, обладающие квазислучайными свойствами, однако он значительно проще кодера турбокода, особенно при больших значениях  $N$ . В данном параграфе будет произведено исследование эффективности КПИБ при декодировании с применением декодера турбокода, при реализации декодирования в соответствии с критерием максимума апостериорной вероятности [185, 186].

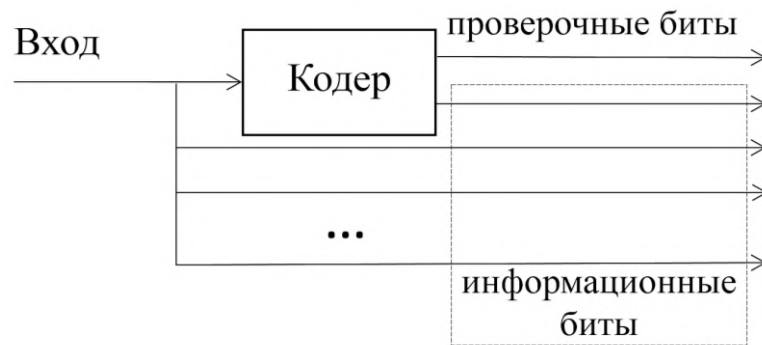


Рисунок 4.15. Структурная схема кодера КПИБ

Рассмотрим процесс создания кодовых последовательностей КПИБ на базе ССК и СБК и особенности их декодирования с использованием декодера турбокода, рассмотренного в п. 4.2 при его модификации. Для простоты вычислений примем  $N = 2$ , этим определив, что кодовая последовательность состоит из двух блоков информационных бит и блока проверочных бит. Тогда структурная схема кодера принимает вид, представленный на рисунке 4.16.

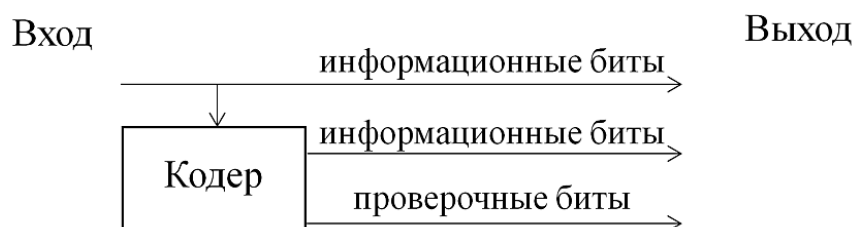


Рисунок 4.16. Структурная схема кодера КПИБ

Для КПИБ, построенного на основе ССК, опишем процесс создания кодовой последовательности. В качестве ССК будем использован код, структурная схема

кодера которого изображена на рисунке 4.17, при длине кодируемого блока информации, равной четырем битам. Процесс кодирования ССК подробно описан в разнообразных источниках [10, 12]. Пусть на вход кодера поступает информационная последовательность 0100. Тогда на выходе кодера формируется последовательность пар бит 00 11 01 01, где первый бит пары – информационный, а второй – проверочный. Последовательность проверочных бит имеет вид: 0111. Кодовое слово КПИБ формируется из проверочных бит и двукратного повторения информационных бит (см. рис. 4.18). Тогда для рассмотренного примера кодовое слово КПИБ имеет вид: 0111 0100 0100, который при передаче биполярными сигналами преобразуется в:  $-1111 -11-1-1 -11-1-1$ .

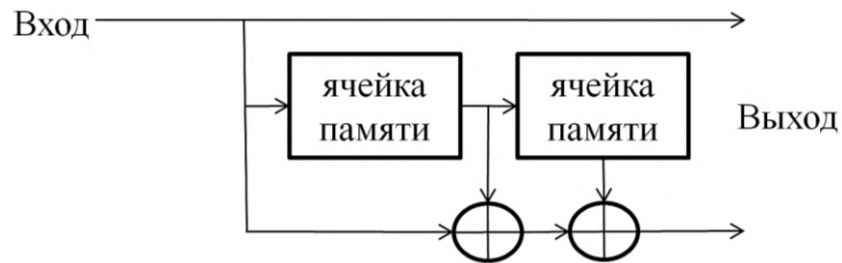


Рисунок 4.17. Структурная схема кодера ССК

Проверочные биты				Информационные биты							
1	2	3	4	Блок 1				Блок 2			
1	2	3	4	5	6	7	8	9	10	11	12

Рисунок 4.18. Структура кодового слова КПИБ на основе выбранного ССК

Опишем процесс создания кодовой последовательности КПИБ на основе СБК. В качестве СБК будем использовать код Хемминга, кодовое слово которого состоит из четырех информационных и трех проверочных бит. Кодирование и декодирование кода Хемминга может быть осуществлено с использованием синдромной решетки, структура которой изображена на рисунке 3.14. При использовании синдромной решетки становится возможно применение алгоритма MAP для декодирования кодового слова КПИБ на основе кода Хемминга. В качестве декодера будет использован декодер турбокода, рассмотренный в п. 4.2.1 после проведенной модификации. Структура кодового слова КПИБ, созданная на

основе кода Хемминга, изображена на рисунке 4.19. Имея информационную последовательность 0100, проверочные биты к которой 011, формируем выход кодера КПИБ: 011 0100 0100 или в биполярном виде:  $-111 -11-1-1 -11-1-1$ .

Проверочные биты			Информационные биты							
			Блок 1				Блок 2			
1	2	3	4	5	6	7	8	9	10	11

Рисунок 4.19. Структура кодового слова КПИБ на основе кода Хемминга

Разберем особенности декодирования принятого кодового слова КПИБ декодером турбокода. Для декодирования будем использовать распространенный модифицированный декодер турбокода [11, 12, 61] с изъятыми из состава перемежителем и расперемежителями, структурная схема которого изображена на рисунке 4.20. Декодирование будет осуществляться с использованием хорошо изученного алгоритма MAP, более подробно описанного в п. 4.2. Модификация алгоритма MAP, позволяющая производить декодирование с использованием нерегулярной синдромной решетки блочного кода, подробно рассмотрена в п. 5.1. При применении алгоритма MAP на выходе каждого из двух составных декодеров формируется мягкое решение. По итогам итерационного декодирования формируется мягкий выход декодера турбокода [12]:

$$L(d_k) = L_e^1(d_k) + L_e^2(d_k) + L_c(d_k),$$

где  $d_k$  – значение  $k$ -го информационного символа,  $L_e^1(d_k)$  – значение логарифмической функции правдоподобия на выходе первого декодера,  $L_e^2(d_k)$  – значение логарифмической функции правдоподобия на выходе второго декодера,  $L_c(d_k)$  – канальное измерение значения символа. В случае предложенного варианта построения кодового слова в наличии имеются два значения канального измерения значения символа из двух блоков информационных бит. Исходя из этого, предлагается вычислять  $L_c(d_k)$  как их среднее арифметическое.

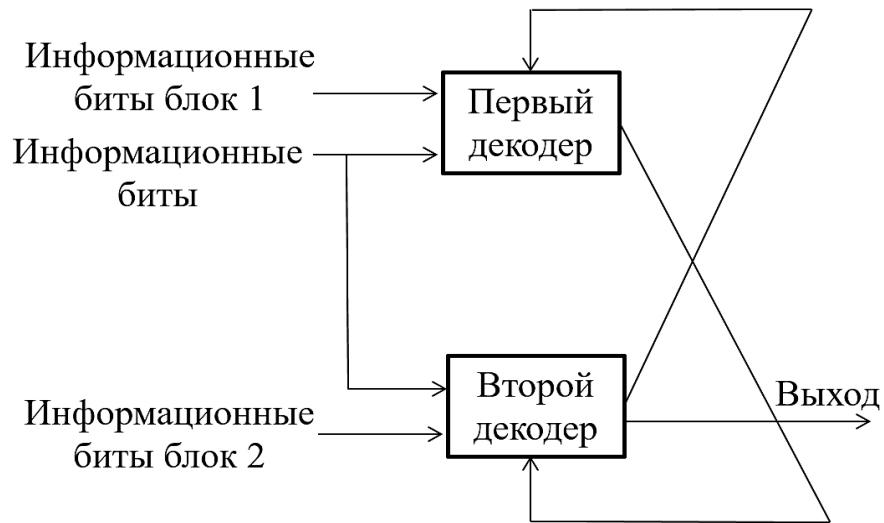


Рисунок 4.20. Декодер КПИБ

С целью исследования корректирующей способности КПИБ на основе ССК и СБК созданы два программных имитатора на языке программирования Python [187, 188]. Программные имитаторы реализуют создание кодовой последовательности, введение в нее ошибок с заданной вероятностью  $p_B$  и декодирование с применением описанного декодера КПИБ.

Рассмотрим результаты работы программных имитаторов для вариантов построения КПИБ на основе выбранных ССК и СБК [185, 186] в сравнении со значениями вероятности битовой ошибки в декодированном информационном сообщении  $p_D$  для блочного кода Хемминга (п.п. 3.2, 3.6) и сверточного кода (п. 2.3), на основе которых были построены КПИБ. В таблице 4.35 приведены значения вероятности появления ошибочного бита в декодированном информационном сообщении для случая декодирования жесткого решения детектора, а в таблице 4.36 – значения  $p_D$  для случая декодирования мягкого решения детектора. По итогам анализа результатов из обеих таблиц можно прийти к выводу о лучшей корректирующей способности КПИБ. Для улучшения восприятия данные из таблиц 4.35 и 4.36 приведены в графическом виде на рисунках 4.21 – 4.24.

Таблица 4.35

Вероятность появления ошибочного бита в декодированном информационном сообщении при декодировании жесткого решения детектора

Применяемый код	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
КПИБ на основе СБК	$4,60 \cdot 10^{-2}$	$6,6 \cdot 10^{-4}$	$6,8 \cdot 10^{-6}$	$6,8 \cdot 10^{-8}$
Код Хемминга	$6,7 \cdot 10^{-2}$	$8,6 \cdot 10^{-4}$	$8,9 \cdot 10^{-6}$	$9,0 \cdot 10^{-8}$
КПИБ на основе ССК	$6,2 \cdot 10^{-2}$	$2,8 \cdot 10^{-3}$	$2,5 \cdot 10^{-4}$	$2,2 \cdot 10^{-5}$
Сверточный код при длине сообщения 4 бита	$7,0 \cdot 10^{-2}$	$4,2 \cdot 10^{-3}$	$4,3 \cdot 10^{-4}$	$4,9 \cdot 10^{-5}$

Таблица 4.36

Вероятность появления ошибочного бита в декодированном информационном сообщении при декодировании мягкого решения детектора

Применяемый код	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
КПИБ на основе СБК	$1,4 \cdot 10^{-2}$	$1,6 \cdot 10^{-5}$	$4,1 \cdot 10^{-8}$	$8,0 \cdot 10^{-11}$
Код Хемминга	$4,1 \cdot 10^{-2}$	$1,7 \cdot 10^{-4}$	$6,3 \cdot 10^{-7}$	$1,3 \cdot 10^{-9}$
КПИБ на основе ССК	$1,9 \cdot 10^{-2}$	$1,2 \cdot 10^{-4}$	$1,5 \cdot 10^{-6}$	$1,6 \cdot 10^{-8}$
Сверточный код при длине сообщения 4 бита	$4,8 \cdot 10^{-2}$	$3,0 \cdot 10^{-4}$	$4,0 \cdot 10^{-5}$	$2,0 \cdot 10^{-5}$

Построение комбинированного кода на основе сверточного и блочного кода при повторной передаче информационных бит и декодировании, согласно максимуму апостериорной вероятности, привело к снижению вероятности битовой ошибки в декодированном информационном сообщении. Для случая построения КПИБ на основе ССК, при вероятности битовой ошибки в канале передачи данных  $p_B=1 \cdot 10^{-3}$ , вероятность битовой ошибки в декодированном информационном сообщении снизилась в 27 раз с  $4,0 \cdot 10^{-5}$  до  $1,5 \cdot 10^{-6}$ . Далее рассмотрим возможность повышения корректирующей способности, основываясь на знаниях построения КПИБ.

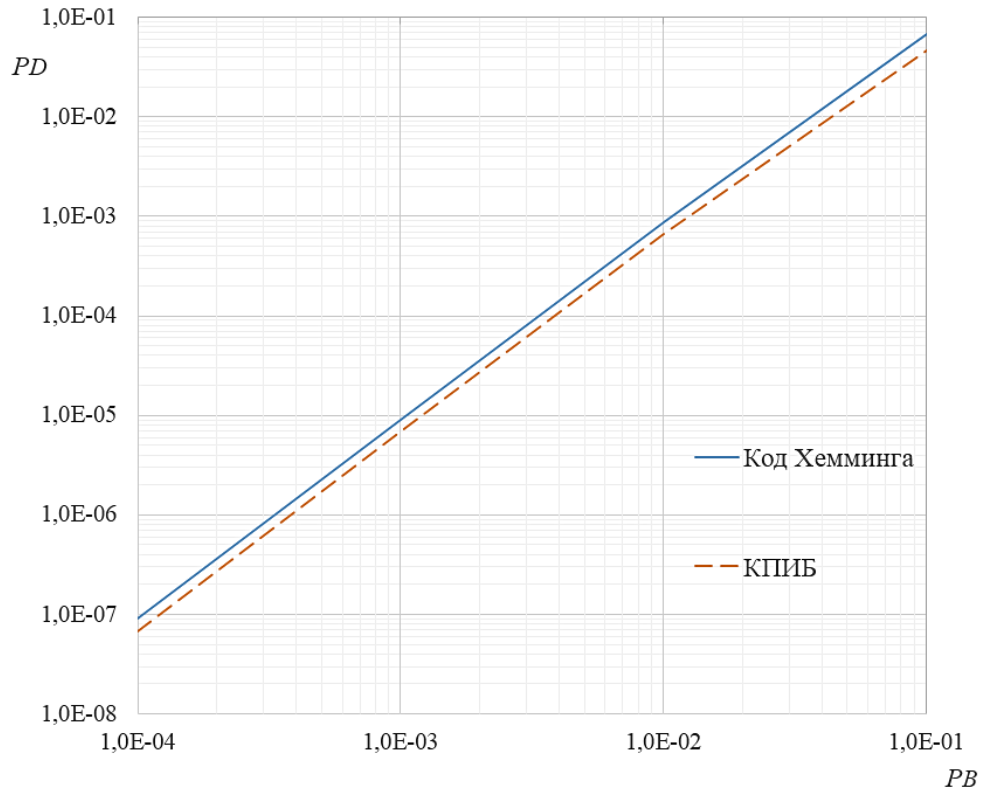


Рисунок 4.21. Вероятность битовой ошибки в декодированном информационном сообщении при декодировании жесткого решения детектора для КПИБ на основе СБП и кода Хемминга

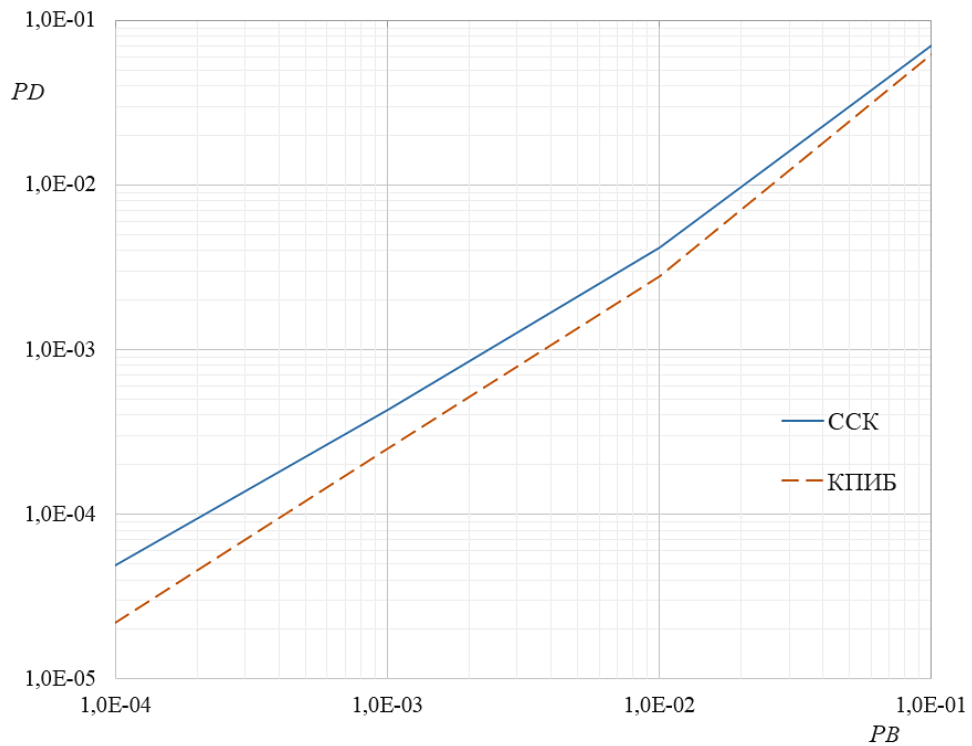


Рисунок 4.22. Вероятность битовой ошибки в декодированном информационном сообщении при декодировании жесткого решения детектора для КПИБ на основе ССК и ССК



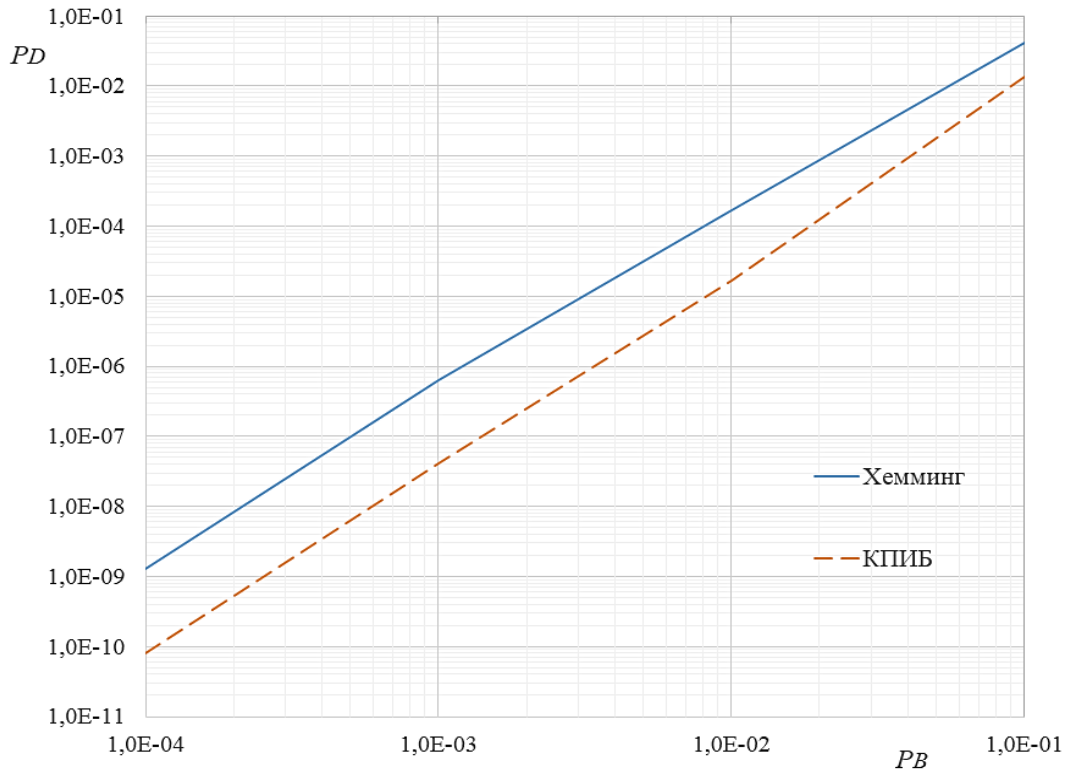


Рисунок 4.23. Вероятность битовой ошибки в декодированном информационном сообщении при декодировании мягкого решения детектора для КПИБ на основе СБП и кода Хемминга

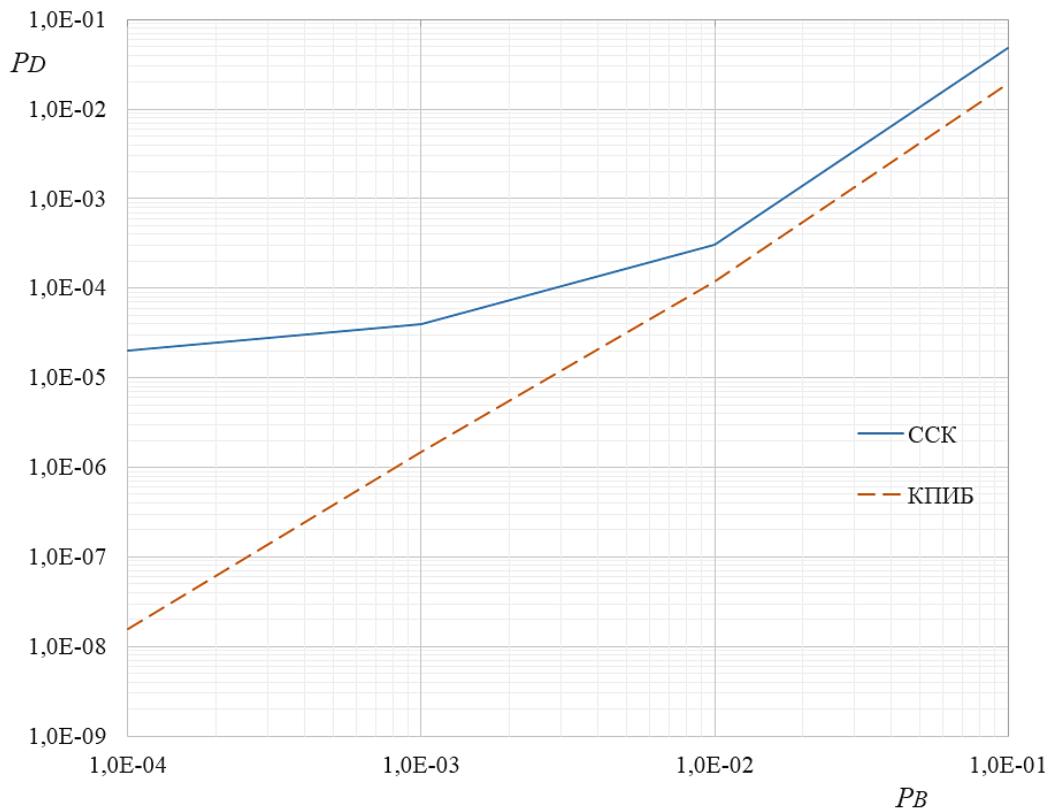


Рисунок 4.24. Вероятность битовой ошибки в декодированном информационном сообщении при декодировании мягкого решения детектора для КПИБ на основе ССК и ССК

#### 4.4. Повышение корректирующей способности КПИБ путем снижения чувствительности декодера к достоверности значений проверочных бит кодового слова

Кодовое слово КПИБ состоит из одного блока проверочных и нескольких блоков информационных бит. Символы блока проверочных бит участвуют в формировании нескольких слов на входе декодера КПИБ. Выскажем предположение, что причина относительно высокого значения  $p_D$  при декодировании КПИБ – высокая чувствительность декодера к достоверности значений проверочных бит кодового слова [185, 186]. Негативной стороной такой чувствительности является высокая вероятность ошибки декодирования при наличии ошибок в проверочных битах кодового слова. Для оценки корректирующей способности при расположении ошибок в разных частях кодового слова произведена разработка в MS Excel реализации декодера КПИБ, рассмотренного в п. 4.3. Подтвердим высказанное предположение о причинах относительно высокого значения  $p_D$ , рассмотрев результаты декодирования кодовых слов КПИБ. Декодирование кодовых слов КПИБ на основе СБК и ССК будет произведено при расположении двух ошибок сначала в проверочных, затем в информационных битах. Для декодирования используем имитатор декодера одного слова КПИБ, разработанный в MS Excel, произведя расчеты мягкого выхода декодера при трех итерациях декодирования.

Для исследований используем кодовое слово КПИБ на основе ССК:  $-1111-11-1-1-11-1-1$ , формирование которого было описано выше. При наличии двух ошибок в проверочных битах кодовое слово принимает вид:  $\underline{11}-\underline{11}-11-1-1-11-1-1$  (ошибочные биты подчеркнуты). По результатам декодирования, приведенным в таблице 4.37, бит № 1 декодирован ошибочно.

При наличии двух ошибок в информационных битах кодовое слово принимает вид:  $-1111-\underline{111}-1-11-1-1$ . По результатам декодирования, приведенным в таблице 4.38, в декодированных информационных битах нет ошибок.

Таблица 4.37

Значения выхода декодера турбокода при двух ошибках в проверочных битах

№ бита	значение переданного бита	Итерации декодирования		
		1	2	3
1	-1	-0,2	3,5	11,0
2	1	4,8	11,4	26,8
3	-1	-3,2	-9,5	-26,6
4	-1	-4,8	-12,9	-29,7

Таблица 4.38

Значения выхода декодера турбокода при двух ошибках в информационных битах

№ бита	значение переданного бита	Итерации декодирования		
		1	2	3
1	-1	-6,2	-19,9	-49,7
2	1	4,8	17,4	44,6
3	-1	-7,3	-19,1	-44,3
4	-1	-4,6	-10,7	-25,1

Рассмотрим декодирование кодового слова КПИБ на основе СБК. Выберем кодовое слово  $-111 \ -11-1-1 \ -11-1-1$ , формирование которого было описано ранее. При наличии двух ошибок в проверочных битах кодовое слово принимает вид:  $\underline{11-1} \ -11-1-1 \ -11-1-1$ . По результатам декодирования, приведенным в таблице 4.39, бит № 4 декодирован ошибочно.

Таблица 4.39

Значения выхода декодера турбокода при двух ошибках в проверочных битах

№ бита	значение переданного бита	Итерации декодирования		
		1	2	3
1	-1	-5,00	-16,03	-39,05
2	1	5,00	16,03	39,05
3	-1	-5,00	-15,50	-41,99
4	-1	0,49	9,63	28,26

При наличии двух ошибок в информационных битах кодовое слово принимает вид:  $-111 \ \underline{111-1} \ -11-1-1$ . По результатам декодирования, приведенным в таблице 4.40, в декодированных информационных битах нет ошибок.

Значения выхода декодера турбокода при двух ошибках в информационных битах

№ бита	значение переданного бита	Итерации декодирования		
		1	2	3
1	-1	-2,75	-7,80	-15,64
2	1	7,75	18,49	39,98
3	-1	-2,75	-10,16	-24,33
4	-1	-7,75	-18,49	-39,98

Рассмотренные примеры подтвердили высказанное предположение о том, что причина высокого значения  $p_D$  для КПИБ – слабая корректирующая способность при наличии ошибок в проверочных битах. Выскажем предположение, что снизить влияние высокой чувствительности проверочных бит к ошибкам, и таким образом повысить корректирующую способность КПИБ, можно путем умножения значений проверочных бит на входе декодера на понижающий коэффициент  $K < 1$  [78]. Подтвердим высказанное предположение, рассмотрев результаты декодирования таким модифицированным методом кодового слова КПИБ на основе ССК при наличии двух ошибок в проверочных битах. Пусть  $K = 0,5$ , тогда декодируемое кодовое слово имеет вид: 0,5 0,5 -0,5 0,5 -11-1-1 -11-1-1. По результатам декодирования, приведенным в таблице 4.41, в информационных битах нет ошибок. Предположение подтверждено практическим примером.

Проведем повторные исследования КПИБ на основе выбранных ССК и СБК с применением описанных в п. 4.3 программных имитаторов с модифицированными алгоритмами декодирования, позволяющими реализовать умножение значений проверочных бит на входе декодера на понижающие коэффициенты [187, 188]. Результаты работы программных имитаторов приведены в таблице 4.42 для случая декодирования жесткого решения детектора и в таблице 4.43 для случая декодирования мягкого решения детектора. В таблицах в скобках приведены значения коэффициента  $K$ , при которых получено минимальное значение  $p_D$ . По сравнению с результатами, приведенными в таблицах 4.33 и 4.34, наблюдается значительный рост корректирующей способности КПИБ, как для случая

декодирования жесткого решения детектора, так и для случая декодирования мягкого решения детектора.

Таблица 4.41

Значения выхода декодера турбокода при двух ошибках в проверочных битах

№ бита	значение переданного бита	Итерации декодирования		
		1	2	3
1	-1	-3,8	-7,6	-15,1
2	1	6,5	18,6	43,2
3	-1	-4,5	-11,0	-24,1
4	-1	-5,3	-14,3	-32,5

Таблица 4.42

Вероятность появления ошибочного бита в декодированном информационном сообщении при декодировании жесткого решения детектора

Применяемый код	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
КПИБ на основе СБК	$4,60 \cdot 10^{-2}$	$6,6 \cdot 10^{-4}$	$6,8 \cdot 10^{-6}$	$6,8 \cdot 10^{-8}$
КПИБ на основе СБК при $K < 1$	$3,60 \cdot 10^{-2}$ ( $K=0,3$ )	$2,5 \cdot 10^{-4}$ ( $K=0,5$ )	$1,9 \cdot 10^{-6}$ ( $K=0,6$ )	$1,7 \cdot 10^{-8}$ ( $K=0,6$ )
КПИБ на основе ССК	$6,2 \cdot 10^{-2}$	$2,8 \cdot 10^{-3}$	$2,5 \cdot 10^{-4}$	$2,2 \cdot 10^{-5}$
КПИБ на основе ССК при $K < 1$	$2,8 \cdot 10^{-2}$ ( $K=0,4$ )	$1,7 \cdot 10^{-4}$ ( $K=0,3$ )	$1,2 \cdot 10^{-6}$ ( $K=0,4$ )	$1,3 \cdot 10^{-8}$ ( $K=0,4$ )

Таблица 4.43

Вероятность появления ошибочного бита в декодированном информационном сообщении при декодировании мягкого решения детектора

Применяемый код	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
КПИБ на основе СБК	$1,35 \cdot 10^{-2}$	$1,65 \cdot 10^{-5}$	$4,10 \cdot 10^{-8}$	$8,01 \cdot 10^{-11}$
КПИБ на основе СБК при $K < 1$	$1,11 \cdot 10^{-2}$ ( $K=0,5$ )	$7,2 \cdot 10^{-6}$ ( $K=0,6$ )	$7,5 \cdot 10^{-9}$ ( $K=0,6$ )	$1,5 \cdot 10^{-11}$ ( $K=0,6$ )
КПИБ на основе ССК	$1,9 \cdot 10^{-2}$	$1,2 \cdot 10^{-4}$	$1,5 \cdot 10^{-6}$	$1,55 \cdot 10^{-8}$
КПИБ на основе ССК при $K < 1$	$1,0 \cdot 10^{-2}$ ( $K=0,4$ )	$1,37 \cdot 10^{-5}$ ( $K=0,5$ )	$4,2 \cdot 10^{-8}$ ( $K=0,4$ )	$4,0 \cdot 10^{-10}$ ( $K=0,4$ )

Для улучшения восприятия, данные из таблиц 4.42 и 4.43 представлены в графическом виде на рисунках 4.25 – 4.28. Пунктиром приведены графики по данным из п. 4.3 без применения коэффициентов.

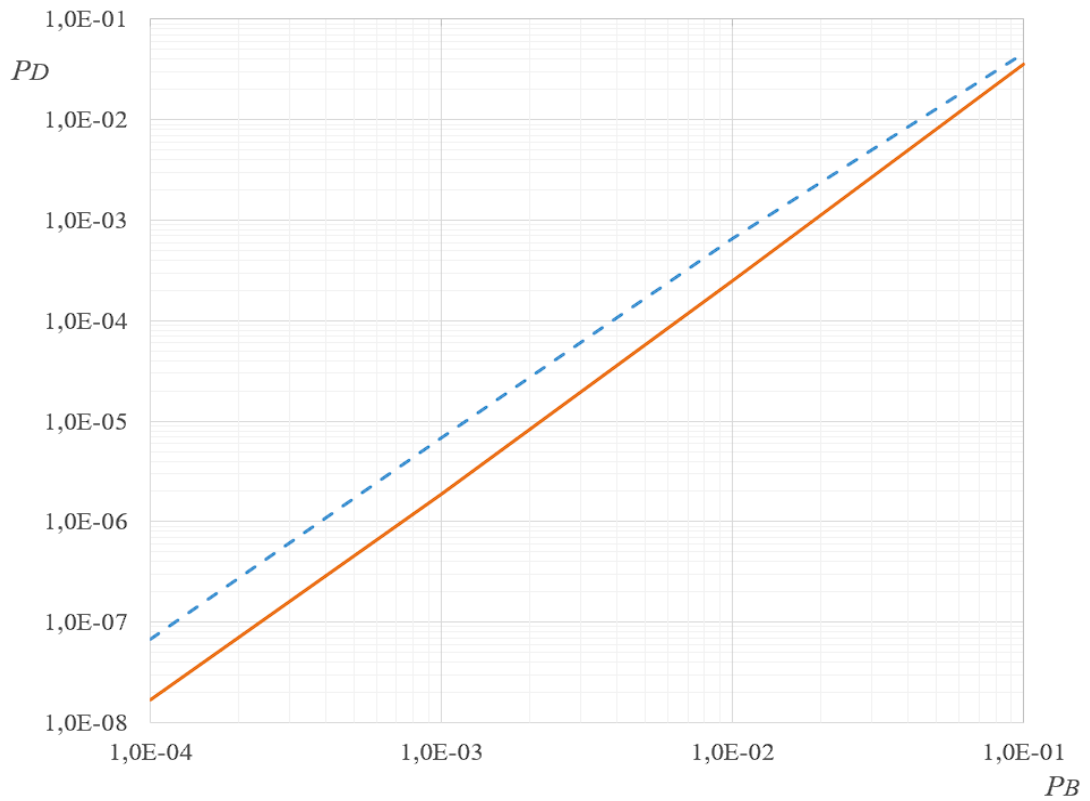


Рисунок 4.25. Оценка корректирующей способности КПИБ на основе СБК для случая декодирования жесткого решения детектора при использовании коэффициентов

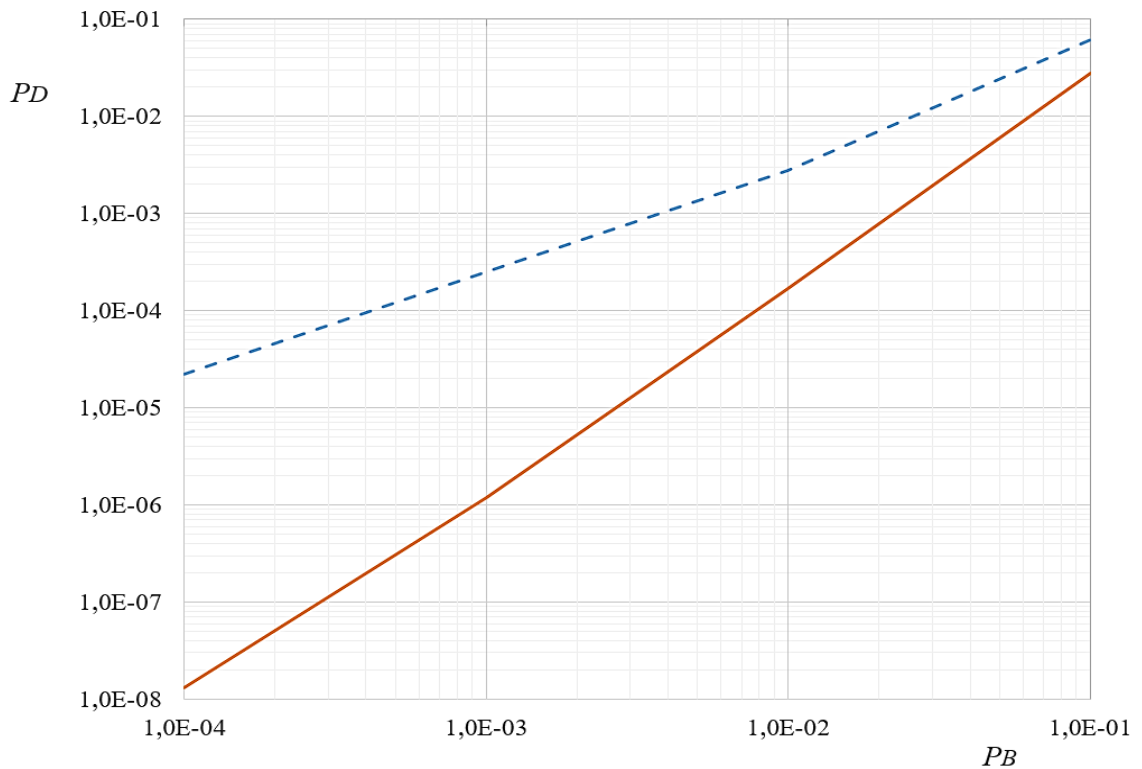


Рисунок 4.26. Оценка корректирующей способности КПИБ на основе ССК для случая декодирования жесткого решения детектора при использовании коэффициентов

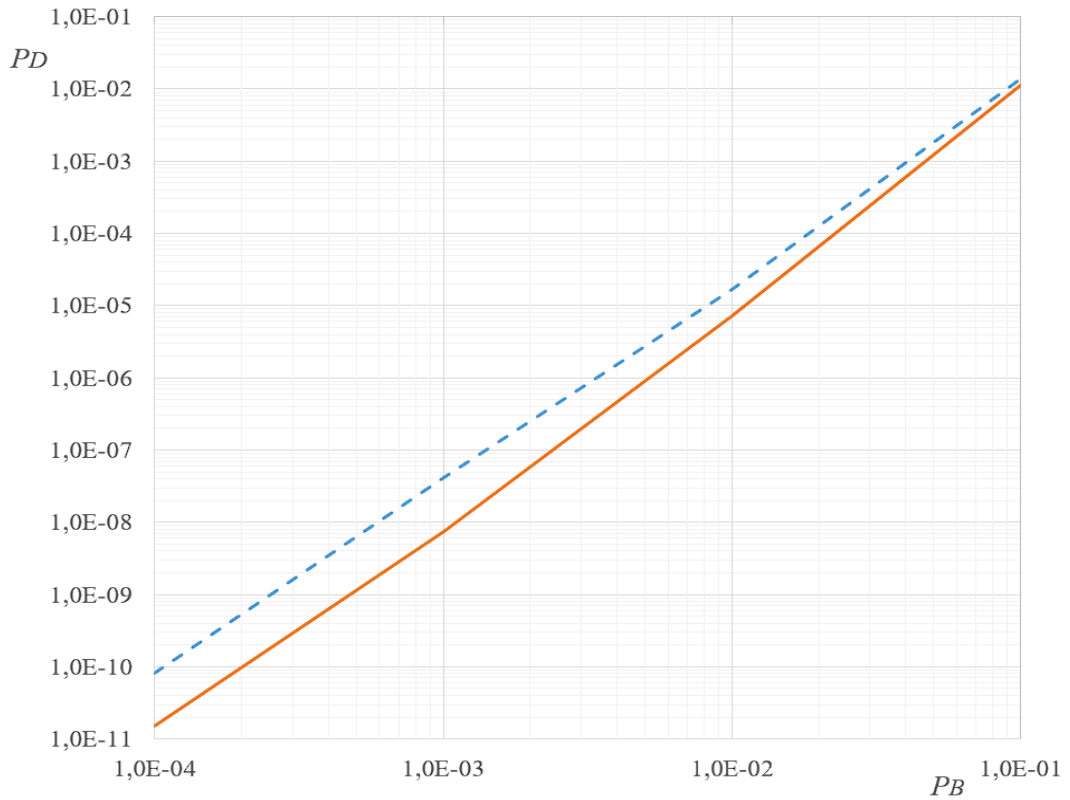


Рисунок 4.27. Оценка корректирующей способности КПИБ на основе СБК для случая декодирования мягкого решения детектора при использовании коэффициентов

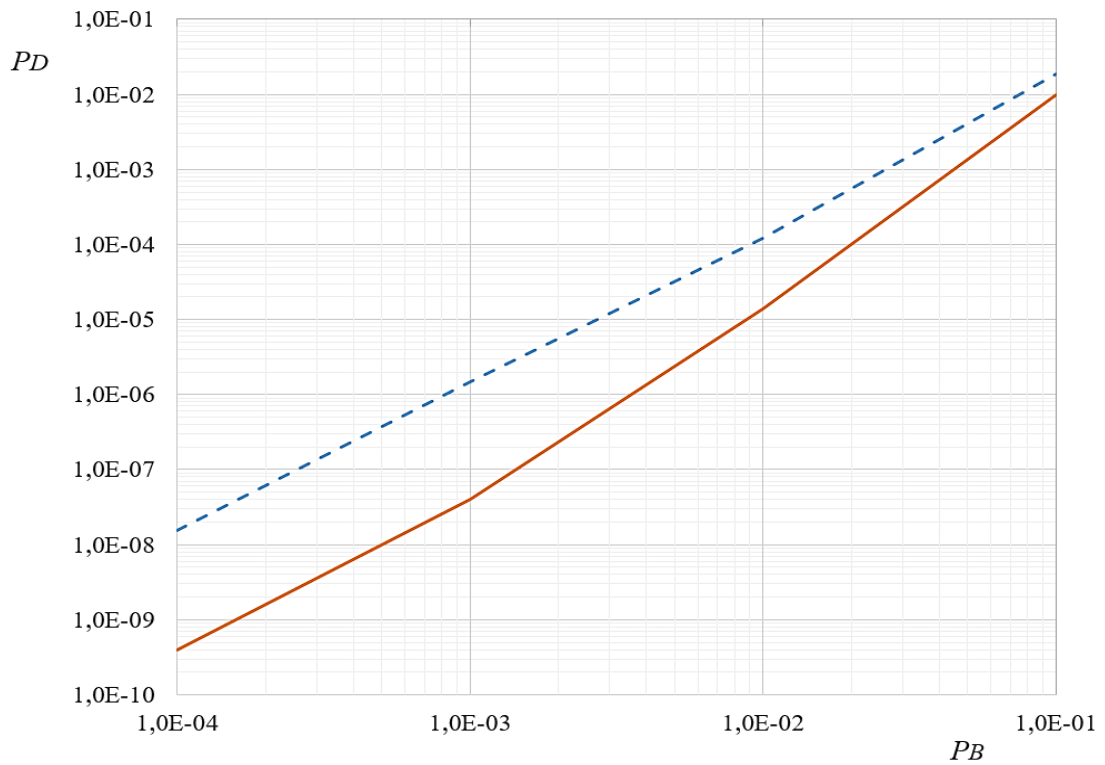


Рисунок 4.28. Оценка корректирующей способности для КПИБ на основе ССК для случая декодирования мягкого решения детектора при использовании коэффициентов

В параграфе рассмотрена реализация повышения корректирующей способности КПИБ путем снижения чувствительности декодера к значениям проверочных бит кодового слова. Путем анализа особенностей декодирования кодового слова, а также экспериментальным путем, выявлена высокая чувствительность декодера к достоверности значений проверочных бит кодового слова КПИБ. Негативной стороной такой чувствительности является высокая вероятность ошибки декодирования при наличии ошибок в проверочных битах кодового слова. Рассмотрен модифицированный метод декодирования по максимуму апостериорной вероятности при применении понижающих коэффициентов для декодируемых значений проверочных бит кодового слова КПИБ. Применение коэффициентов позволило снизить вероятность битовой ошибки в декодированном информационном сообщении. Например, для случая декодирования КПИБ на основе ССК, снижение составило 208 раз – с  $2,5 \cdot 10^{-4}$  до  $1,2 \cdot 10^{-6}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ).

#### **Выводы по главе 4**

В главе описан процесс декодирования турбокодов с использованием распространенных алгоритмов SOVA и MAP. Подробно рассмотрены особенности декодирования перфорированного турбокода. Произведена оценка корректирующей способности турбокодов при введении перфорации в разные части кодового слова турбокода. Введение перфорации позволяет плавно изменять корректирующую способность турбокода и кодовую скорость. При декодировании жесткого решения детектора изменение кодовой скорости возможно от 0,33 до 0,57, то есть на 72 %. При декодировании мягкого решения детектора изменение кодовой скорости возможно от 0,33 до 0,8, то есть на 142 %.

Установлено, что возникновение ошибок в части кодового слова, содержащей информационные биты, с наибольшей вероятностью приводит к ошибочному



декодированию. Сделан вывод о большей чувствительности декодера к значениям информационных бит кодового слова турбокода.

Реализация предложенного варианта построения комбинированного кода на основе сверточного или блочного кода при повторной передаче информационных бит и декодировании, согласно максимуму апостериорной вероятности, позволила снизить вероятность битовой ошибки в декодированном информационном сообщении. Для случая построения КПИБ на основе сверточного кода при повторной передаче информационных бит кодового слова, вероятность битовой ошибки в декодированном информационном сообщении снизилась в 27 раз с  $4,0 \cdot 10^{-5}$  до  $1,5 \cdot 10^{-6}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ).

В главе изучена возможность повышения корректирующей способности КПИБ путем снижения чувствительности декодера к значениям проверочных бит кодового слова. Путем анализа особенностей декодирования кодового слова, а также экспериментальным путем, выявлена высокая чувствительность декодера к достоверности значений проверочных бит кодового слова КПИБ. Негативной стороной такой чувствительности является высокая вероятность ошибки декодирования при наличии ошибок в проверочных битах кодового слова. Рассмотрены изменения метода декодирования по максимуму апостериорной вероятности при применении понижающих коэффициентов для декодируемых значений проверочных бит кодового слова КПИБ. Применение коэффициентов позволило снизить вероятность битовой ошибки в декодированном информационном сообщении. Например, для случая декодирования КПИБ на основе ССК, снижение составило 208 раз с  $2,5 \cdot 10^{-4}$  до  $1,2 \cdot 10^{-6}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ ).

## Глава 5. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ТУРБОКОДОВ

Турбокоды имеют наилучшую корректирующую способность в каналах с высокой вероятностью появления ошибочного бита, однако и они имеют потенциал для дальнейшего развития. Данная глава посвящена вопросам повышения эффективности турбокодов. В начале главы рассматриваются принципы построения турбокода на основе СБК при декодировании в соответствии с критерием максимума апостериорной вероятности с использованием рассмотренной в п. 3.6 синдромной решеткой.

Проведенные в п. 4.2 исследования корректирующей способности турбокода при расположении ошибок в разных частях кодового слова показали более высокую чувствительность декодера турбокода к достоверности значений информационных бит. Отрицательной стороной такой чувствительности является высокая вероятность ошибки декодирования кодового слова при наличии ошибок в информационных битах. В главе предлагается два направления борьбы с указанным недостатком. Первое направление – снижение чувствительности информационных бит кодового слова турбокода к ошибкам, второе – введение дополнительного избирательного кодирования информационных бит.

### **5.1. Построение ТСБК при декодировании в соответствии с критерием максимума апостериорной вероятности**

Построение турбокода возможно как на основе систематического сверточного кода, так и на основе систематического блочного кода. Для варианта построения ТССК разработан и хорошо исследован целый ряд алгоритмов декодирования (SOVA, MAP, Log-MAP, Max-Log-MAP и др.) [10, 11, 12]. Для варианта построения турбокода на основе СБК (далее – ТСБК) в разнообразных источниках описано построение турбокода в виде кода-произведения при декодировании, как правило, согласно алгоритму Чейза-Пиндиаха [189, 190, 191, 192]. Алгоритмы

декодирования кодов-произведений значительно усложняются при декодировании мягкого решения детектора [190, 193] в отличие от перечисленных алгоритмов декодирования турбокодов на основе ССК. Алгоритмы декодирования турбокодов на основе ССК имеют более гибкие аппаратные реализации и показывают лучшие характеристики при низких относительных скоростях кодирования. Можно сделать предположение, что использование при декодировании СБК синдромной решетки позволит строить ТСБК с применением при декодировании перечисленных выше алгоритмов, используемых при построении ТССК [193, 194, 195].

В данном параграфе будет рассмотрен модифицированный метод декодирования в соответствии с критерием максимума апостериорной вероятности, позволяющий осуществлять декодирование кодовой последовательности ТСБК. Для примера используем простой турбокод, кодер которого состоит из двух кодеров блочного кода и устройства перемежения (см. рис. 5.1). В качестве СБК выберем код Хемминга, кодовое слово которого состоит из трех проверочных и четырех информационных бит. Декодирование будет произведено в соответствии с принципом максимума апостериорной вероятности информационного бита. Процесс декодирования ТССК в соответствии с алгоритмом MAP подробно изложен в п. 4.2.

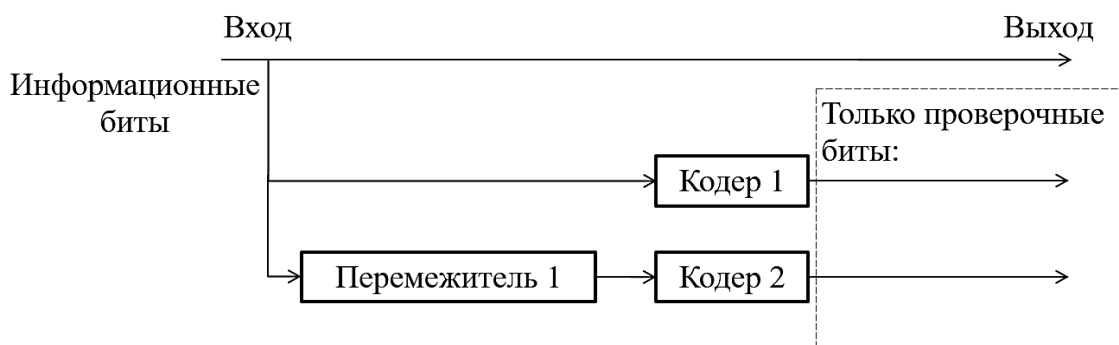


Рисунок 5.1. Структурная схема кодера турбокода

В соответствии с рисунком 5.1 кодовое слово турбокода состоит из информационных бит, проверочных бит с выхода первого кодера и проверочных бит с выхода второго кодера. Структура кодового слова турбокода, построенного на основе кода Хемминга, изображена на рисунке 5.2. Синдромная решетка может

быть использована для кодирования и декодирования. Решетка для выбранного кода Хемминга изображена на рисунке 3.14. Процесс кодирования и декодирования кода Хемминга с применением синдромной решетки подробно рассмотрен в п. 3.6.

Опишем процесс создания кодовой последовательности турбокода, построенного на основе СБК. Пусть 0100 – информационная последовательность. В процессе кодирования она поступает на вход первого кодера Хемминга без перемежения. В результате кодирования, процесс которого показан на рисунке 5.3, формируются проверочные биты 011. На вход второго кодера Хемминга поступает информационная последовательность после перемежения. Пусть при перемежении первый по счету бит информационной последовательности меняется с третьим, а второй с четвертым, тогда на вход второго кодера Хемминга поступает последовательность 0001, а на выходе формируются проверочные биты 101. Тогда выход декодера турбокода, состоящий из блока информационных бит, блока проверочных бит с выхода первого кодера и блока проверочных бит с выхода второго кодера имеет вид: 0100 011 101. При передаче биполярными сигналами кодовое слово ТСБК имеет вид  $-11-1-1 -111 1-11$ .

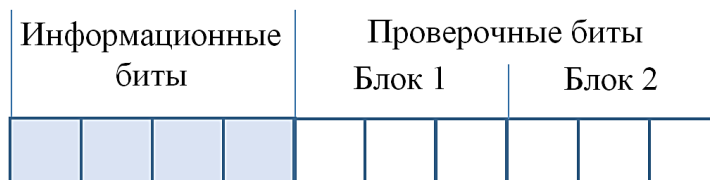


Рисунок 5.2. Структура кодового слова турбокода на основе кода Хемминга

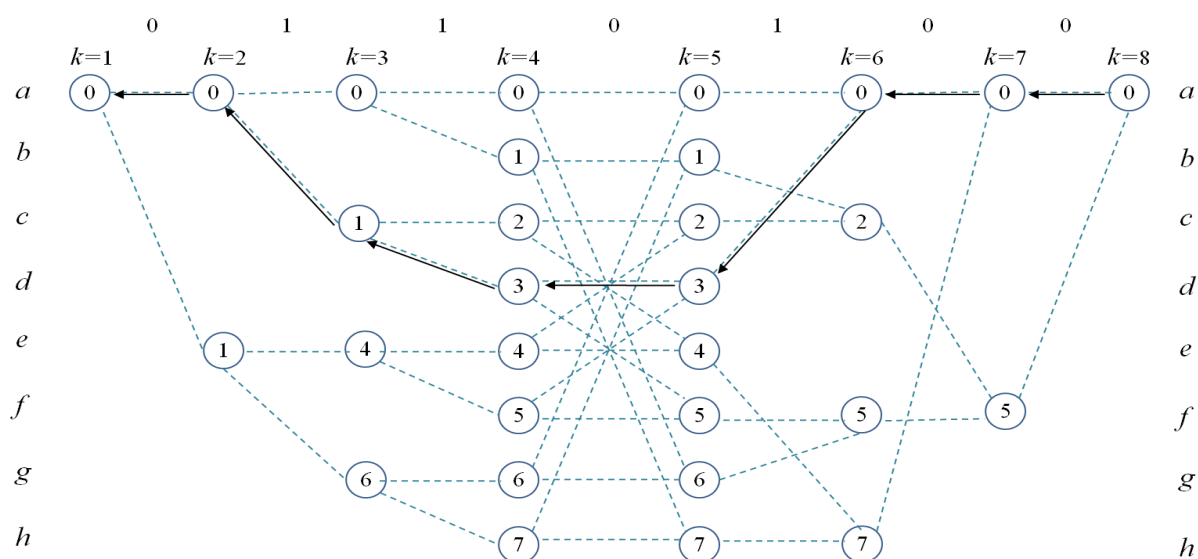


Рисунок 5.3. Кодирование по синдромной решетке кода Хемминга

Рассмотрим процесс декодирования описанного кодового слова турбокода в соответствии с алгоритмом MAP. В качестве декодера будем использовать декодер турбокода, структурная схема которого изображена на рисунке 4.13. Последовательность  $w_k$ , поступающая на вход первого декодера, состоит из детектированных значений проверочных бит первого кодера  $y_1$  и детектированных значений информационных бит  $x$ . Последовательность, поступающая на вход второго декодера, состоит из детектированных значений проверочных бит второго кодера  $y_2$ , детектированных и расперемеженных значений информационных бит  $x$ . Процесс декодирования по синдромной решетке итеративен. На вход первого декодера в каждый момент времени  $k$  (кроме  $k = 8$ ) поступает очередное детектированное значение принятого кодового бита  $w_k$ .

Произведем модификацию алгоритма декодирования MAP с целью реализации процесса декодирования с использованием нерегулярной решетки блочного кода. Процесс декодирования ТССК согласно алгоритму MAP подробно рассмотрен в п. 4.2. и математически описан с применением формул (4.4) – (4.9). Рассмотрим вносимые в процесс декодирования изменения и изменения в указанные формулы с целью их применения для декодирования турбокода на основе кода Хемминга. Рассмотрим процесс декодирования принятого кодового слова, начав с работы первого составного декодера. Произведем расчет метрик ветвей. С учетом того, что ветвь решетки блочного кода (в отличие от ветви решетки сверточного кода) соответствует всего одному кодовому биту, формула для вычисления метрик ветвей имеет следующий вид:

$$\delta_k^{i,m} = 0,5 \cdot \exp(w_k \cdot v_k^{i,m}), \quad (5.1)$$

где  $m$  – параметр, принимающий значение  $a, b, \dots, h$  и обозначающий принадлежность узла решетки одному из восьми состояний, параметр  $i$  принимает значение 0 или 1 в зависимости от того, генерации какого бита соответствует данная ветвь (для ветвей идущих между узлами с одинаковыми состояниями  $i = 0$ , для других ветвей идущих вверх или вниз  $i = 1$ ),  $v_k$  – значения бита, генерируемого данной ветвью. После расчета значений метрик ветвей производится расчет метрик

прямого прохода по решетке декодера. Формула для вычисления метрик прямого прохода имеет следующий вид:

$$\alpha_{k+1}^m = \sum_{j=0}^1 \delta_k^{j,b(j,m)} \cdot \alpha_k^{b(j,m)}, \quad (5.2)$$

где функция  $b(j, m)$  определяет предыдущее состояние в зависимости от параметра  $j$  и текущего состояния  $m$ . Суть формулы (5.2) в том, что метрика состояния  $m$  в момент времени  $k + 1$  является суммой произведения значений метрик состояний в момент  $k$  и метрик ветвей, ведущих в данное состояние. Начальное состояние пути по решетке (при  $k = 1$ ) нам известно:  $m = a$ , тогда при  $k = 1$  и  $m = a$  метрика прямого прохода равна единице, а все остальные метрики прямого прохода (для  $m = b \dots h$ ) равны нулю. Вычисляются все метрики прямого прохода для  $k = 2 \dots 7$ . Далее производится расчет метрик обратного прохода по решетке декодера из состояния  $k = 8$  в состояние  $k = 1$ . Формула для вычисления метрик обратного прохода имеет следующий вид:

$$\beta_k^m = \sum_{j=0}^1 \delta_k^{j,m} \cdot \beta_{k+1}^{f(j,m)}, \quad (5.3)$$

где функция  $f(j, m)$  определяет следующее состояние в зависимости от параметра  $j$  и текущего состояния  $m$ . Суть формулы (5.3) в том, что метрика состояния  $m$  в момент времени  $k$  является суммой произведения значений метрик состояний в момент  $k + 1$  и метрик ветвей, ведущих в данное состояние. Финальное состояние решетки известно, тогда при  $k = 8$  и  $m = a$  метрика обратного прохода равна единице, а все остальные метрики обратного прохода (для  $m = b \dots h$ ) равны нулю. Далее вычисляются метрики обратного прохода по формуле (5.3) для  $k = 7, 6, 5, 4, 3, 2$ .

Вычислив значения всех трех видов метрик, производим расчет логарифмической функции правдоподобия  $k$ -го бита по следующей формуле:

$$L_e(d_k) = \ln \left( \frac{\sum_m \alpha_k^m \delta_k^{1,m} \beta_{k+1}^{f(1,m)}}{\sum_m \alpha_k^m \delta_k^{0,m} \beta_{k+1}^{f(0,m)}} \right), \quad (5.4)$$

где  $d_k$  – значение символа.

На вход второго декодера поступают детектированные и перемеженные значения информационных бит и детектированные значения проверочных бит второго кодера  $u_{k2}$ . Для второго декодера проводятся аналогичные вычисления по

формулам (5.1) – (5.4). Полученные в итоге значения логарифмической функции правдоподобия снова обозначаются как  $L_e(d_k)$ , а вычисленные ранее значения логарифмической функции правдоподобия (на выходе первого декодера) теперь обозначаются как  $L'_e(d_k)$ . После расчета логарифмической функции правдоподобия для всех информационных бит на выходе второго декодера и перемежения результатов расчета первая полная итерация декодирования считается завершенной. Далее вычисляются значения мягкого выхода декодера турбокода:

$$L(d_k) = L_c(d_k) + L'_e(d_k) + L_e(d_k), \quad (5.5)$$

где  $L_c(d_k)$  – канальное измерение значения символа (в нашем случае выход детектора),  $L'_e(d_k)$  – априорное значение логарифмической функции правдоподобия (в нашем случае мягкий выход первого декодера),  $L_e(d_k)$  – значение логарифмической функции правдоподобия по результатам декодирования (в данном случае мягкий выход второго декодера). Для достижения необходимой надежности решения может быть произведено несколько итераций, после чего принимается жесткое решение о значении символа в соответствии с критерием (4.9).

Рассмотрим процесс декодирования в виде числового расчета по формулам (5.1) – (5.5) для конкретного кодового слова. Пусть передано ранее сформированное кодовое слово  $-11-1-1-1111-11$ . Допустим, что поступившая на вход декодера последовательность содержит три ошибки и имеет вид:  $0,2$   $0,8$   $-0,4$   $-0,9$   $-0,6$   $-0,1$   $0,5$   $0,4$   $0,1$   $0,5$ , где ошибочные символы подчеркнуты. Исходя из структуры кодового слова турбокода, вход первого декодера имеет вид:  $-0,6$   $-0,1$   $0,5$   $0,2$   $0,8$   $-0,4$   $-0,9$ . Вход второго декодера состоит из последних трех символов и перемеженных информационных символов:  $0,4$   $0,1$   $0,5$   $-0,4$   $-0,9$   $0,2$   $0,8$ .

Ниже рассмотрим расчет выхода первого декодера по описанному ранее алгоритму.

Для последовательности  $-0,6$   $-0,1$   $0,5$   $0,2$   $0,8$   $-0,4$   $-0,9$  произведем расчет метрик ветвей по формуле (5.1). Для упрощения здесь и далее из формулы (5.1) можно убрать коэффициент  $0,5$ , так как с учетом формулы (5.4) одинаковые

множители числителя и знаменателя сократятся. В момент времени  $k = 1$  имеются только две ветви, исходящие из состояния  $a$ . Рассчитаем метрики ветвей, идущих от состояния  $k = 1$  в  $k = 2$ .

Для ветви, идущей в то же состояние ( $i = 0$ ):

$$\delta_1^{0,m=a} = \exp(w_1 \cdot v_1^{0,m=a}) = \exp(-0,6 \cdot (-1)) = 1,82.$$

Для ветви, идущей в другое состояние ( $i=1$ ):

$$\delta_1^{1,m=a} = \exp(w_1 \cdot v_1^{1,m=a}) = \exp(-0,6 \cdot 1) = 0,55.$$

В момент времени  $k = 2$  имеются уже четыре ветви, исходящие из состояний  $a$  и  $e$ . Найдем метрики ветвей, идущих от состояния  $a$ .

Для ветви, идущей в то же состояние ( $i = 0$ ):

$$\delta_2^{0,m=a} = \exp(w_2 \cdot v_2^{0,m=a}) = \exp(-0,1 \cdot (-1)) = 1,11.$$

Для ветви, идущей в другое состояние ( $i=1$ ):

$$\delta_2^{1,m=a} = \exp(w_2 \cdot v_2^{1,m=a}) = \exp(-0,1 \cdot 1) = 0,9.$$

Найдем метрики ветвей, идущих от состояния  $e$ .

Для ветви, идущей в то же состояние ( $i=0$ ):

$$\delta_2^{0,m=e} = \exp(w_2 \cdot v_2^{0,m=e}) = \exp(-0,1 \cdot (-1)) = 1,11.$$

Для ветви, идущей в другое состояние ( $i=1$ ):

$$\delta_2^{1,m=e} = \exp(w_2 \cdot v_2^{1,m=e}) = \exp(-0,1 \cdot 1) = 0,9.$$

Так как значение значения бита, генерируемого ветвью, не зависит от  $m$ , то формулу (5.1) можно упростить:

$$\delta_k^i = \exp(w_k \cdot v_k^i). \quad (5.6)$$

С учетом формулы (5.6) для каждого  $k$  будет вычисляться всего две метрики ветвей: метрика ветвей, идущих в то же состояние ( $i = 0$ ), и метрика ветвей, идущих в другое состояние ( $i = 1$ ). Результаты расчета метрик всех ветвей решетки приведены в таблице 5.1.

Рассчитаем метрики прямого прохода. При  $k = 1$  для состояния  $m = a$  метрика прямого прохода равна 1. Рассчитаем метрики прямого прохода при  $k = 2$  по формуле (5.2). При  $k = 2$  возможными являются только состояния  $a$  и  $e$ . Метрика прямого прохода для состояния  $m = a$  определяется метриками входящих в это



состояние ветвей, перемноженных на значения метрик прямого прохода для узлов, из которых эти ветви исходят. С учетом того, что в состояние  $m = a$  входит только ветвь из того же состояния, метрика прямого прохода равна:

$$\alpha_2^{m=a} = \delta_1^0 \cdot \alpha_1^{m=a} = 1,82 \cdot 1 = 1,82.$$

С учетом того, что в состояние  $m = e$  входит только ветвь из состояний  $a$  ( $i=1$ ), метрика прямого прохода равна:

$$\alpha_2^{m=e} = \delta_1^1 \cdot \alpha_1^{m=a} = 0,55 \cdot 1 = 0,55.$$

Аналогично рассчитываются метрики прямого прохода для  $k = 3 \dots 7$ . Результаты расчета метрик прямого прохода для всей решетки приведены в таблице 5.2.

Таблица 5.1

Значения метрик ветвей

$k$	$i$	$m=a$	$m=b$	$m=c$	$m=d$	$m=e$	$m=f$	$m=g$	$m=h$
1	0	1,82	-	-	-	-	-	-	-
	1	0,55	-	-	-	-	-	-	-
2	0	1,11	-	-	-	1,11	-	-	-
	1	0,90	-	-	-	0,90	-	-	-
3	0	0,61	-	0,61	-	0,61	-	0,61	-
	1	1,65	-	1,65	-	1,65	-	1,65	-
4	0	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
	1	1,22	1,22	1,22	1,22	1,22	1,22	1,22	1,22
5	0	0,45	-	0,45	-	-	0,45	-	0,45
	1	-	2,23	-	2,23	2,23	-	2,23	-
6	0	1,49	-	-	-	-	1,49	-	-
	1	-	-	0,67	-	-	-	-	0,67
7	0	2,46	-	-	-	-	-	-	-
	1	-	-	-	-	-	0,41	-	-

Значения метрик прямого прохода

$k$	$m=a$	$m=b$	$m=c$	$m=d$	$m=e$	$m=f$	$m=g$	$m=h$
1	1	-	-	-	-	-	-	-
2	1,82	-	-	-	0,55	-	-	-
3	2,01	-	1,65	-	0,61	-	0,5	-
4	1,22	3,32	1	2,72	0,37	1	0,3	0,82
5	1,37	3,72	1,27	3,45	1,52	4,14	1,74	4,73
6	8,29	-	8,84	-	-	5,73	-	5,51
7	16,06	-	-	-	-	14,48	-	-

Рассчитаем метрики обратного прохода. Как говорилось ранее, при  $k = 8$  для состояния  $m = a$  метрика обратного прохода равна 1. Рассчитаем метрики обратного прохода при  $k = 7$  по формуле (5.3). При  $k = 7$  возможными являются только состояния  $a$  и  $f$ . Метрика обратного прохода для состояния  $m = a$  определяется метриками входящих в это состояние при обратном проходе ветвей, перемноженных на значения метрик обратного прохода для узлов, из которых эти ветви исходят. С учетом того, что при обратном проходе справа налево в состояние  $m = a$  входит ветвь из того же состояния  $a$  ( $i = 0$ ), метрика обратного прохода равна:

$$\beta_7^{m=a} = \delta_7^0 \cdot \beta_8^{m=a} = 2,46 \cdot 1 = 2,46.$$

С учетом того, что при обратном проходе справа налево в состояние  $f$  входит только ветвь из состояний  $a$  ( $i = 1$ ), метрика обратного прохода равна:

$$\beta_7^{m=f} = \delta_7^1 \cdot \beta_8^{m=a} = 0,41 \cdot 1 = 0,41.$$

Аналогично рассчитываются метрики обратного прохода для  $k = 6 \dots 2$ . Результаты расчета метрик обратного прохода для всей решетки приведены в таблице 5.3.

Произведем расчет логарифмической функции правдоподобия для всех информационных бит. При  $k = 1$  имеется всего одна ветвь, соответствующая приходу единицы, и одна ветвь, соответствующая приходу нуля. Тогда согласно формуле (5.4):

$$L_e(d_1) = \ln(1 \cdot 0,55 \cdot 24,42 / 1 \cdot 1,82 \cdot 17,55) = -0,87.$$

Аналогично рассчитываются значения логарифмической функции правдоподобия для  $k = 2 \dots 7$ . Результаты расчета приведены в таблице 5.4.

Таблица 5.3

## Значения метрик обратного прохода

$k$	$m=a$	$m=b$	$m=c$	$m=d$	$m=e$	$m=f$	$m=g$	$m=h$
8	1	0	0	0	0	0	0	0
7	2,46	0	0	0	0	0,41	0	0
6	3,67	0	0,27	0	0	0,61	0	1,65
5	1,64	0,61	0,12	8,2	3,7	0,27	1,35	0,74
4	2,99	1,4	4,58	7,01	3,15	10,19	3,12	1,35
3	4,13	0	14,35	0	18,73	0	4,11	0
2	17,55	0	0	0	24,42	0	0	0

Таблица 5.4

## Мягкий выход первого декодера

$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
-0,87	0,27	1,22	-0,09	1,69	-1,08	-1,9

Мягкий выход второго декодера рассчитывается аналогично с учетом вышесказанных замечаний. Результаты расчета приведены в таблице 5.5.

Таблица 5.5

## Мягкий выход второго декодера

$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
0,87	-0,23	0,98	-0,93	-1,78	-0,15	1,34

После расчета мягких выходов первого и второго декодеров первая итерация декодирования турбокода считается завершенной. Теперь имеются все необходимые данные для расчета по формуле (5.5) мягкого выхода декодера турбокода. Значения информационных бит соответствуют  $k = 4 \dots 7$ . Жесткий выход декодера турбокода, рассчитанный в соответствии с критерием (4.9), для всех информационных бит не содержит ошибок. Результаты расчета мягкого и

жесткого выхода декодера турбокода приведены в таблице 5.6. Исправление трех ошибочных бит из десяти кодовых после первой итерации демонстрирует успешную работу декодера.

Таблица 5.6

Результаты расчета мягкого выхода декодера турбокода.

номер информационного бита	выход детектора	выход первого декодера	выход второго декодера	мягкий выход декодера турбокода	жесткий выход декодера турбокода
1	0,2	-0,09	-0,15	-0,04	0
2	0,8	1,69	1,34	3,83	1
3	-0,4	-1,08	-0,93	-2,41	0
4	-0,9	-1,9	-1,78	-4,58	0

В параграфе рассмотрено построение модифицированного метода декодирования в соответствии с критерием максимума апостериорной вероятности, позволяющего осуществлять декодирование кодовой последовательности ТСБК. Такой вариант построения декодера ТСБК позволяет без усложнения алгоритма осуществлять декодирование мягкого решения детектора и декодирование перфорированного кода, имеет более гибкие аппаратные реализации. Описана модификация алгоритма MAP для реализации декодирования с использованием нерегулярной синдромной решетки. Далее произведем количественные оценки корректирующей способности ТСБК.

## 5.2. Сравнение корректирующей способности турбокодов, созданных на основе сверточных и блочных кодов при декодировании в соответствии с критерием максимума апостериорной вероятности

### 5.2.1. Сравнение результатов декодирования кодовых слов турбокода, созданного на основе сверточного и блочного кода

Произведем сравнение корректирующей способности ТССК и ТСБК, рассмотрев процесс декодирования отдельных кодовых слов. Формирование из информационной последовательности 0100, кодового слова ТСБК 0100 011 101, рассмотрено в п. 5.1 [194, 195].

Процесс создания кодовой последовательности ТССК подробно описан в п. 4.2. Для информационной последовательности 0100 выход декодера турбокода, состоящий из блока информационных бит, блока проверочных бит с выхода первого кодера и блока проверочных бит с выхода второго кодера имеет вид: 0100 0111 0001.

Полученные кодовые слова двух видов представлены в таблице 5.7. Для ТССК полученная кодовая последовательность состоит из 12 бит, а для ТСБК, созданного на основе кода Хемминга, – из 10 бит. Чтобы уравнивать длины последовательностей, произведем выкалывание двух проверочных бит из кодовой последовательности ТССК. Передачу будем осуществлять биполярными сигналами. С учетом выкалывания двух бит в кодовом слове ТССК кодовые слова представлены в таблице 5.8.

Таблица 5.7

Кодовое слово турбокода

Код	Информационные биты				Проверочные биты первого кодера				Проверочные биты второго кодера			
	СБК	0	1	0	0	0	1		1	1		0
ССК	0	1	0	0	0	1	1	1	0	0	0	1

Кодовое слово турбокода на входе модулятора

Код	Информационные биты				Проверочные биты первого кодера				Проверочные биты второго кодера			
	СБК	-1	1	-1	-1	-1	1		1	1		-1
ССК	-1	1	-1	-1	-1	1	X	1	-1	X	-1	1

Сравним результаты декодирования кодовых слов ТСБК и ТССК для четырех случаев нахождения ошибок в кодовом слове:

- имеется по одной ошибке во всех трех частях кодового слова;
- имеются три ошибки в проверочных битах первого кодера;
- имеются три ошибки в проверочных битах второго кодера;
- имеются три ошибки в информационных битах.

Обозначим значения детектированных без ошибки бит на выходе мягкого детектора как  $\pm 0,5$ . Выколотым битам присвоим значение 0.

Кодовые слова на входе декодера турбокода для случая наличия по одной ошибке в каждой части кодового слова приведены в таблице 5.9. Ошибочные биты подчеркнуты. Результаты вычисления мягкого решения декодера о значениях информационных бит и принятия жесткого решения приведены в таблице 5.10. По итогам первой итерации выходы обоих декодеров содержат ошибки. По итогам второй итерации ошибка имеется на выходе декодера СБК. Выходы обоих декодеров по итогам трех итераций не содержат ошибок. У всех декодированных информационных бит их абсолютные значения выше для декодера турбокода на основе ССК, что показывает их большую достоверность.

Таблица 5.9

Вход декодера турбокода

Вид турбокода	Информационные биты				Проверочные биты первого кодера				Проверочные биты второго кодера			
	ТСБК	<u>0,2</u>	0,5	-0,5	-0,5	-0,5	<u>-0,3</u>		0,5	0,5		<u>0,3</u>
ТССК	<u>0,2</u>	0,5	-0,5	-0,5	-0,5	<u>-0,3</u>	0	0,5	-0,5	0	<u>0,3</u>	0,5

Выход декодера турбокода

Выход декодера	№ итерации	Значения декодированных информационных бит			
		Мягкий выход декодера ТСБК	1	<u>0,7</u>	2,2
Мягкий выход декодера ТССК	1	<u>0,3</u>	2,2	-3,6	-2,7
Мягкий выход декодера ТСБК	2	<u>0,2</u>	3,5	-4,9	-4,9
Мягкий выход декодера ТССК	2	-0,2	4,6	-7,8	-5,4
Мягкий выход декодера ТСБК	3	-1,4	6,5	-10,4	-10,2
Мягкий выход декодера ТССК	3	-1,5	8,7	-16,9	-11,2
Жесткий выход декодера ТСБК	3	0	1	0	0
Жесткий выход декодера ТССК	3	0	1	0	0

Кодовые слова на входе декодера турбокода для случая наличия трех ошибок в проверочных битах первого кодера приведены в таблице 5.11. Ошибочные биты подчеркнуты. Результаты вычисления мягкого решения декодера о значениях информационных бит и принятия жесткого решения приведены в таблице 5.12. Выходы обоих декодеров не содержат ошибок. Абсолютные значения выхода для трех бит из четырех выше для декодера на базе СБК.

Кодовые слова на входе декодера турбокода для случая наличия трех ошибок в проверочных битах второго кодера приведены в таблице 5.13. Ошибочные биты подчеркнуты. Результаты вычисления мягкого решения декодера о значениях информационных бит и жесткого решения приведены в таблице 5.14. Абсолютные значения выхода для трех бит из четырех выше для декодера на базе СБК.

Таблица 5.11

Вход декодера турбокода

Вид турбокода	Информационные биты				Проверочные биты первого кодера				Проверочные биты второго кодера			
	ТСБК	-0,5	0,5	-0,5	-0,5	<u>0,6</u>	<u>-0,6</u>		<u>-0,6</u>	0,5		-0,5
ТССК	-0,5	0,5	-0,5	-0,5	<u>0,6</u>	<u>-0,6</u>	0	<u>-0,6</u>	-0,5	0	-0,5	0,5

Таблица 5.12

## Выход декодера турбокода

Выход	№ итерации	Значения декодированных информационных бит			
		Мягкий выход декодера ТСБК	1	-3,3	3,3
Мягкий выход декодера ТССК	1	-1,4	2,9	-3,6	-2,7
Мягкий выход декодера ТСБК	2	-5,5	5,7	-2,8	-5,7
Мягкий выход декодера ТССК	2	-1,1	3,5	-6,9	-3,4
Мягкий выход декодера ТСБК	3	-10,7	11,0	-3,5	-11,0
Мягкий выход декодера ТССК	3	-1,1	7,4	-14,5	-6,3
Жесткий выход декодера ТСБК	3	0	1	0	0
Жесткий выход декодера ТССК	3	0	1	0	0

Таблица 5.13

## Вход декодера турбокода

Вид турбокода	Информационные биты				Проверочные биты первого кодера				Проверочные биты второго кодера			
	ТСБК	-0,5	0,5	-0,5	-0,5	-0,5	0,5		0,5	<u>-0,6</u>		<u>0,6</u>
ТССК	-0,5	0,5	-0,5	-0,5	-0,5	0,5	0	0,5	<u>0,1</u>	0	<u>0,1</u>	<u>-0,1</u>

Таблица 5.14

## Выход декодера турбокода

Выход	№ итерации	Значения декодированных информационных бит			
		Мягкий выход декодера ТСБК	1	-2,3	3,3
Мягкий выход декодера ТССК	1	-3,8	3,1	-1,6	-2,6
Мягкий выход декодера ТСБК	2	-2,8	5,7	-5,5	-5,7
Мягкий выход декодера ТССК	2	-7,2	5,7	-1,4	-2,8
Мягкий выход декодера ТСБК	3	-3,5	11,1	-10,7	-11
Мягкий выход декодера ТССК	3	-15,3	11,0	-1,0	-5,0
Жесткий выход декодера ТСБК	3	0	1	0	0
Жесткий выход декодера ТССК	3	0	1	0	0



Кодовые слова на входе декодера турбокода для случая наличия трех ошибок в информационных битах кодового слова приведены в таблице 5.15. Ошибочные биты подчеркнуты. Результаты вычисления мягкого решения декодера о значениях информационных бит и жесткого решения приведены в таблице 5.16.

Таблица 5.15

## Вход декодера турбокода

Вид турбокода	Информационные биты				Проверочные биты первого кодера				Проверочные биты второго кодера			
	ТСБК	<u>0,1</u>	<u>-0,1</u>	-0,5	<u>0,1</u>	-0,5	0,5		0,5	0,5		-0,5
ТССК	<u>0,1</u>	<u>-0,1</u>	-0,5	<u>0,1</u>	-0,5	0,5	0	0,5	-0,5	0	-0,5	0,5

Таблица 5.16

## Выход декодера турбокода

Выход декодера	№ итерации	Значения декодированных информационных бит			
		Мягкий выход декодера ТСБК	1	<u>0,5</u>	<u>-0,5</u>
Мягкий выход декодера ТССК	1	-0,3	0,1	-3,5	<u>0,5</u>
Мягкий выход декодера ТСБК	2	<u>1,4</u>	<u>-1,1</u>	-4,9	<u>1,1</u>
Мягкий выход декодера ТССК	2	-1,6	0,7	-7,3	<u>0,2</u>
Мягкий выход декодера ТСБК	3	<u>4,3</u>	<u>-3,5</u>	-10,5	<u>3,5</u>
Мягкий выход декодера ТССК	3	-4,5	2,9	-15,3	-1,0
Жесткий выход декодера ТСБК	3	0	1	0	0
Жесткий выход декодера ТССК	3	<u>1</u>	<u>0</u>	0	<u>1</u>

Подводя общий итог результатов декодирования обоими декодерами принятых кодовых слов при четырех вариантах распределения ошибок, можно судить о значительном преимуществе варианта построения турбокода на основе ССК. Еще одним преимуществом турбокода на основе ССК является то, что решетка ССК (в отличие от решетки СБК) имеет регулярную структуру, что требует меньших объемов памяти для хранения информации о ее структуре. Следует отметить, что декодер кода Хемминга, работающий с жестким решением

детектора, способен исправлять только один ошибочный бит. Создание на основе кода Хемминга турбокода и использование при декодировании мягкого решения детектора значительно увеличивает корректирующую способность кода, позволяя исправлять множественные ошибки. Для этого на передающей стороне достаточно добавить перемежитель. На приемной стороне изменения будут более существенные, заключающиеся в реализации декодера турбокода.

### **5.2.2. Сравнение корректирующей способности турбокодов, созданных на основе сверточного и блочного кода**

Произведем сравнение корректирующей способности турбокодов, созданных на основе сверточного и блочного кода [196, 197]. Результаты исследования корректирующей способности ТССК для случая наличия в канале независимых ошибок путем нахождения вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$  приведены в п. 4.2.

Проведем исследования результатов декодирования кодового слова ТСБК при разном количестве и разнообразном расположении ошибок. На основе анализа результатов исследования осуществим вывод формулы для расчета вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$ . С целью проведения исследований в MS Excel произведена разработка реализации декодера одного кодового слова ТСБК. Реализованный в таком виде декодер при задании пользователем кодового слова выполняет три итерации декодирования жесткого решения детектора. Один ошибочный бит исправляется разработанным декодером при любом расположении. Рассмотрим возможности реализации декодера ТСБК по исправлению двух ошибок при их произвольном расположении. Всего существует четыре случая расположения двух ошибочных бит в кодовом слове турбокода:

- 1 случай – обе ошибки располагаются в информационных битах (в этом случае существует 6 вариантов расположения ошибок);
- 2 случай – одна ошибка располагается в проверочных битах, а вторая в блоке информационных бит (24 варианта расположения ошибок);

- 3 случай – обе ошибки располагаются в одном блоке проверочных бит (6 вариантов расположения ошибок);
- 4 случай – ошибки располагаются в разных блоках проверочных бит (9 вариантов расположения ошибок).

Таблица 5.17

Количество вариантов исправления двух ошибок в кодовом слове

Случаи расположения ошибок				Всего исправлено
1 случай	2 случай	3 случай	4 случай	
0	20	6	9	35

Рассмотрев приведенные в таблице 5.17 результаты декодирования при разном расположении двух ошибок, можно прийти к следующим заключениям. Две ошибки в проверочных битах исправляются при любом их расположении. Если хотя бы один ошибочный бит располагается в информационных битах, то может произойти ошибка декодирования. Ошибка декодирования кодового слова происходит в 10 случаях из 45 возможных, то есть в 22 % случаев. Вероятность появления 2 ошибочных бит из 10 равна:

$$P_{2uz10} = (10! / (2! \cdot (10-2)!)) \cdot p_B^2 \cdot (1-p_B)^{10-2} = 45 \cdot p_B^2 \cdot (1-p_B)^8,$$

где  $p_B$  вероятность появления ошибочного бита в канале передачи данных.

Так как вероятность наличия двух ошибочных бит в кодовом слове значительно превышает вероятность возникновения большего числа ошибок, то с небольшой долей приближения можно считать, что ошибочное декодирование слова вызвано наличием двух ошибок. С учетом того, что при наличии двух ошибок в кодовом слове его ошибочное декодирование осуществляется в 22 % случаев, вероятность ошибочного декодирования кодового слова  $p_W$  вычисляется по следующей формуле:

$$p_W \approx 0,22 \cdot P_{2uz10} = 9,9 \cdot p_B^2 \cdot (1-p_B)^8.$$

Исследования показали, что при ошибочном декодировании кодового слова ошибка возникает в среднем в 2,25 битах из четырех информационных бит. Тогда

вероятность появления ошибочного бита в декодированном информационном сообщении  $p_D$  равна:

$$p_D = (2,25 / 4) \cdot p_W = 5,6 \cdot p_B^2 \cdot (1 - p_B)^8. \quad (5.7)$$

Для оценки корректирующей способности ТСБК при разных значениях  $p_B$  произведен расчет  $p_D$  по формуле (5.7). Также для исследования корректирующей способности ТСБК создан программный имитатор на языке программирования Python [198]. Программный имитатор состоит из кодера ТСБК, блока моделирования выхода детектора приемника с заданной вероятностью появления ошибочного бита и декодера ТСБК. Результаты расчетов по формуле (5.7), а также результаты работы программных имитаторов ТСБК и ТССК [198, 184] для случая декодирования жесткого решения приведены в таблице 5.18.

Таблица 5.18

Значение  $p_D$  при декодировании жесткого решения детектора

Применяемый код	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
ТСБК по формуле	$2,4 \cdot 10^{-2}$	$5,2 \cdot 10^{-4}$	$5,5 \cdot 10^{-6}$	$5,6 \cdot 10^{-8}$
ТСБК (результаты работы программного имитатора)	$5,2 \cdot 10^{-2}$	$5,2 \cdot 10^{-4}$	$5,5 \cdot 10^{-6}$	$5,6 \cdot 10^{-8}$
ТССК (результаты работы программного имитатора)	$3,1 \cdot 10^{-2}$	$1,3 \cdot 10^{-4}$	$8,5 \cdot 10^{-7}$	$4,7 \cdot 10^{-9}$
ТССК с выкалыванием двух бит (результаты работы программного имитатора)	$4,9 \cdot 10^{-2}$	$5,4 \cdot 10^{-4}$	$6,1 \cdot 10^{-6}$	$6,0 \cdot 10^{-8}$

Ввиду того, что при одинаковой длине информационной последовательности (4 бита) длина кодового слова ТССК (12 бит) на 2 бита превышает длину кодового слова ТСБК (10 бит), вызывает интерес сравнение корректирующей способности ТСБК с ТССК при одинаковой длине кодового слова. В связи с этим в нижней строке таблицы 5.18 приведены результаты работы программного имитатора ТССК при выкалывании двух бит в кодовом слове турбокода (см. п. 4.2.2). Проанализировав полученные данные, можно сделать следующие выводы:

- расчеты по формуле (5.7) для значений  $p_B \leq 1 \cdot 10^{-2}$  совпадают с результатами работы программы;
- корректирующая способность ТСБК ниже, чем у ТССК;
- при наличии в кодовом слове ТССК двух выколотых бит корректирующие способности турбокодов отличаются незначительно.

В дополнение к сделанным выводам следует отметить существенную разницу значений  $p_D$ , полученных по формуле (5.7) и по результатам работы программного имитатора при  $p_B = 1 \cdot 10^{-1}$ . По всей видимости, это вызвано тем, что формула (5.7) не учитывает возникновение ошибки декодирования при наличии в кодовом слове более двух ошибок, что при  $p_B = 1 \cdot 10^{-1}$  достаточно вероятно.

Известно, что корректирующая способность турбокодов значительно возрастает при декодировании мягких решений детектора. Результаты работы программных имитаторов ТСБК и ТССК для случая декодирования мягкого решения детектора, квантованного на 8 уровней, приведены в таблице 5.19. Рассмотрев полученные результаты, можно сделать следующие выводы:

- корректирующая способность ТСБК значительно ниже, чем у ТССК;
- при наличии в кодовом слове ТССК двух выколотых бит корректирующие способности турбокодов отличаются незначительно.

Таблица 5.19

Значение  $p_D$  при декодировании мягкого решения детектора

Применяемый код	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
ТСБК (результаты работы программного имитатора)	$2,50 \cdot 10^{-2}$	$5,47 \cdot 10^{-5}$	$1,40 \cdot 10^{-7}$	$3,79 \cdot 10^{-10}$
ТССК (результаты работы программного имитатора)	$9,55 \cdot 10^{-3}$	$2,81 \cdot 10^{-6}$	$3,50 \cdot 10^{-9}$	$2,91 \cdot 10^{-12}$
ТССК с выкалыванием двух бит (результаты работы программного имитатора)	$1,96 \cdot 10^{-2}$	$2,90 \cdot 10^{-5}$	$1,01 \cdot 10^{-7}$	$2,4 \cdot 10^{-10}$

Исследуем возможности ТСБК по адаптации к изменениям характеристик канала передачи данных, рассмотрев влияние перфорации на корректирующую способность ТСБК. Результаты работы программного имитатора для случая декодирования жесткого решения детектора и наличия выколотых бит приведены в таблице 5.20. Проведя анализ полученных значений, можно сделать вывод о том, что с ростом числа выколотых бит до трех, происходит плавное увеличение  $p_D$ . При выкалывании четырех бит корректирующая способность резко снижается. При числе выколотых бит более трех  $p_D > p_B$ , и процедура кодирования не имеет смысла. Таким образом, изменение кодовой скорости возможно от 0,3 до 0,43, то есть на 43 %.

Таблица 5.20

Значение  $p_D$  при разном количестве выколотых бит  
для случая декодирования жесткого решения детектора

Число выколотых бит	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
1	$6,0 \cdot 10^{-2}$	$7,3 \cdot 10^{-4}$	$6,9 \cdot 10^{-6}$	$7,2 \cdot 10^{-8}$
2	$6,1 \cdot 10^{-2}$	$7,4 \cdot 10^{-4}$	$7,2 \cdot 10^{-6}$	$7,3 \cdot 10^{-8}$
3	$6,6 \cdot 10^{-2}$	$8,6 \cdot 10^{-4}$	$8,4 \cdot 10^{-6}$	$9,6 \cdot 10^{-8}$
4	$6,7 \cdot 10^{-2}$	$1,0 \cdot 10^{-2}$	$1,0 \cdot 10^{-3}$	$1,0 \cdot 10^{-4}$

Рассмотрим влияние перфорации на успешность декодирования мягкого решения детектора ТСБК. Результаты работы программного имитатора приведены в таблице 5.21. Проведя анализ полученных значений, можно сделать вывод о том, что с увеличением числа выколотых бит повышение  $p_D$  происходит более резко, чем для случая декодирования жесткого решения. При количестве выколотых бит более пяти  $p_D > p_B$ , и процедура кодирования не имеет смысла. Таким образом, при декодировании мягкого решение детектора изменение кодовой скорости возможно от 0,3 до 0,6, то есть на 100 %.

Значение  $p_D$  при разном количестве выколотых бит  
для случая декодирования мягкого решения детектора

Число выколотых бит	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
1	$3,2 \cdot 10^{-2}$	$1,0 \cdot 10^{-4}$	$4,6 \cdot 10^{-7}$	$2,4 \cdot 10^{-9}$
2	$3,4 \cdot 10^{-2}$	$1,8 \cdot 10^{-4}$	$9,7 \cdot 10^{-7}$	$4,9 \cdot 10^{-9}$
3	$4,3 \cdot 10^{-2}$	$3,8 \cdot 10^{-4}$	$3,7 \cdot 10^{-6}$	$4,2 \cdot 10^{-8}$
4	$6,7 \cdot 10^{-2}$	$2,6 \cdot 10^{-3}$	$1,4 \cdot 10^{-4}$	$9,3 \cdot 10^{-6}$
5	$8,2 \cdot 10^{-2}$	$5,5 \cdot 10^{-3}$	$4,4 \cdot 10^{-4}$	$3,3 \cdot 10^{-5}$
6	$1,0 \cdot 10^{-1}$	$1,0 \cdot 10^{-2}$	$1,0 \cdot 10^{-3}$	$1,0 \cdot 10^{-4}$

Скорость реализации итеративного процесса декодирования можно варьировать, изменяя число итераций. Результаты работы программного имитатора для случая декодирования жесткого решения детектора приведены в таблице 5.22. Проведя анализ полученных значений, можно сделать вывод о том, что с увеличением числа итераций вероятность появления ошибочного бита в декодированном информационном сообщении снижается, однако при числе итераций более трех это снижение крайне незначительно. Результаты работы программного имитатора для случая декодирования мягкого решения детектора приведены в таблице 5.23. По полученным данным можно сделать вывод о том, что с увеличением числа итерации с одной до трех происходит резкое снижение вероятности появления ошибочного бита в декодированном информационном сообщении (более резкое, чем для случая декодирования жесткого решения детектора). При числе итераций более трех снижение  $p_D$  крайне незначительно.

В параграфе даны количественные оценки корректирующей способности турбокода на основе СБК при декодировании согласно максимуму апостериорной вероятности. Для описанного турбокода осуществлен вывод формулы расчета вероятности битовой ошибки в декодированном информационном сообщении. Продемонстрирована возможность гибкой адаптации параметров турбокода без

изменения алгоритма декодирования, путем введения перфорации. При декодировании жесткого решения детектора изменение кодовой скорости возможно от 0,3 до 0,5, то есть на 67 %. При декодировании мягкого решения детектора изменение кодовой скорости возможно от 0,3 до 0,6, то есть на 100 %. Проведено сравнение корректирующей способности турбокодов на основе СБК и ССК при декодировании согласно максимуму апостериорной вероятности.

Таблица 5.22

Значение  $p_D$  при разном количестве итераций  
декодирования жесткого решения детектора

Число итераций декодирования	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
1	$6,21 \cdot 10^{-2}$	$8,49 \cdot 10^{-4}$	$8,41 \cdot 10^{-6}$	$7,22 \cdot 10^{-8}$
2	$5,67 \cdot 10^{-2}$	$7,51 \cdot 10^{-4}$	$8,12 \cdot 10^{-6}$	$6,31 \cdot 10^{-8}$
3	$5,30 \cdot 10^{-2}$	$5,21 \cdot 10^{-4}$	$5,31 \cdot 10^{-6}$	$5,69 \cdot 10^{-8}$
4	$5,29 \cdot 10^{-2}$	$5,02 \cdot 10^{-4}$	$5,03 \cdot 10^{-6}$	$5,42 \cdot 10^{-8}$
5	$5,30 \cdot 10^{-2}$	$5,01 \cdot 10^{-4}$	$4,81 \cdot 10^{-6}$	$5,21 \cdot 10^{-8}$
6	$5,30 \cdot 10^{-2}$	$4,99 \cdot 10^{-4}$	$4,39 \cdot 10^{-6}$	$4,90 \cdot 10^{-8}$

Таблица 5.23

Значение  $p_D$  при разном количестве итераций  
декодирования мягкого решения детектора

Число итераций декодирования	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
1	$4,74 \cdot 10^{-2}$	$7,10 \cdot 10^{-4}$	$1,31 \cdot 10^{-5}$	$2,95 \cdot 10^{-7}$
2	$2,93 \cdot 10^{-2}$	$8,84 \cdot 10^{-5}$	$2,87 \cdot 10^{-7}$	$6,80 \cdot 10^{-10}$
3	$2,55 \cdot 10^{-2}$	$5,47 \cdot 10^{-5}$	$1,40 \cdot 10^{-7}$	$3,79 \cdot 10^{-10}$
4	$2,43 \cdot 10^{-2}$	$4,11 \cdot 10^{-5}$	$1,08 \cdot 10^{-7}$	$2,51 \cdot 10^{-10}$
5	$2,44 \cdot 10^{-2}$	$4,18 \cdot 10^{-5}$	$1,26 \cdot 10^{-7}$	$2,33 \cdot 10^{-10}$
6	$2,42 \cdot 10^{-2}$	$4,48 \cdot 10^{-5}$	$1,18 \cdot 10^{-7}$	$2,24 \cdot 10^{-10}$



### **5.3. Повышение корректирующей способности турбокодов путем снижения чувствительности декодера к достоверности значений информационных бит кодового слова**

Поток передаваемых данных, формируемый на выходе кодера турбокода, состоит из блока информационных бит и  $N$  блоков проверочных бит (см. рис. 4.9). Каждый информационный бит участвует в формировании  $N$  слов на входе декодера турбокода. Исходя из отмеченных особенностей, а также проведенных исследований в п. 4.2.2, можно сделать вывод о высокой чувствительности декодера турбокода к достоверности значений информационных бит кодового слова. Негативной стороной такой чувствительности является высокая вероятность ошибки декодирования при наличии ошибок в информационных битах кодового слова. В качестве способа повышения эффективности декодирования предлагается снизить чувствительность декодера к значениям информационных бит, аналогично рассмотренному в п. 4.4 [185, 186], с учетом того, что в п. 4.4 рассматривалось снижение чувствительности декодера к достоверности значений проверочных бит кодового слова.

Проверку действенности предложенного способа предлагается начать путем оценки успешности декодирования содержащих ошибки кодовых слов ТССК.

Для случая информационной последовательности 1100 при декодировании (см. п. 4.2) формируется кодовая последовательность турбокода, состоящая из блока информационных бит и двух блоков проверочных бит 1100 1001 0010. Пусть передача происходит биполярными сигналами, тогда кодовая последовательность принимает следующий вид: 11-1-1 1-1-11 -1-11-1.

Изучим влияние расположения ошибок в той или иной части кодового слова турбокода на результат декодирования. Рассмотрим три варианта расположения трех ошибок в ранее сформированном кодовом слове ТССК: в блоке информационных бит, в первом блоке проверочных бит, во втором блоке проверочных бит. Пусть ошибки располагаются в 1, 2, и 4 битах блока. В случае

расположения ошибок в блоке информационных бит на входе декодера ТССК имеем последовательность: -1-1-11 1-1-11 -1-11-1. Здесь и далее ошибочные биты выделены подчеркиванием. Входные последовательности первого и второго декодеров имеют вид: -11-1-1-1-111 и -1-11-1-11-1-1 соответственно. В случае расположения ошибок в первом блоке проверочных бит на входе декодера ТССК имеем последовательность: 11-1-1 -11-1-1 -1-11-1. При этом входные последовательности первого и второго декодеров имеют вид: 1-111-1-1-1-1 и -1-1-1-1111-1 соответственно. В случае расположения ошибок во втором блоке проверочных бит на входе декодера ТССК имеем последовательность: 11-1-1 1-1-11 1111. В данном случае входные последовательности первого и второго декодеров имеют вид: 111-1-1-1-11 и -11-111111 соответственно. Анализируя последовательности на входе декодеров из состава декодера турбокода, можно отметить тот факт, что ошибочные символы в блоке информационных бит попадают на вход каждого из декодеров, то есть общее число ошибок удваивается.

Для оценки эффективности корректирующей способности описанного турбокода произведена разработка в MS Excel реализации декодера одного кодового слова ТССК, позволяющего осуществлять декодирование жесткого решения детектора и функционирующего на основе описанного алгоритма MAP. Произведены расчеты выхода декодера для трех описанных вариантов расположения ошибочных бит при трех итерациях декодирования. Мягкие решения о значении информационных бит на выходе декодера приведены в таблице 5.24. Для случая наличия трех ошибок в информационных битах кодового слова в декодированном информационном сообщении присутствуют ошибки. Для случаев наличия трех ошибок в блоках проверочных бит выход декодера не содержит ошибок. При других вариантах расположения трех ошибок в блоках кодового слова результаты аналогичны. Таким образом, подтверждено предположение, что при расположении ошибок в информационных битах вероятность ошибочного декодирования кодового слова выше, чем при расположении ошибок в проверочных битах.

Значения мягкого выхода декодера турбокода при трех вариантах расположения ошибок

№ бита	значение переданного бита	Расположение ошибок		
		в блоке информационных бит	в блоке проверочных бит первого кодера	в блоке проверочных бит второго кодера
1	1	<u>-21,4</u>	20,6	36,7
2	1	<u>-21,2</u>	18,3	29,4
3	-1	-43,6	-36,7	-20,6
4	-1	<u>15,4</u>	-29,4	-18,3

Рассмотренные примеры подтвердили высказанное предположение о высокой чувствительности декодера к значениям информационных бит. Предлагается к рассмотрению способ искусственного снижения высокой чувствительности декодера к достоверности значений информационных бит путем умножения их значений на входе декодера на понижающий коэффициент  $K < 1$ . При этом можно спрогнозировать следующие эффекты. Положительный эффект – вероятность ошибочного декодирования кодового слова турбокода при наличии ошибок в информационных битах снизится. Отрицательный эффект – вероятность ошибочного декодирования кодового слова турбокода при наличии ошибок в блоках проверочных бит возрастет. В силу доказанного ранее большего влияния значений информационных бит на результат декодирования предполагается, что положительный эффект превысит отрицательный.

Рассмотрим действенность предлагаемой модификации метода декодирования на практическом примере. При  $K = 0,5$  оценим результаты декодирования описанных ранее вариантов расположения трех ошибок в кодовом слове ТССК. В случае расположения ошибок в блоке информационных бит входные последовательности первого и второго декодеров имеют вид: -0,5 1 -0,5 -1 -0,5 -1 0,5 1 и -0,5 -1 0,5 -1 -0,5 1 -0,5 -1 соответственно. В случае расположения ошибок в первом блоке проверочных бит входные последовательности первого и второго декодеров имеют вид: 0,5 -1 0,5 1 -0,5 -1

$-0,5 \underline{-1}$  и  $-0,5 \underline{-1} \underline{-0,5} \underline{-1} \underline{0,5} \underline{1} \underline{0,5} \underline{-1}$ . В случае расположения ошибок во втором блоке проверочных бит входные последовательности первого и второго декодеров имеют вид:  $0,5 \underline{1} \underline{0,5} \underline{-1} \underline{-0,5} \underline{-1} \underline{-0,5} \underline{1}$  и  $-0,5 \underline{1} \underline{-0,5} \underline{1} \underline{0,5} \underline{1} \underline{0,5} \underline{1}$ . Результаты декодирования приведены в таблице 5.25. Для всех вариантов расположения ошибок в кодовом слове информационное сообщение декодировано без ошибок. При этом наблюдается некоторое снижение абсолютных значений мягкого выхода декодера при расположении ошибок в проверочных битах в сравнении с данными таблицы 5.24. На приведенном примере продемонстрирована способность предлагаемой модификации метода декодирования повысить корректирующую способность турбокода.

Так как декодирование ТСБК и ТССК осуществляется согласно максимуму апостериорной вероятности, можно предположить, что и для ТСБК предлагаемая модификация метода декодирования приведет к повышению корректирующей способности.

Таблица 5.25

Значения мягкого выхода декодера турбокода при  $K = 0,5$

№ бита	значение переданного бита	Расположение ошибок		
		в блоке информационных бит	в блоке проверочных бит первого кодера	в блоке проверочных бит второго кодера
1	1	5,5	6,7	21,1
2	1	3,9	13,3	20,2
3	-1	-28,2	-21,1	-6,7
4	-1	-6,1	-20,2	-13,3

В связи с изложенным, представляется целесообразным в качестве следующего шага в подтверждение действенности предлагаемой модификации метода декодирования осуществить исследования, дающие количественную информацию о значении вероятности появления ошибочного бита в декодированном информационном сообщении  $p_D$  в зависимости от вероятности ошибки в канале передачи данных  $p_B$  и величины  $K$ .

В целях реализации заявленных исследований проведена модификация программных продуктов, описанных в п. 4.2 и п. 5.2, моделирующих работу ТССК и ТСБК соответственно [184, 198]. Модификация произведена на основе изменения алгоритмов декодирования, позволяющих реализовать умножение значений информационных бит на входе декодера на понижающие коэффициенты.

Исследования ТССК осуществлялись для случаев длины блока информации, равной четырем и восьми битам. При длине блока информации, равной четырем битам, минимальная величина  $p_D$  получена при умножении принятых значений информационных бит на  $K = 0,6$  – для случая декодирования жесткого решения детектора, и на  $K = 0,65$  – для случая декодирования мягкого решения детектора. В таблице 5.26 приведены данные о значениях  $p_D$  по результатам работы программного имитатора. Проведя анализ результатов, можно сделать вывод о снижении вероятности появления ошибочного бита в декодированном информационном сообщении при умножении информационных бит на  $K$ . Эффект увеличивается при снижении  $p_B$ . Вывод одинаков как для случая декодирования жесткого, так и для случая декодирования мягкого решения детектора.

Таблица 5.26

Полученные значения  $p_D$  для ТССК при длине блока информации, равной 4 бита

Особенности декодирования	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
декодирование жесткого решения детектора	$3,09 \cdot 10^{-2}$	$1,32 \cdot 10^{-4}$	$8,49 \cdot 10^{-7}$	$4,70 \cdot 10^{-9}$
декодирование жесткого решения детектора, $K = 0,6$	$2,82 \cdot 10^{-2}$	$4,31 \cdot 10^{-5}$	$6,08 \cdot 10^{-8}$	$3,01 \cdot 10^{-10}$
декодирование мягкого решения детектора	$9,55 \cdot 10^{-3}$	$2,81 \cdot 10^{-6}$	$3,50 \cdot 10^{-9}$	$2,91 \cdot 10^{-12}$
декодирование мягкого решения детектора, $K = 0,65$	$8,72 \cdot 10^{-3}$	$1,77 \cdot 10^{-6}$	$1,82 \cdot 10^{-9}$	$1,20 \cdot 10^{-12}$

Для улучшения восприятия данные из таблицы 5.26 представлены в графическом виде (см. рис. 5.4 – 5.5). Пунктиром приведены графики для  $K = 1$ .

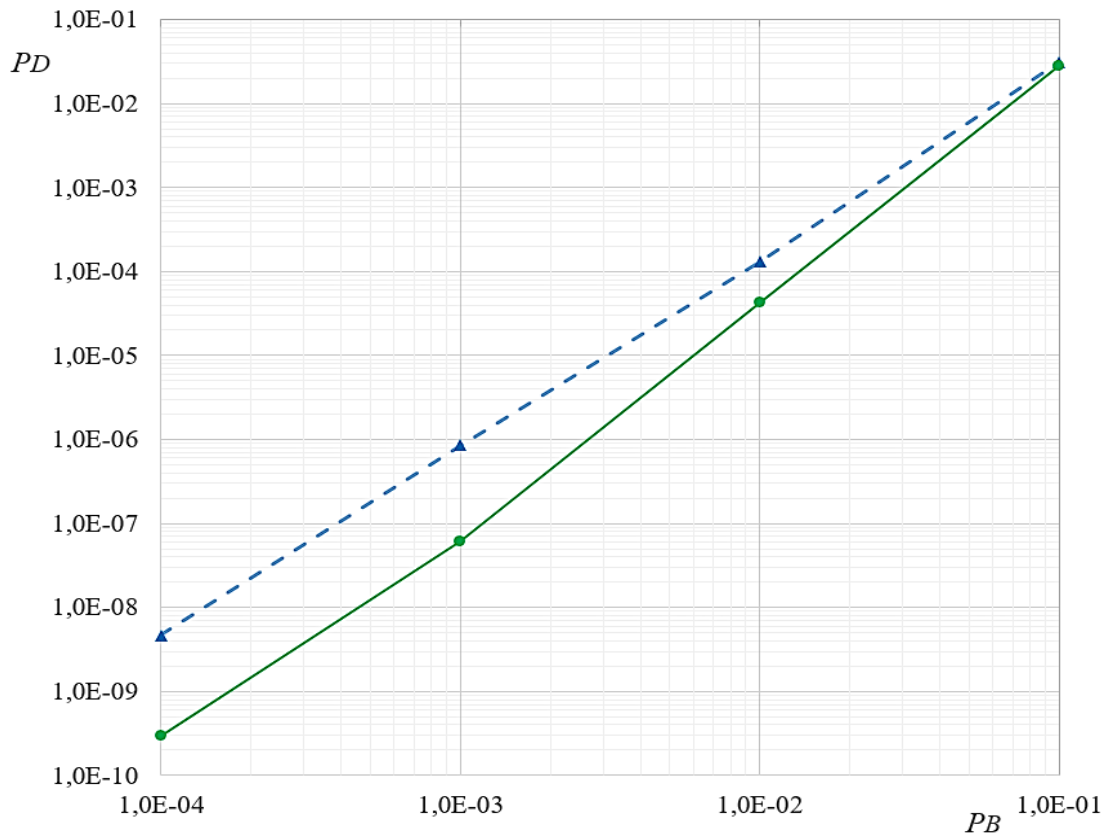


Рисунок 5.4. Оценка корректирующей способности ТССК при использовании коэффициентов для случая декодирования жесткого решения детектора

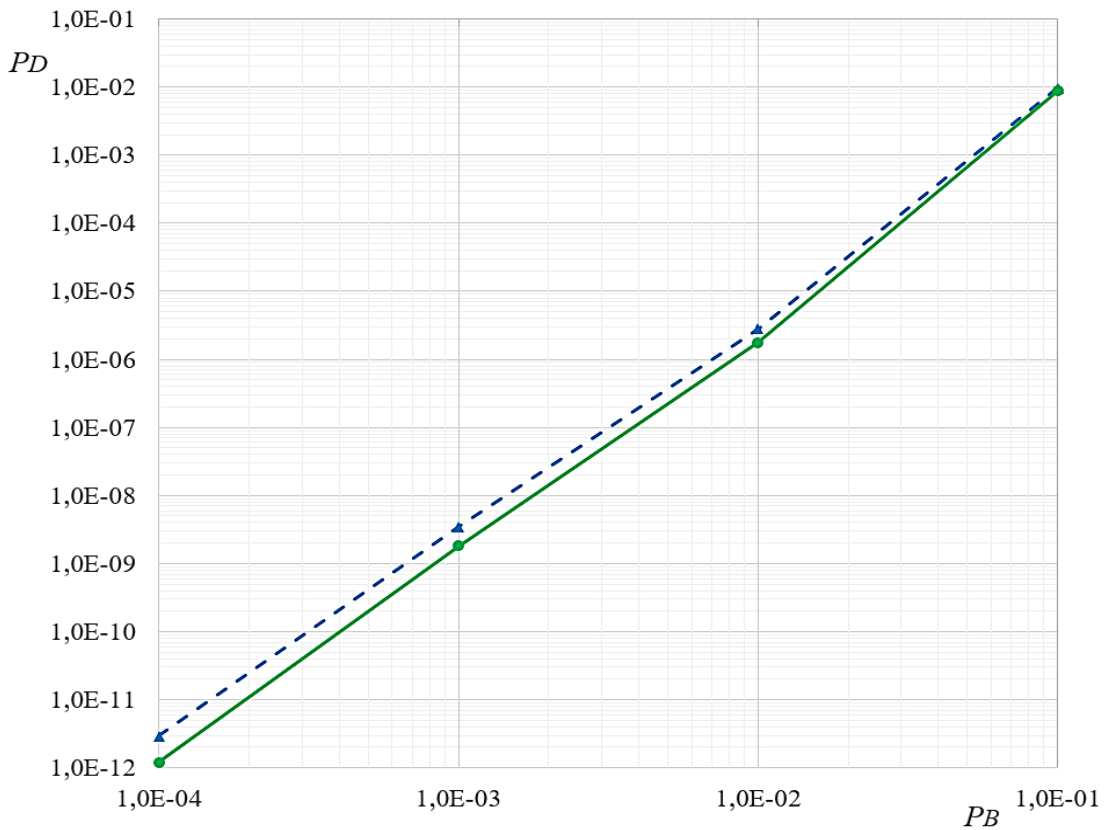


Рисунок 5.5. Оценка корректирующей способности ТССК при использовании коэффициентов для случая декодирования мягкого решения детектора

В ходе исследования ТССК при длине блока информации, равной восьми битам, минимальная величина  $p_D$  получена при умножении принятых значений информационных бит на  $K = 0,65$  – при декодировании жесткого решения детектора и на  $K = 0,6$  – при декодировании мягкого решения детектора. В таблице 5.27 приведены данные о значениях  $p_D$  по результатам работы программного имитатора. Выводы по итогам анализа результатов полностью совпадают с выводами для случая длины блока информации, равной четырем битам.

В ходе исследования ТСБК при декодировании жесткого решения детектора влияние значения  $K$  на величину  $p_D$  обнаружено не было, что может быть связано с особенностью построения блочного кода Хемминга. Для случая декодирования мягкого решения детектора минимальная величина  $p_D$  получена при умножении принятых значений информационных бит на  $K = 0,45$ . В таблице 5.28 приведены данные о значениях  $p_D$  по результатам работы программного имитатора. Проведя анализ результатов, можно сделать вывод о снижении вероятности появления ошибочного бита в декодированном информационном сообщении при умножении информационных бит на  $K$ . Как и при исследовании ТССК, зафиксировано, что эффект от применения  $K$  увеличивается при снижении  $p_B$ . Для улучшения восприятия данные таблицы 5.28 представлены в графическом виде (см. рис. 5.6). Пунктиром приведен график для  $K = 1$ .

Таблица 5.27

Полученные значения  $p_D$  для ТССК при длине блока информации, равной 8 бит

Особенности декодирования	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$
декодирование жесткого решения детектора	$2,91 \cdot 10^{-2}$	$7,91 \cdot 10^{-5}$	$5,75 \cdot 10^{-7}$
декодирование жесткого решения детектора, $K = 0,65$	$2,62 \cdot 10^{-2}$	$3,31 \cdot 10^{-5}$	$7,01 \cdot 10^{-8}$
декодирование мягкого решения детектора	$8,34 \cdot 10^{-3}$	$2,12 \cdot 10^{-6}$	$2,08 \cdot 10^{-9}$
декодирование мягкого решения детектора, $K = 0,6$	$7,11 \cdot 10^{-3}$	$1,07 \cdot 10^{-6}$	$6,04 \cdot 10^{-10}$

По итогам анализа результатов проведенных экспериментов можно сделать заключение, что применение предложенной модификации метода декодирования приводит к повышению корректирующей способности рассмотренных турбокодов.

Таблица 5.28

Полученные значения  $p_D$  для ТСБК

Особенности декодирования	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$	$p_B=1 \cdot 10^{-4}$
декодирование мягкого решения детектора	$2,5 \cdot 10^{-2}$	$5,5 \cdot 10^{-5}$	$1,4 \cdot 10^{-7}$	$3,8 \cdot 10^{-10}$
декодирование мягкого решения детектора, $K = 0,45$	$2,3 \cdot 10^{-2}$	$3,3 \cdot 10^{-5}$	$6,0 \cdot 10^{-8}$	$1,5 \cdot 10^{-10}$

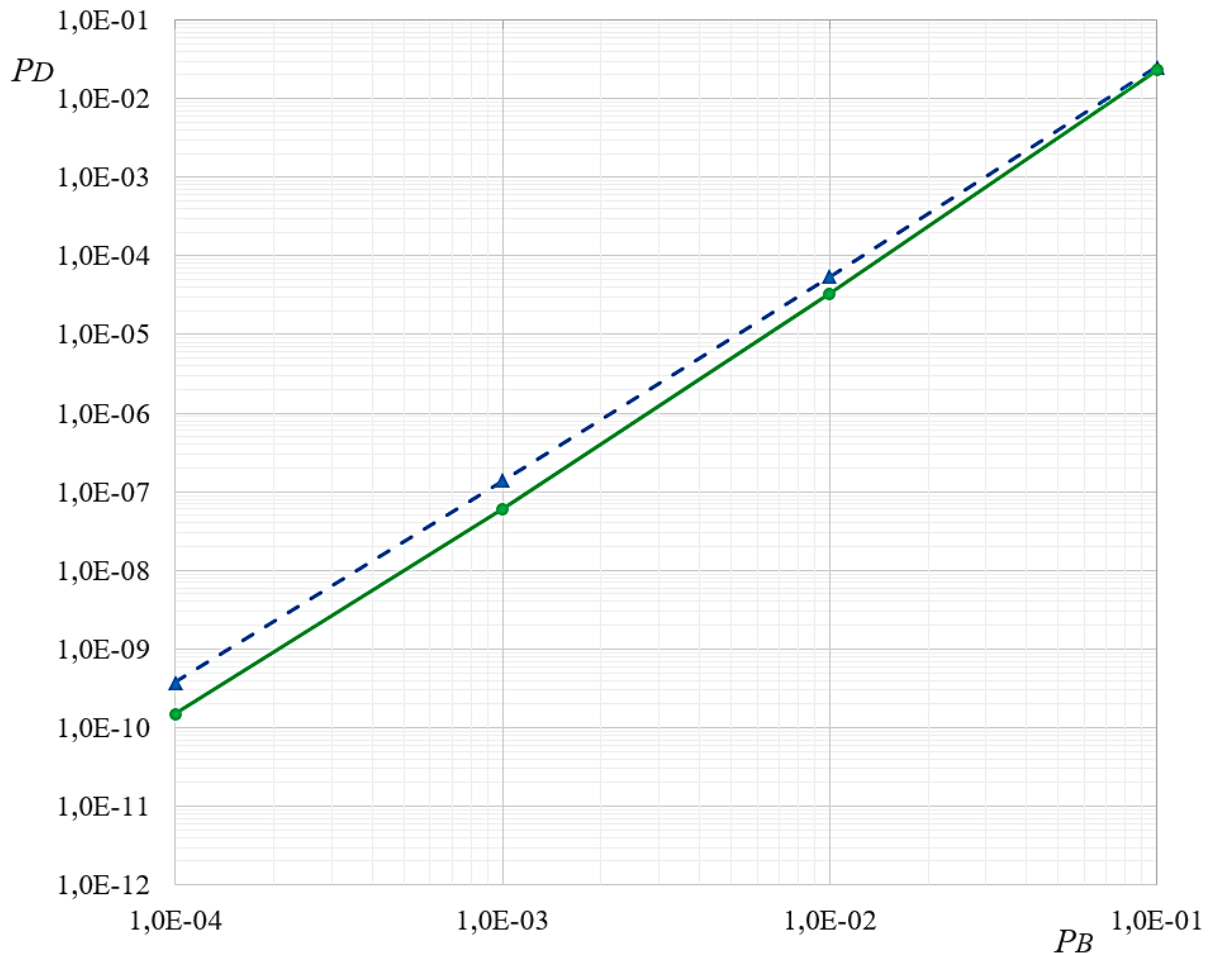


Рисунок 5.6. Оценка корректирующей способности ТСБК при использовании коэффициентов, для случая декодирования мягкого решения детектора

В параграфе рассмотрена реализация повышения корректирующей способности турбокодов путем снижения чувствительности декодера к значениям



информационных бит кодового слова. Путем анализа особенности декодирования кодового слова, а также экспериментальным путем выявлена высокая чувствительность декодера к достоверности значений информационных бит кодового слова турбокода. Негативной стороной такой чувствительности является высокая вероятность ошибки декодирования при наличии ошибок в информационных битах кодового слова. Рассмотрены изменения метода декодирования по максимуму апостериорной вероятности при применении понижающих коэффициентов для декодируемых значений информационных бит кодового слова. Применение коэффициентов позволило снизить вероятность битовой ошибки в декодированном информационном сообщении. В частности, для турбокода на основе сверточного кода при декодировании жесткого решения детектора с применением коэффициентов, снижение вероятности битовой ошибки в декодированном информационном сообщении составило: 1,1 раза (с  $3,1 \cdot 10^{-2}$  до  $2,82 \cdot 10^{-2}$ ) при  $p_B = 1 \cdot 10^{-1}$ , 3 раза (с  $1,3 \cdot 10^{-4}$  до  $4,3 \cdot 10^{-5}$ ) при  $p_B = 1 \cdot 10^{-2}$ , 14 раз (с  $8,5 \cdot 10^{-7}$  до  $6,1 \cdot 10^{-8}$ ) при  $p_B = 1 \cdot 10^{-3}$ .

Проблему высокой чувствительности декодера турбокода к достоверности значений информационных бит также можно решить путем их дополнительного кодирования, что будет рассмотрено в следующем параграфе.

#### **5.4. Построение комбинированного кода на основе турбокода с дополнительным кодированием информационных бит**

В п. 5.3 предложено снижать высокую чувствительность декодера к достоверности значений информационных бит кодового слова турбокода путем применения понижающих коэффициентов для декодируемых значений информационных бит кодового слова. Можно предположить, что дополнительное кодирование всех или определенного числа информационных бит кодового слова турбокода повысит их достоверность, что благоприятно отразится на

корректирующей способности кода [199, 200]. Будем называть такой турбокод – турбокодом с дополнительным кодированием информационных бит (далее – ТДК).

Предлагается повысить корректирующую способность турбокода на основе систематического сверточного кода, защитив информационные биты от возникновения в них ошибок, дополнительно кодируя их еще раз, например, по столбцам блока, как изображено на рисунке 5.7, где направления дополнительного кодирования показаны стрелками. В таком случае на приемной стороне сначала происходит декодирование дополнительного кода, а затем турбокода. Очевидно, что при конструировании ТДК, согласно рисунку 5.7, вероятность появления ошибок в информационных битах кодового слова на входе декодера турбокода будет снижена.

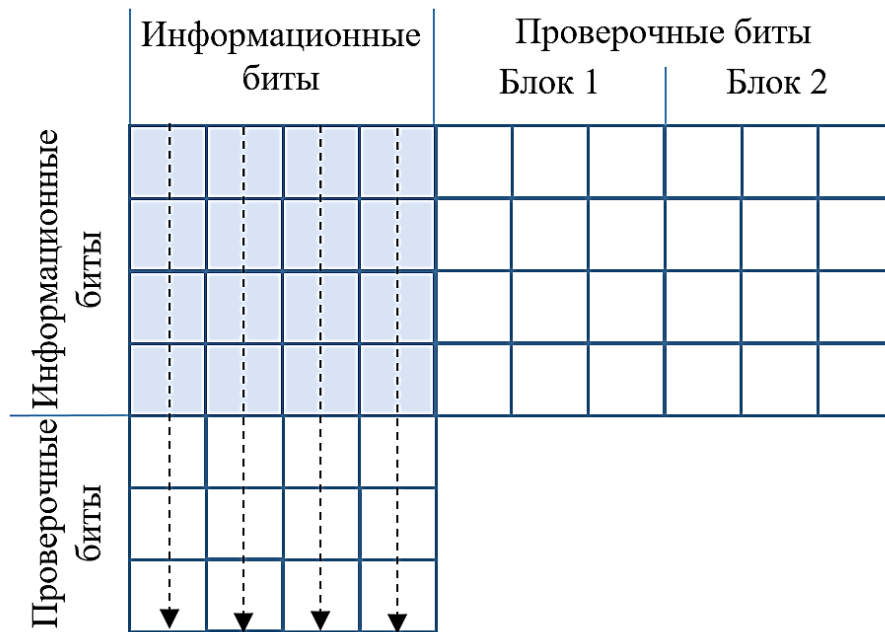


Рисунок 5.7. Структурная схема реализации дополнительного кодирования информационных бит

Недостатком дополнительного кодирования всего блока информационных бит является значительный рост избыточности. Возникающую избыточность можно снизить, дополнительно кодируя только часть информационных бит. Значения терминирующих бит заранее известны. При наличии в информационной последовательности терминирующих бит подвергать их повторному кодированию не целесообразно. Схематично такой вариант кодирования изображен на рисунке

5.8. Можно применять дополнительное кодирование некоторых столбцов информационных бит, например, через один столбец, как показано на рисунке 5.9.

Применение терминирующей последовательности для защиты последнего информационного бита можно заменить его дополнительным кодированием, также можно защитить и информационный бит, который станет последним после перемежения. При этом можно наблюдать значительное снижение избыточности. Так, например, можно перейти от блока размером 96 бит к структуре из 78 бит, как показано на рисунке 5.10.

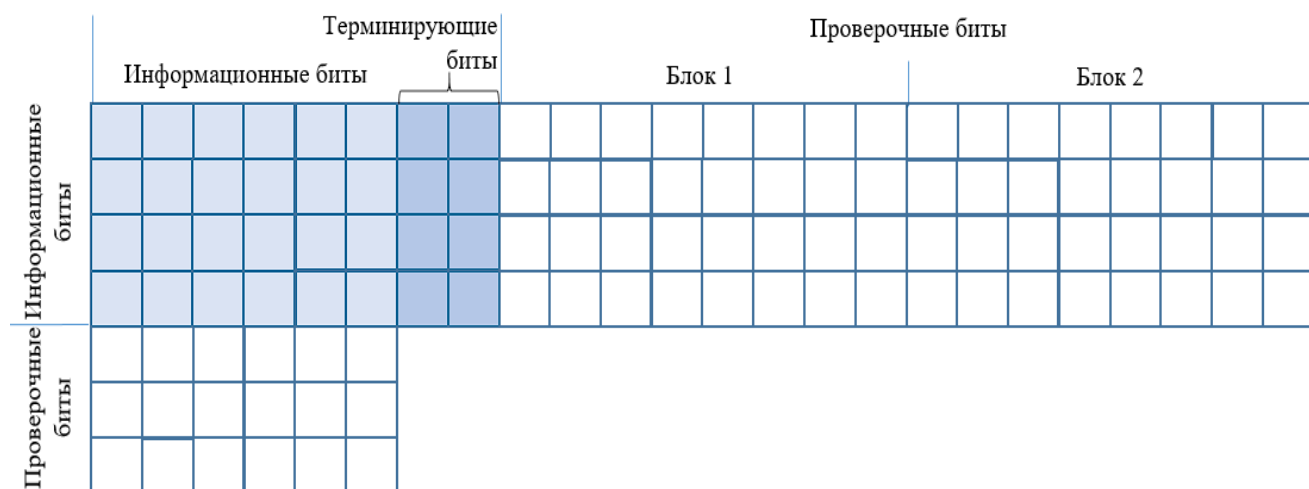


Рисунок 5.8. Структурная схема реализации дополнительного кодирования информационных бит за исключением терминирующих бит вспомогательным кодом

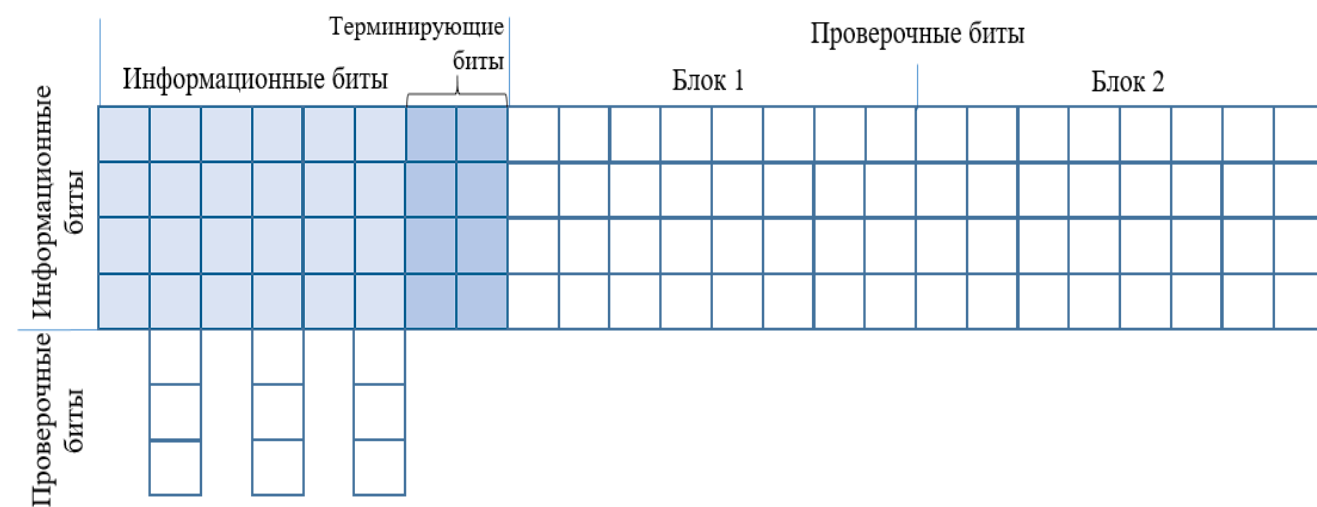


Рисунок 5.9. Структурная схема реализации дополнительного кодирования части информационных бит вспомогательным кодом

Продemonстрируем возможности по исправлению пакета ошибок в блоке информационных бит при использовании в качестве дополнительного кода Хемминга. Код Хемминга обладает минимальной избыточностью и способен исправить только один ошибочный бит для случая декодирования жесткого решения.

Пусть в блоке информационных бит, закодированных по столбцам кодом Хемминга, одно информационное слово полностью искажено пакетом ошибок, а также имеются две независимые ошибки. Описанный блок и результаты его декодирования кодом Хемминга схематично изображены на рисунке 5.11. Ошибочные биты на рисунке отмечены X-образным перечеркиванием. В процессе декодирования декодер кода Хемминга без ошибок декодировал четыре кодовых слова, содержащих по одному ошибочному биту. Два кодовых слова содержали по два ошибочных бита и были декодированы с ошибками (по итогам ошибочного декодирования в кодовом слове появился еще один ошибочный бит). В результате на вход декодера ПТК поступили четыре информационных слова. Одно информационное слово без ошибок, одно – с двумя ошибками, два – с одной ошибкой (см. рис. 5.11). С учетом того, что декодер ПТК способен успешно исправлять все варианты расположения двух ошибок, можно сделать вывод об эффективности использования дополнительного кода Хемминга для противостояния пакетам ошибок.

С целью более точной оценки корректирующей способности ТДК в ходе дальнейших исследований найдем вероятность битовой ошибки в декодированном информационном сообщении для разных вариантов дополнительного кодирования информационных бит кодового слова ТДК. В качестве дополнительного кода используем код Голея. Примем допущение, что ошибки, имеющиеся в символах на выходе декодера дополнительного кода, являются независимыми, для реализации чего в ряде случаев потребуется наличие перемежителя. Тогда, зная значения вероятности битовой ошибки  $p_V$  в декодированном информационном сообщении для дополнительного кода, можно смоделировать его работу, сгенерировав появление ошибок с вероятностью  $p_V$  на входе декодера турбокода.

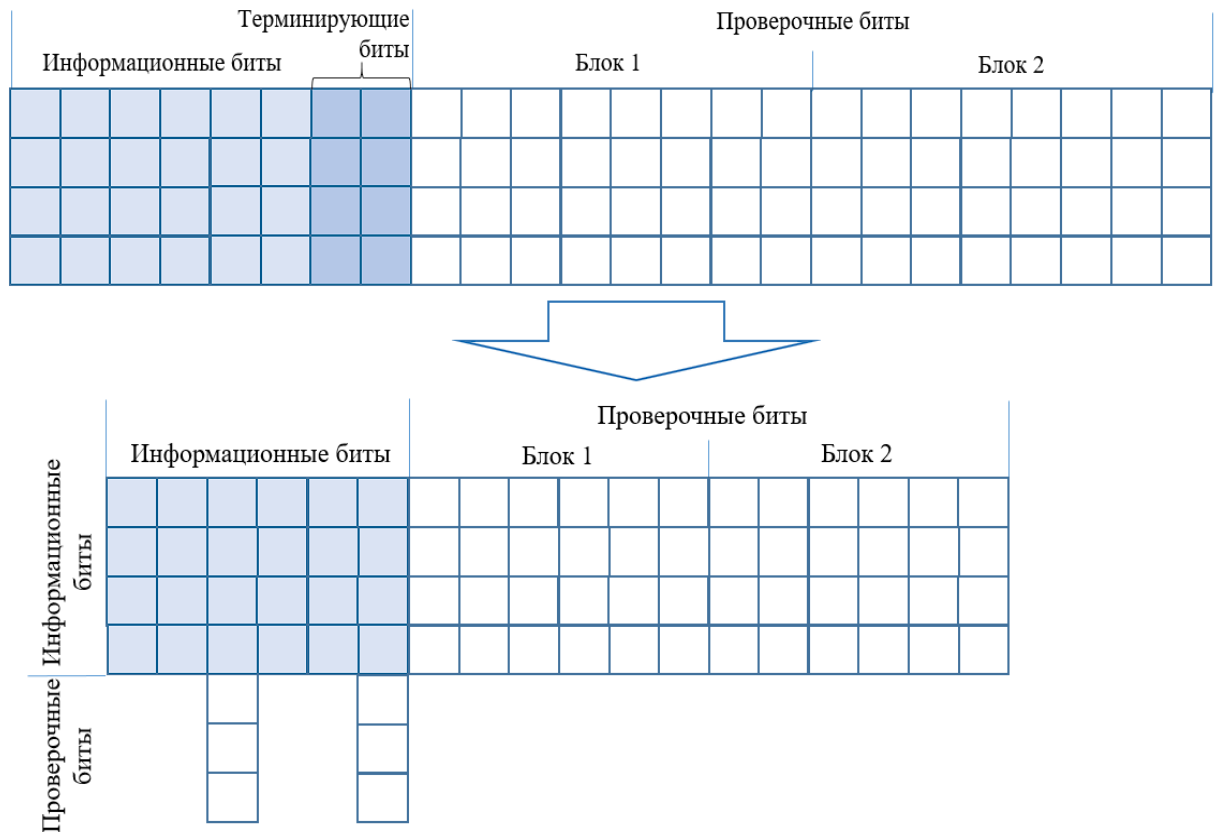


Рисунок 5.10. Структурная схема реализации повышения защищенности последних информационных бит кодового слова турбокода.

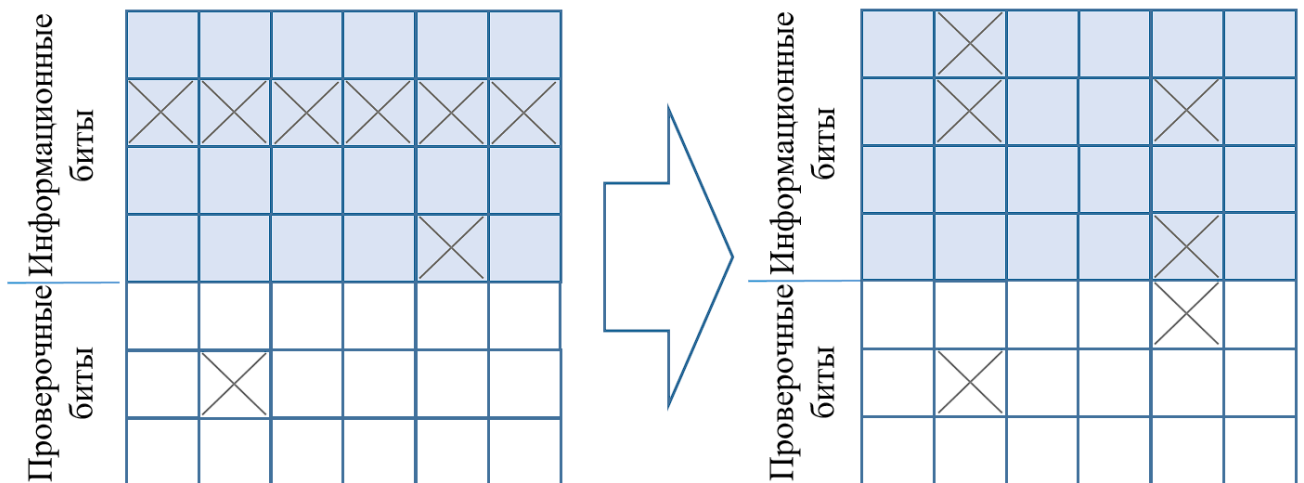


Рисунок 5.11. Схематичная демонстрация результатов декодирования блока информации.

С целью реализации исследований ТДК будет применен программный имитатор, созданный на языке программирования Python [184]. Программный имитатор содержит в своем составе следующие основные компоненты:

- генератор информационной последовательности;
- кодер ТССК;
- блок моделирования выхода детектора приемника с заданной вероятностью появления ошибочного бита  $p_B$ ;
- блок моделирования выхода декодера дополнительного кода с вероятностью появления ошибочного бита  $p_V$ ;
- декодер ТССК.

Используя результаты работы программного имитатора, подтвердим предположение о наибольшей целесообразности дополнительного кодирования информационных бит кодового слова турбокода. В таблице 5.29 приведены результаты расчета вероятности появления ошибочного бита в декодированном информационном сообщении при кодирования дополнительным кодов двух информационных бит и двух проверочных бит. По результатам анализа полученных данных можно сделать вывод о значительном выигрыше в корректирующей способности при дополнительном кодировании информационных бит кодового слова.

Таблица 5.29

Корректирующая способность ТДК при введении дополнительного кодирования в разные части кодового слова турбокода

Особенности введения дополнительного кодирования в кодовое слово турбокода	$p_B=1 \cdot 10^{-1}$	$p_B=1 \cdot 10^{-2}$	$p_B=1 \cdot 10^{-3}$
Дополнительное кодирование двух информационных бит кодового слова	$1,55 \cdot 10^{-2}$	$1,34 \cdot 10^{-5}$	$1,10 \cdot 10^{-8}$
Дополнительное кодирование двух проверочных бит кодового слова	$2,54 \cdot 10^{-2}$	$1,15 \cdot 10^{-4}$	$1,02 \cdot 10^{-7}$

Обозначим число информационных бит кодового слова турбокода, дополнительно закодированных кодом Голея, как  $T_D$ . С использованием программного имитатора проведем исследования ТДК, дающие количественную информацию о значении  $p_D$  в зависимости от следующих параметров:

- вероятности ошибки в канале передачи данных  $p_B$ ;

– значений  $T_D$ .

Результаты исследования ТДК приведены в таблице 5.30. Проанализировав результаты, можно сделать вывод о значительном повышении корректирующей способности турбокода при дополнительном кодировании даже одного бита из информационной части кодового слова турбокода. Путем дополнительного кодирования возможно изменение в широких пределах корректирующей способности, регулируя число дополнительно закодированных бит. Для улучшения восприятия данные из таблицы 5.30 представлены в графическом виде на рисунке 5.12.

Таблица 5.30

Корректирующая способность ТДК

$T_D$	$p_B=10^{-1}, p_V=3,1 \cdot 10^{-2}$	$p_B=10^{-2}, p_V=1,9 \cdot 10^{-5}$	$p_B=10^{-3}, p_V=2,3 \cdot 10^{-9}$
0	$3,1 \cdot 10^{-2}$	$1,3 \cdot 10^{-4}$	$8,5 \cdot 10^{-7}$
1	$2,32 \cdot 10^{-2}$	$7,31 \cdot 10^{-5}$	$3,9 \cdot 10^{-7}$
2	$1,55 \cdot 10^{-2}$	$1,34 \cdot 10^{-5}$	$1,1 \cdot 10^{-8}$
3	$1,20 \cdot 10^{-2}$	$8,01 \cdot 10^{-6}$	$4,0 \cdot 10^{-9}$

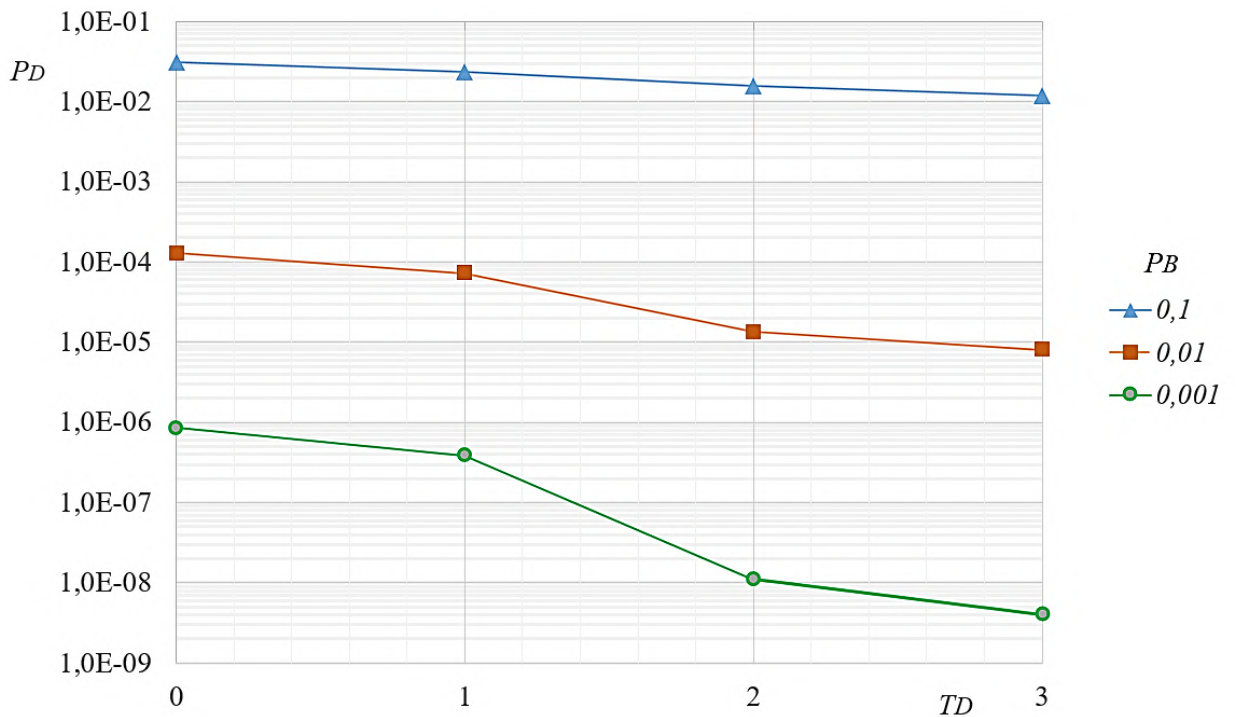


Рисунок 5.12. Вероятность появления ошибочного бита в декодированном информационном сообщении ТДК

Рассчитаем энергетические потери, связанные с введением дополнительной избыточности при дополнительном кодировании. Для дополнительного кодирования кодом Голя ( $k = 12, n = 23$ ) необходимо сформировать блок из 12 кодовых слов турбокода по 12 бит. Объем блока кодовых бит турбокода равен 144 бита, при этом число информационных бит равно 48 бит. Кодовая скорость равна 0,33. Одно кодовое слово кода Голя имеет 11 проверочных бит. При кодировании дополнительным кодом одного бита кодового слова турбокода объем блока кодовых бит составит:  $144 + 11 = 155$ , кодовая скорость снижается с 0,33 до 0,31. При кодировании дополнительным кодом двух бит кодового слова турбокода кодовая скорость снижается с 0,33 до 0,29. При кодировании дополнительным кодом трех бит кодового слова турбокода кодовая скорость снижается с 0,33 до 0,27. Снижение кодовой скорости при неизменном значении времени на передачу сообщения может быть конвертировано в энергетические потери, вызванные снижением энергии одного бита. По характеристике манипуляции BPSK, представленной на рисунке 2.8, можно рассчитать энергетический выигрыш, вызванный снижением вероятности битовой ошибки в декодированном информационном сообщении. Проведя анализ данных, представленных в таблице 5.31, можно сделать вывод о том, что энергетический выигрыш, вызванный снижением вероятности битовой ошибки в декодированном информационном сообщении, превышает потери, вызванные снижением кодовой скорости.

Таблица 5.31

## Энергетический выигрыш от применения дополнительного кодирования

$T_D$	Потери из-за введения дополнительной избыточности, дБ	Выигрыш за счет снижения $p_D$ , дБ		
		$p_B = 1 \cdot 10^{-1}$	$p_B = 1 \cdot 10^{-2}$	$p_B = 1 \cdot 10^{-3}$
1	0,32	0,6	0,4	0,35
2	0,62	1,4	1,19	1,4
3	0,9	1,7	1,4	1,7



В параграфе рассмотрено построение комбинированного кода на основе турбокода с дополнительным кодированием информационных бит. Таким образом предложено решение проблемы высокой чувствительности декодера турбокода к достоверности значений информационных бит кодового слова турбокода. Для исследуемого турбокода при дополнительном кодировании лишь одного информационного бита кодового слова турбокода зафиксировано снижение вероятности появления ошибочного бита в декодированном информационном сообщении в 2,18 раза при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ . Изменяя число бит, подвергшихся дополнительному кодированию, можно регулировать кодовую скорость и корректирующую способность комбинированного кода. При дополнительном кодировании трех информационных бит кодовая скорость изменилась с 0,33 до 0,27, а вероятность битовой ошибки в декодированном информационном сообщении снизилась в 212 раз. Для случая использования модуляции BPSK произведен расчет энергетического выигрыша от применения дополнительного кодирования информационных бит кодового слова турбокода, который составил 0,28 дБ – при дополнительном кодировании одного бита, 0,78 дБ – при дополнительном кодировании двух бит, 0,8 дБ – при дополнительном кодировании трех бит.

### **5.5. Повышение корректирующей способности комбинированного кода на основе турбокода с дополнительным кодированием информационных бит**

Достоверность значений кодовых бит турбокода, закодированных дополнительным кодом, выше чем достоверность значений бит кодового слова турбокода не подвергшихся дополнительному кодированию. Для рассмотренного в п. 5.4 ТДК учесть большую степень достоверности дополнительно закодированных информационных символов можно путем умножения их значений на повышающие коэффициенты  $K > 1$ , или путем умножения значений менее достоверных символов на понижающие коэффициента  $K < 1$ . Предполагается, что

при этом повысится корректирующая способность ТДК. Аналогичное предположение было описано в п. 2.7 и п. 3.9 при рассмотрении возможностей по повышению эффективности последовательных каскадных кодов [199, 200].

Корректирующую способность ТДК при наличии ошибок в проверочных битах можно повысить путем умножения их значений на коэффициент  $K < 1$  перед поступлением на вход декодера в ущерб корректирующей способности при наличии ошибок в более защищенных информационных битах. Рассмотрим эффект повышения корректирующей способности в зависимости от значения  $K$  на примере декодирования поступивших на вход декодера турбокода искаженных кодовых слов, приведенных в таблице 5.32. Такие кодовые слова получены путем введения ошибок в кодовое слово  $-11-1-1 \text{ } -1111 \text{ } -1-1-11$ . В таблице 5.32 ошибочные символы выделены подчеркиванием. Для проведения исследований воспользуемся реализацией декодера одного кодового слова ТССК в MS Excel, ранее описанной в п. 4.2.2.

Таблица 5.32

Кодовые слова при разных значениях  $K$ 

Значение $K$	Информационные биты				Проверочные биты							
					Блок 1				Блок 2			
$K = 1$	-1	1	-1	-1	-1	1	<u>-1</u>	<u>-1</u>	-1	-1	-1	<u>-1</u>
$K = 0,8$	-1	1	-1	-1	-0,8	0,8	<u>-0,8</u>	<u>-0,8</u>	-0,8	-0,8	-0,8	<u>-0,8</u>
$K = 0,7$	-1	1	-1	-1	-0,7	0,7	<u>-0,7</u>	<u>-0,7</u>	-0,7	-0,7	-0,7	<u>-0,7</u>
$K = 0,6$	-1	1	-1	-1	-0,6	0,6	<u>-0,6</u>	<u>-0,6</u>	-0,6	-0,6	-0,6	<u>-0,6</u>

В таблице 5.33 приведены значения мягкого выхода декодера турбокода, полученные в результате трех итераций декодирования кодового слова из таблицы 5.32. Проведя анализ результатов декодирования, можно сделать вывод о повышении корректирующей способности кода при снижении значения  $K$ .

Произведем исследование корректирующей способности ТДК при наличии в проверочных битах кодового слова от трех до семи ошибок для разных значений  $K$ .

Таблица 5.33

Мягкий выход декодера турбокода при разных значениях  $K$ 

Значения информационных бит		-1	1	-1	-1
Значения мягкого выхода декодера турбокода при различных значениях $K$	$K = 1$	-41,3	<u>-10,4</u>	-67,0	-66,8
	$K = 0,8$	-34,4	<u>-4,8</u>	-65,8	-65,8
	$K = 0,7$	-30,9	<u>-1,9</u>	-65,2	-63,6
	$K = 0,6$	-27,9	0,9	-64,6	-62,7

Рассмотрим возможности декодера по исправлению трех ошибок при их расположении в проверочных битах. Для удобства восприятия результатов (см. таблицу 5.34) выделим четыре случая расположения ошибочных бит в кодовом слове турбокода:

– 1 случай – все три ошибки располагаются в первом блоке проверочных бит (4 варианта расположения ошибок);

– 2 случай – все три ошибки располагаются во втором блоке проверочных бит (4 варианта расположения ошибок);

– 3 случай – одна ошибка в первом блоке проверочных бит, две – во втором (24 варианта расположения ошибок);

– 4 случай – одна ошибка во втором блоке проверочных бит, две – в первом (24 варианта расположения ошибок).

Таблица 5.34

## Количество вариантов исправления трех ошибок в кодовом слове

$K$	Случаи расположения ошибок				Всего вариантов исправления
	1 случай	2 случай	3 случай	4 случай	
1	3	4	24	19	50
0,8	3	4	24	21	52
0,6	4	4	24	24	56

Рассмотрим возможности декодера по исправлению четырех ошибок при их расположении в проверочных битах. Для удобства восприятия результатов (см. таблицу 5.35) выделим четыре случая расположения ошибочных бит:

- 1 случай – все три ошибки располагаются в одном блоке проверочных бит (2 варианта расположения ошибок);
- 2 случай – три ошибки в первом блоке проверочных бит, одна – во втором (16 вариантов расположения ошибок);
- 3 случай – три ошибки во втором блоке проверочных бит, одна – в первом (16 вариантов расположения ошибок);
- 4 случай – в каждом из блоков проверочных бит имеется по две ошибки (36 вариантов расположения ошибок).

Таблица 5.35

Количество вариантов исправления четырех ошибок в кодовом слове

K	Случаи расположения ошибок				Всего вариантов исправления
	1 случай	2 случай	3 случай	4 случай	
1	2	4	16	23	45
0,8	2	7	16	26	51
0,6	2	11	16	33	62

Рассмотрим возможности декодера по исправлению пяти ошибок при их расположении в проверочных битах. Для удобства восприятия результатов (см. таблицу 5.36) выделим четыре случая расположения ошибочных бит:

- 1 случай – четыре ошибки располагаются в первом блоке проверочных бит, одна – во втором (4 варианта расположения ошибок);
- 2 случай – четыре ошибки располагаются во втором блоке проверочных бит, одна – в первом (4 варианта расположения ошибок);
- 3 случай – три ошибки располагаются в первом блоке проверочных бит, две – во втором (24 варианта расположения ошибок);
- 4 случай – три ошибки располагаются во втором блоке проверочных бит, две – в первом (24 варианта расположения ошибок).

Таблица 5.36

Количество вариантов исправления пяти ошибок в кодовом слове

K	Случаи расположения ошибок				Всего вариантов исправления
	1 случай	2 случай	3 случай	4 случай	
1	2	4	2	10	18
0,8	2	4	6	11	23
0,6	2	4	16	19	41

Рассмотрим возможности декодера по исправлению шести ошибок при их расположении в проверочных битах. Для удобства восприятия результатов (см. таблицу 5.37) выделим четыре случая расположения ошибочных бит:

- 1 случай – четыре ошибки располагаются в первом блоке проверочных бит, две – во втором (6 вариантов расположения ошибок);
- 2 случай – четыре ошибки располагаются во втором блоке проверочных бит, две – в первом (6 вариантов расположения ошибок);
- 3 случай – три ошибки располагаются в первом блоке проверочных бит, три – во втором (16 вариантов расположения ошибок).

Таблица 5.37

Количество вариантов исправления шести ошибок в кодовом слове

K	Случаи расположения ошибок			Всего вариантов исправления
	1 случай	2 случай	3 случай	
1	1	1	2	4
0,8	1	2	4	7
0,6	3	6	6	15

Рассмотрим возможности декодера по исправлению семи ошибок при их расположении в проверочных битах. Для удобства восприятия результатов (см. таблицу 5.38) выделим два случая расположения ошибочных бит:

- 1 случай – четыре ошибки располагаются в первом блоке проверочных бит, три – во втором (4 варианта расположения ошибок);

– 2 случай – четыре ошибки располагаются во втором блоке проверочных бит, три – в первом (4 варианта расположения ошибок).

Таблица 5.38

Количество вариантов исправления семи ошибок в кодовом слове.

$K$	Случаи расположения ошибок		Всего вариантов исправления
	1 случай	2 случай	
1	1	0	1
0,8	2	0	2
0,6	2	2	4

По итогам рассмотрения результатов исследований, приведенных в таблицах 5.34 – 5.38, можно сделать вывод о значительном росте корректирующей способности ТДК при появлении ошибок в проверочных битах кодовой последовательности турбокода. Однако снижение значения  $K$  ухудшает корректирующую способность при наличии ошибок в информационных битах кодового слова. Определение наилучшего значения  $K$  проводилось эмпирическим путем.

С целью числовой оценки корректирующей способности ТДК, с использованием описанного в п. 5.4 программного имитатора, осуществлено нахождение вероятности появления ошибки в декодированном информационном сообщении для разных вариантов дополнительного кодирования информационных бит кодового слова ТДК при различных значениях  $K$ .

Результаты работы программного имитатора приведены в таблице 5.39. В таблице в скобках приведены конкретные значения  $K$ , для которых получены наилучшие результаты корректирующей способности. Применение повышающего коэффициента к значениям символов кодового слова турбокода, закодированных дополнительным кодом, приводит к росту корректирующей способности. Эффект от применения коэффициентов увеличивается при снижении вероятности ошибочного бита в декодируемой последовательности.

Корректирующая способность ТДК

$T_D$	$p_B=10^{-1}, p_V=3,1 \cdot 10^{-2}$		$p_B=10^{-2}, p_V=1,9 \cdot 10^{-5}$		$p_B=10^{-3}, p_V=2,3 \cdot 10^{-9}$	
	$K = 1$	$K > 1$	$K = 1$	$K > 1$	$K = 1$	$K > 1$
0	$3,1 \cdot 10^{-2}$	–	$1,3 \cdot 10^{-4}$	–	$8,5 \cdot 10^{-7}$	–
1	$2,32 \cdot 10^{-2}$	$2,23 \cdot 10^{-2}$ ( $K=1,2$ )	$7,31 \cdot 10^{-5}$	$4,18 \cdot 10^{-5}$ ( $K=3$ )	$3,9 \cdot 10^{-7}$	$2,0 \cdot 10^{-7}$ ( $K=3$ )
2	$1,55 \cdot 10^{-2}$	$1,49 \cdot 10^{-2}$ ( $K=1,1$ )	$1,34 \cdot 10^{-5}$	$7,80 \cdot 10^{-6}$ ( $K=2$ )	$1,1 \cdot 10^{-8}$	$2,35 \cdot 10^{-9}$ ( $K=3$ )
3	$1,20 \cdot 10^{-2}$	$1,12 \cdot 10^{-2}$ ( $K=1,1$ )	$8,01 \cdot 10^{-6}$	$2,9 \cdot 10^{-6}$ ( $K=2$ )	$4,0 \cdot 10^{-9}$	$7,58 \cdot 10^{-10}$ ( $K=3$ )

Для улучшения восприятия данные из таблицы 5.39 представлены в графическом виде на рисунке 5.13. Пунктиром приведены графики для  $K = 1$ .

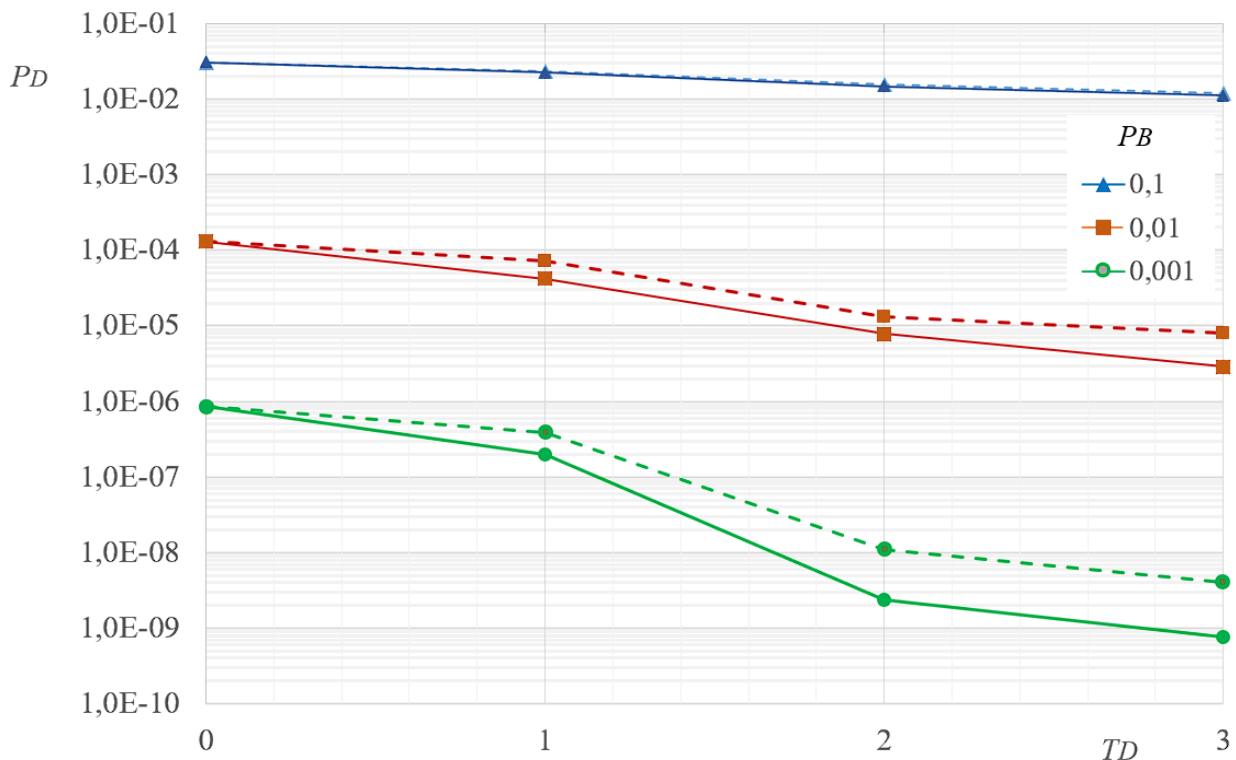


Рисунок 5.13. Вероятность появления ошибочного бита в декодированном информационном сообщении ТДК

В параграфе предложен способ повышения корректирующей способности комбинированного кода, рассмотренного в п. 5.4. С этой целью предложено

внесение изменений в метод декодирования по максимуму апостериорной вероятности при применении коэффициентов пропорциональных степени достоверности значений символов, то есть различных для кодовых бит турбокода, подвергшихся и не подвергшихся дополнительному кодированию. С целью исследования корректирующей способности модифицированного комбинированного кода разработан программный имитатор. Определены значения коэффициентов, обеспечивающие минимальную величину вероятности битовой ошибки в декодированном информационном сообщении. Выявлено, что эффект роста корректирующей способности увеличивается при снижении вероятности битовой ошибки в канале передачи данных. При дополнительном кодировании трех информационных бит и применении повышающего коэффициента к значениям дополнительно кодированных информационных бит снижение вероятности битовой ошибки в декодированном информационном сообщении составило: 1,1 раза (с  $1,2 \cdot 10^{-2}$  по  $1,1 \cdot 10^{-2}$ ) при  $p_B = 1 \cdot 10^{-1}$ , 2,8 раза (с  $8,01 \cdot 10^{-6}$  по  $2,9 \cdot 10^{-6}$ ) при  $p_B = 1 \cdot 10^{-2}$ , 5,3 раза (с  $4,0 \cdot 10^{-9}$  по  $7,6 \cdot 10^{-10}$ ) при  $p_B = 1 \cdot 10^{-3}$ . Числовые результаты получены по итогам работы разработанного программного имитатора.

### **Выводы по главе 5**

В главе рассмотрена возможность построения турбокода на основе СБК при декодировании в соответствии с критерием максимума апостериорной вероятности. Такой вариант построения ТСБК позволяет использовать более гибкие аппаратные реализации без усложнения алгоритма, осуществлять декодирование мягкого решения детектора и перфорацию. Описана модификация алгоритма MAP для реализации декодирования с использованием нерегулярной синдромной решетки блочного кода. Даны количественные оценки корректирующей способности ТСБК при декодировании согласно максимуму апостериорной вероятности. Для описанного турбокода осуществлен вывод формулы расчета вероятности битовой ошибки в декодированном



информационном сообщении. Продемонстрирована возможность гибкой адаптации параметров турбокода без изменения алгоритма декодирования при введении перфорации. При декодировании жесткого решения детектора, изменение кодовой скорости возможно от 0,3 до 0,5, то есть на 67 %. При декодировании мягкого решения детектора, изменение кодовой скорости возможно от 0,3 до 0,6, то есть на 100 %. Проведено сравнение корректирующей способности турбокодов на основе СБК и ССК при декодировании согласно максимуму апостериорной вероятности.

Путем теоретического анализа особенности декодирования кодового слова турбокода, а также экспериментальным путем выявлена высокая чувствительность декодера к достоверности значений информационных бит кодового слова. Негативной стороной такой чувствительности является высокая вероятность ошибки декодирования при наличии ошибок в информационных битах кодового слова. С целью ликвидации данного недостатка предложено снизить чувствительность декодера к значениям информационных бит. Описаны изменения метода декодирования турбокода по максимуму апостериорной вероятности путем применения понижающих коэффициентов для значений информационных бит. С использованием модифицированного алгоритма декодирования на языке программирования Python разработан программный имитатор исследования корректирующей способности турбокода. Анализ результатов исследования позволил сделать вывод о снижении вероятности битовой ошибки в декодированном информационном сообщении. Определены значения коэффициентов, обеспечивающие минимальную величину вероятности битовой ошибки в декодированном информационном сообщении. Выявлено, что эффект роста корректирующей способности увеличивается при снижении вероятности битовой ошибки в канале передачи данных. Для турбокода на основе сверточного кода при декодировании жесткого решения детектора снижение вероятности битовой ошибки в декодированном информационном сообщении составило: 1,1 раза (с  $3,1 \cdot 10^{-2}$  до  $2,82 \cdot 10^{-2}$ ) при  $p_B = 1 \cdot 10^{-1}$ , 3 раза (с  $1,3 \cdot 10^{-4}$  до  $4,3 \cdot 10^{-5}$ ) при  $p_B = 1 \cdot 10^{-2}$ , 14 раз (с  $8,5 \cdot 10^{-7}$  до  $6,1 \cdot 10^{-8}$ ) при  $p_B = 1 \cdot 10^{-3}$ .

Предложен еще один вариант решения проблемы высокой чувствительности декодера турбокода к достоверности значений информационных бит, путем построения комбинированного кода на основе турбокода с дополнительным кодированием информационных бит. Для исследуемого турбокода, при дополнительном кодировании лишь одного информационного бита, зафиксировано снижение вероятности появления ошибочного бита в декодированном информационном сообщении, в частности, в 1,34 раза при  $p_B = 1 \cdot 10^{-1}$ , в 1,78 раза при  $p_B = 1 \cdot 10^{-3}$ , в 2,18 раза при  $p_B = 1 \cdot 10^{-3}$ . Изменяя число бит, подвергшихся дополнительному кодированию, можно регулировать кодовую скорость и корректирующую способность комбинированного кода. Далее в главе приводится описание предлагаемого способа повышения корректирующей способности такого комбинированного кода. С этой целью предложено внесение изменений в метод декодирования по максимуму апостериорной вероятности при применении коэффициентов пропорциональных степени достоверности значений символов, то есть различных для кодовых бит турбокода, подвергшихся и не подвергшихся дополнительному кодированию. С целью исследования корректирующей способности модифицированного комбинированного кода разработан программный имитатор. Определены значения коэффициентов, обеспечивающие минимальную величину вероятности битовой ошибки в декодированном информационном сообщении. Выявлено, что эффект роста корректирующей способности увеличивается при снижении вероятности битовой ошибки в канале передачи данных. При дополнительном кодировании трех информационных бит и применении повышающего коэффициента к значениям дополнительно кодированных информационных бит получены следующие результаты снижения вероятности битовой ошибки в канале передачи данных: в 1,1 раза (с  $1,2 \cdot 10^{-2}$  по  $1,1 \cdot 10^{-2}$ ) при  $p_B = 1 \cdot 10^{-1}$ , в 2,8 раза (с  $8,01 \cdot 10^{-6}$  по  $2,9 \cdot 10^{-6}$ ) при  $p_B = 1 \cdot 10^{-2}$ , в 5,3 раза (с  $4,0 \cdot 10^{-9}$  по  $7,58 \cdot 10^{-10}$ ) при  $p_B = 1 \cdot 10^{-3}$ .

## ЗАКЛЮЧЕНИЕ

В диссертационном исследовании решена научная проблема повышения эффективности комбинированных помехоустойчивых кодов, имеющая важное теоретическое и прикладное значение. Получены следующие основные результаты:

1. По итогам анализа научных трудов и доступных источников в области исследования вопросов оценки эффективности определены следующие основные критерии эффективности помехоустойчивого кодирования:

- степень приближения характеристик кода к пределу К. Шеннона;
- достижение требуемой величины вероятности битовой ошибки;
- достижение требуемого значения кодовой скорости;
- снижение величины вероятности битовой ошибки при заданной избыточности кода;
- снижение избыточности кода для обеспечения требуемой вероятности битовой ошибки;
- снижение энергетических и временных затрат на кодирование-декодирование.

2. На основе анализа научных публикаций последних лет выделены основные современные подходы к повышению эффективности помехоустойчивых кодов: комбинирование кодов, декодирование мягкого решения детектора, перемежение бит и символов, адаптация параметров кода. Адаптация параметров кода чаще всего была реализована путем модификации порождающих и проверочных матриц блочных кодов и перфорацией кодовых последовательностей сверточных кодов.

3. С целью повышения адаптивных возможностей последовательного каскадного кода, предложен модифицированный метод последовательного каскадного кодирования, позволяющий изменять параметры кода, путем осуществления кодирования внутренним кодом только определенной части бит с выхода кодера внешнего кода. С учетом избыточности, вносимой при кодировании внутренним кодом, плавное изменение скорости передачи информации в

зависимости от состояния канала возможно до 48 % при применении кода Голея в качестве внутреннего кода, что соответствует энергетическому выигрышу до 1,7 дБ.

4. Осуществлено построение турбокода, на основе блочного кода, при декодировании в соответствии с критерием максимума апостериорной вероятности. Разработан модифицированный метод декодирования в соответствии с критерием максимума апостериорной вероятности, позволяющий осуществлять декодирование кодовой последовательности ТСБК. Предложенный вариант построения декодера позволяет без усложнения алгоритма осуществлять декодирование мягкого решения детектора и декодирование перфорированного кода. Продемонстрирована возможность гибкой адаптации параметров ТСБК без изменения алгоритма декодирования при введении перфорации. При декодировании жесткого решения детектора изменение кодовой скорости возможно на 67 %. При декодировании мягкого решения детектора изменение кодовой скорости возможно на 100 %.

5. Осуществлено построение комбинированного кода на основе сверточного и блочного кода при повторной передаче информационных бит и декодировании в соответствии с критерием максимума апостериорной вероятности. Для блочных и сверточных кодов при повторной передаче информационных бит кодового слова на приемной стороне возможно формирование нескольких кодовых слов и их декодирование с использованием модифицированного декодера турбокода. Таким образом, при отсутствии изменений в кодере блочного или сверточного кода, возможна адаптация характеристик кода к ухудшению состояния канала передачи данных. Для случая построения комбинированного кода на основе сверточного кода, при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ , вероятность битовой ошибки в декодированном информационном сообщении снизилась в 27 раз с  $4,0 \cdot 10^{-5}$  до  $1,5 \cdot 10^{-6}$ .

6. Реализовано построение комбинированного кода на основе турбокода с дополнительным кодированием информационных бит. Таким образом решена проблема высокой чувствительности декодера турбокода к достоверности

значений информационных бит кодового слова турбокода. Для исследуемого турбокода при дополнительном кодировании лишь одного информационного бита кодового слова турбокода зафиксировано снижение вероятности появления ошибочного бита в декодированном информационном сообщении в 2,18 раза при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ . Изменяя число бит, подвергшихся дополнительному кодированию, можно регулировать кодovou скорость и корректирующую способность комбинированного кода. При дополнительном кодировании трех информационных бит кодовая скорость изменилась с 0,33 до 0,27, а вероятность битовой ошибки в декодированном информационном сообщении снизилась в 212 раз. Для случая использования модуляции BPSK произведен расчет энергетического выигрыша от применения дополнительного кодирования информационных бит кодового слова турбокода, который составил 0,28 дБ – при дополнительном кодировании одного бита, 0,78 дБ – при дополнительном кодировании двух бит, 0,8 дБ – при дополнительном кодировании трех бит.

7. Предложена модификация метода декодирования по максимуму правдоподобия путем применения к значениям декодируемых символов коэффициентов, пропорциональных их достоверности. В методы декодирования внесены изменения, позволяющие реализовать умножение значений символов перед декодированием на коэффициенты. Применение модификации метода привело к росту корректирующей способности ГПКК.

Для ГПКК на основе внешнего сверточного кода получено снижение вероятности битовой ошибки в декодированном информационном сообщении в 920 раз – с  $1,1 \cdot 10^{-10}$  до  $1,2 \cdot 10^{-13}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ). Для ГПКК на основе внешнего блочного кода получено снижение вероятности битовой ошибки в декодированном информационном сообщении в 2,5 раза с  $4,7 \cdot 10^{-7}$  до  $1,9 \cdot 10^{-7}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 1 \cdot 10^{-3}$ ), что в 368 раз меньше снижения вероятности битовой ошибки при построении ГПКК на основе внешнего сверточного кода. Такое преимущество

ГПКК на основе внешнего сверточного кода обусловлено лучшей взаимосвязью бит кодового слова сверточного кода.

8. Предложена модификация метода декодирования в соответствии с критерием максимума апостериорной вероятности путем применения к значениям декодируемых символов коэффициентов.

Путем теоретического анализа особенности декодирования кодового слова турбокода, а также экспериментальным путем выявлена высокая чувствительность декодера к достоверности значений информационных бит кодового слова. Негативной стороной такой чувствительности является высокая вероятность ошибки декодирования при наличии ошибок в информационных битах кодового слова. Применение модифицированного метода декодирования с использованием понижающих коэффициентов для декодируемых значений информационных бит кодового слова позволило снизить вероятность битовой ошибки в декодированном информационном сообщении. Для турбокода на основе сверточного кода при декодировании жесткого решения детектора снижение вероятности битовой ошибки в декодированном информационном сообщении составило: 1,1 раза (с  $3,1 \cdot 10^{-2}$  до  $2,82 \cdot 10^{-2}$ ) при  $p_B = 1 \cdot 10^{-1}$ , 3 раза (с  $1,3 \cdot 10^{-4}$  до  $4,3 \cdot 10^{-5}$ ) при  $p_B = 1 \cdot 10^{-2}$ , 14 раз (с  $8,5 \cdot 10^{-7}$  до  $6,1 \cdot 10^{-8}$ ) при  $p_B = 1 \cdot 10^{-3}$ .

Проведя теоретический и экспериментальный анализ особенности декодирования кодового слова ТДК, определено, что учесть большую степень достоверности дополнительно закодированных информационных символов можно путем умножения их значений на повышающие коэффициенты  $K > 1$ , или путем умножения значений менее достоверных символов на понижающие коэффициента  $K < 1$ . Применение коэффициентов позволило снизить вероятность битовой ошибки в декодированном информационном сообщении. Определены значения коэффициентов, обеспечивающие минимальную величину вероятности битовой ошибки в декодированном информационном сообщении. При дополнительном кодировании трех информационных бит, снижение вероятности битовой ошибки в декодированном информационном сообщении составило: 1,1 раза (с  $1,2 \cdot 10^{-2}$  по

$1,1 \cdot 10^{-2}$ ) при  $p_B = 1 \cdot 10^{-1}$ , 2,8 раза (с  $8,01 \cdot 10^{-6}$  по  $2,9 \cdot 10^{-6}$ ) при  $p_B = 1 \cdot 10^{-2}$ , 5,3 раза (с  $4,0 \cdot 10^{-9}$  по  $7,6 \cdot 10^{-10}$ ) при  $p_B = 1 \cdot 10^{-3}$ .

При разностороннем анализе особенностей декодирования кодового слова КПИБ, выявлена высокая чувствительность декодера к достоверности значений проверочных бит кодового слова. Негативной стороной такой чувствительности является высокая вероятность ошибки декодирования при наличии ошибок в проверочных битах кодового слова. Осуществлено повышение корректирующей способности КПИБ путем снижения чувствительности декодера к значениям проверочных бит кодового слова. Применение модифицированного метода декодирования с использованием коэффициентов позволило снизить вероятность битовой ошибки в декодированном информационном сообщении. Для случая декодирования КПИБ на основе ССК, снижение составило 208 раз с  $2,5 \cdot 10^{-4}$  до  $1,2 \cdot 10^{-6}$  (при вероятности битовой ошибки в канале передачи данных  $p_B = 10^{-3}$ ).

9. С целью исследования корректирующей способности рассмотренных вариантов комбинированных кодов и исследования действенности предложений по модификации методов декодирования разработаны программные имитаторы кодеков комбинированных кодов на языках программирования Python и C++. На семь программных продуктов оформлены свидетельства о государственной регистрации программ для ЭВМ. Разработанные программные имитаторы могут быть использованы в научных и учебных целях.

Опыт внедрения в практическую деятельность разработанных кодовых конструкций, модифицированных методов декодирования, алгоритмов, воплощенных в программные продукты (см. Приложение 1) позволяет рекомендовать результаты диссертационного исследования к практическому использованию в системах цифровой связи.

Перспективы дальнейшей разработки темы автору видятся следующими.

1. Разработка адаптивных к изменениям характеристик канала передачи данных каскадных сигнально-кодовых конструкций.

2. Повышение эффективности кодов Галлагера.
3. Разработка комбинированных кодов с применением в их составе полярных кодов.



## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

БЧХ – Боуз-Чоудхури-Хоквенгем (код)

ГПКК – гибкий последовательный каскадный код

КПИБ – комбинированный код кодом с повтором информационных бит

МЧС – многочастотный сигнал

ПКК – последовательный каскадный код

ППРЧ – псевдослучайное перестроение рабочей частоты

СБК – систематический блочный код

ССК – систематический сверточный код

ТСБК – турбокод, созданный на основе систематического блочного кода

ТССК – турбокод, созданный на основе систематического сверточного кода

ТДК – турбокод с дополнительным кодированием информационных бит

МАР – алгоритм декодирования по максимуму апостериорной вероятности (от английского «maximum a posteriori»)

SOVA – алгоритм Витерби с мягким выходом (от английского «soft-output Viterbi algorithm»)

## СПИСОК ЛИТЕРАТУРЫ

1. Кудряшев Б.Д. Теория информации : учебник. – СПб. : Питер, 2009. – 320 с.
2. Блинова И.В., Попов И.Ю. Теория информации. Учебное пособие. – СПб : Университет ИТМО, 2018. – 84 с.
3. Гошин Е.В. Теория информации и кодирования : учеб. пособие. – Самара : Изд-во Самарского университета, 2018. – 124 с.
4. Джеймс Г. Информация. История. Поток. – М. : АСТ, Corpus, 2013. – 576 с.
5. Шеннон К. Математическая теория связи // Работы по теории информатики и кибернетике : пер. с англ ; под ред. Р.Л. Добрушина, О.В. Луканова. – М. : Иностранная литература, 1963. – 830 с.
6. Матвеев Б.В. Основы корректирующего кодирования. Теория и лабораторный практикум : учеб. пособие. – СПб. : Лань, 2014. – 192 с.
7. Голиков А.М. Модуляция, кодирование и моделирование в телекоммуникационных системах. Теория и практика : учеб. пособие для вузов / А.М. Голиков. – 3-е изд., стер. – СПб : Лань, 2022. – 452 с.
8. Котельников А.В. Теория потенциальной устойчивости. – М. : Радио и связь, 1998. – 152 с.
9. Аджемов А.С., Санников В. Г. Общая теория связи: учебник для вузов. – М.: Горячая линия – Телеком, 2023. – 624 с.
10. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. – М. : Техносфера, 2006. – 319 с.
11. Варгузин В.А., Цикин И.А. Методы повышения энергетической и спектральной эффективности цифровой радиосвязи. – СПб. : БХВ-Петербург, 2013. – 352 с.
12. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. – М. : Издательский дом «Вильямс», 2016. – 1104 с.
13. Лихобабин Е.А. Методы и алгоритмы декодирования кодов с низкой плотностью проверки на четность в системах цифрового телерадиовещания : дис. ... канд. техн. наук : 05.12.04. – Рязань., 2014. – 177 с.

14. Трифонов П.В. Методы построения и декодирования многочленных кодов : дис. ... д-ра техн. наук : 05.13.17. – СПб., 2018. – 254 с.
15. Акмалходжаев А.И. Разработка и исследование эффективных алгоритмов декодирования турбокодов в системах мобильной связи : дис. ... канд. техн. наук : 05.12.13. – СПб., 2015. – 126 с.
16. Синицин Д.В. Повышение помехоустойчивости радиотехнических систем передачи информации с использованием сверточных алгоритмов обработки сигналов : дис. ... канд. техн. наук : 05.12.04. – Владимир, 2014. – 127 с.
17. Карболин В.А. Исследование методов повышения помехоустойчивости короткоимпульсных сверхширокополосных систем радиосвязи при работе в каналах с многолучевым распространением : дис. ... канд. техн. наук : 2.2.15. – Новосибирск, 2023. – 188 с.
18. Тамразян Г.М. Программно-аппаратная реализация оптимального алгоритма декодирования каскадных кодов на базе кодов Рида-Соломона в адаптивных системах обмена данными : дис. ... канд. техн. наук : 05.12.13. – Ульяновск, 2017. – 142 с.
19. Чилихин М.Ю. Разработка и моделирование алгоритмов декодирования полярных кодов в системе информационно-управляющих комплексов : дис. ... канд. техн. наук : 05.12.13. – Ульяновск, 2015. – 149 с.
20. Демидов Д.С. Методы и алгоритмы повышения и исследования эффективности многопороговых декодеров помехоустойчивых кодов в высокодостоверных системах передачи информации : дис. ... канд. техн. наук : 05.12.04. – Рязань, 2016. – 141 с.
21. Санников В.Г. Теория информации и кодирования : учебн. пособие / Санников В.Г. – М. : Московский технический университет связи и информатики, 2015. – 95 с.
22. Возенкрафт Дж., Джекобс И. Теоретические основы техники связи. – М. : Мир, 1969. – 392 с.
23. Робинсон Ф.Н. Шумы и флуктуации в электронных схемах и цепях. – М. : Атомиздат, 1980. – 256 с.

24. Голд Б., Рейдер Ч. Цифровая обработка сигналов. – М. : Советское радио, 1973. – 367 с.
25. Финк Л.М. Сигналы, помехи, ошибки. – М. : Радио и связь, 1984. – 256 с.
26. Кловский Д.Д. Передача дискретных сообщений по радиоканалам. – М. : Связь, 1982. – 304 с.
27. Справочник по радиорелейной связи / под. ред. С.В. Бородича. – М. : Радио и связь, 1981. – 415 с.
28. Советов Б.Я. Помехоустойчивость передачи команд телеуправления в системе с запросом // Автоматика и телемеханика, 1966. – № 12. – с. 43–48.
29. Rappaport T.S. Wireless Communication (Principles and Practice). – N.Y. Prentice Hall, 1996. – 736 p.
30. Цифровая обработка сигналов / под ред. А.Б. Сергиенко – СПб. : Питер, 2003. – 604 с.
31. Феер К. Беспроводная цифровая связь. Методы модуляции и расширения спектра. Пер. с англ / под ред. В.И. Журавлева. – М. : Радио и связь, 2000. – 520 с.
32. Котов П.А. Повышение достоверности передачи цифровой информации. – М. : Связь, 1966. – 192 с.
33. Шлома А.М., Бакулин М.Г., Крейнделин В.Б., Шумов А.П. Новые алгоритмы формирования и обработки сигналов в системах подвижной связи / под ред. А.М. Шломы. – М. : Горячая линия – Телеком, 2008. – 344 с.
34. Левин Б.Р. Теоретические основы статистической радиотехники. Книга 1. – М. : Советское радио, 1974. – 552 с.
35. Сергиенко А.Б. Цифровая связь : учеб. пособие. – СПб. : Изд-во СПбГЭТУ «ЛЭТИ», 2012. – 164 с.
36. Борисов В.И., Зинчук В.М., Лимарев А.Е. Помехозащищенность систем радиосвязи с расширением спектра сигналов методом псевдослучайной перестройки рабочей частоты. – М. : РадиоСофт, 2008. – 512 с.
37. Студенникова Д.А. Основы телекоммуникаций : учебник. – М. : Академия, 2019. – 256 с.

38. Ekroot L., Dolinar S. A Decoding of Block Codes // IEEE Transaction Inform Theory, 1993. – P. 1052–1056.
39. Сидоренко А.А. Применение помехоустойчивого кодирования с исправлением ошибок при передаче цифровых сигналов // Материалы вторых Всероссийских Армандовских чтений. Сб. тезисов докладов научно-практического семинара. – Муром, 2012. – С. 36–37.
40. Berlekamp E.R., Algebraic Coding Theory, rev. ed., Aegean Park Press, 1984. – 461 p.
41. Виттерби А.Д., Омура Дж.К. Принципы цифровой связи и кодирования. – М. : Радио и связь, 1982. – 536 с.
42. Белов М.В., Новиков Д.А. Методология комплексной деятельности. – М. : Ленанд, 2018. – 320 с.
43. Большая советская энциклопедия / гл. ред. О.Ю. Шмидт. – М. : Советская энциклопедия, 1926–1947.
44. Оценка эффективности деятельности учреждений социальной поддержки населения / под ред. П.В. Романова и Е.Р. Ярской-Смирновой. Серия «Научные доклады: независимый экономический анализ». – М. : Московский общественный научный фонд ; Центр социальной политики и гендерных исследований, 2007. – 234 с.
45. Мачулин В.В., Пятибратов А.П. Эффективность систем обработки информации – М. : Советское радио, 1972. – 280 с.
46. Новиков Д.А. Методология управления. – М. : Либрокком, 2011. – 128 с.
47. Горохов В.Г. Методологический анализ системотехники. – М. : Радио и связь, 1982. – 325 с.
48. Ожегов С.И. Словарь русского языка. 21-е издание, переработанное и дополненное / под ред. чл. кор. АН СССР Н.Ю. Шведовой. – М. : Русский язык, 1989. – 921 с.
49. Ушаков Д.Н. Большой толковый словарь русского языка : современная редакция. – М. : Хит-книга, 2017. – 1553 с.
50. Форни Д. Каскадные коды. – М. : Мир, 1970. – 207 с.

51. Галлагер Р. Теория информации и надежная связь. – М. : Советское радио, 1974. – 568 с.
52. Блох Э.Л., Зяблов В.В. Обобщенные каскадные коды. – М. : Связь, 1976. – 240 с.
53. Пчелин Н.А., Гладких А.А. Повышение эффективности декодирования двоичных блочных избыточных кодов // Сборник трудов XXIII Международной научно-технической конференции. – Воронеж, 2017. – С. 370–380.
54. Егоров С.И., Борзов Д.Б., Дегтярев С.В. Повышение эффективности декодирования кодов Рида-Соломона по обобщенному минимальному расстоянию / Известия Юго-Западного государственного университета. 2018. – Т. 22, – № 3 (78). – С. 51–58.
55. Гладких А.А. Методы эффективного декодирования избыточных кодов и их современные приложения / А.А. Гладких, Р.В. Климов, Н.Ю. Чилихин. – Ульяновск : УлГТУ, 2016. – 258 с.
56. Трифонов П.В., Основы помехоустойчивого кодирования : учебн. пособие. – СПб. : Университет ИТМО, 2022. – 231 с.
57. Новиков Р.С. Анализ эффективности методов перемежения данных для помехоустойчивых кодов в каналах связи с помехами / Вестник науки и образования Северо-Запада России. – 2015, –Т. 1. –№ 2. – С. 1–7.
58. Баринов А.Ю., Асеев А.Ю. Модифицированная математическая модель системы генерирования перемеженной дискретной последовательности турбоподобного кода / Вестник Череповецкого государственного университета. – 2017, – № 6. – С. 9–17.
59. Калыгина Л.А., Галичев Е.В. Расчет параметров перемежителя для помехоустойчивых кодов / Проектирование и технология электронных средств. – 2021, –№ 1. – С. 41–44.
60. Тамразян Г.М. Современные методы адаптивного помехоустойчивого кодирования / Г.М. Тамразян // Автоматизация процессов управления. – 2016. – №2 (44). – С.45–49

61. Кудряшев Б.Д. Основы теории кодирования: учебное пособие. – СПб. : БХВ-Петербург, 2016. – 400 с.

62. Ганин Д.В., Гладких А.А., Шамин А.А., Шагарова А.А. Комплексный метод повышения энергетической и спектральной эффективности цифровой радиосвязи / Вестник НГИЭИ, 2016, – № 6, – С. 16–23.

63. Богачев И.В., Левенец А.В. Адаптивный выбор параметров помехоустойчивых кодов на основе оценки текущего состояния канала передачи данных // Материалы XI Международной научно-технической конференции с элементами научной школы и конкурсом научно-исследовательских работ для студентов, аспирантов и молодых ученых. – Пенза, 2019. – С. 19–22.

64. Сидоренко А.А. Адаптивное помехоустойчивое кодирование // Материалы X международной научно-технической конференции «Перспективные технологии в средствах передачи информации». – Владимир-Суздаль, 2013. – Т. 1. – С. 152–154.

65. Шахтанов С.В. Перестановочное декодирование недвоичных избыточных кодов / Вестник НГИЭИ. – 2017. – № 8 (75). – С. 7–13.

66. Gladkikh A.A., Ganin D.V., Pchelin N.A., Shakhtanov S.V., Ocherovsky A.V. Coding methods and permutation decoding in the systems for network processing of data / International Journal of Control and Automation. – 2020. – Т. 13. – № 1. – С. 93–110.

67. Илькевич В.А. Анализ эффективности сверточных кодов по верхней границе вероятности битовой ошибки / В.А. Илькевич // Доклады секции «Радиотехника и электроника»: 56-я научная конференция аспирантов, магистрантов и студентов БГУИР – 2020. – 152 с.

68. Илькевич В.А. Критерии поиска оптимальных сверточных кодов / В.А. Илькевич // Доклады секции «Радиотехника и электроника»: 54-я научная конференция аспирантов, магистрантов и студентов БГУИР – 2018. – 137 с.

69. Корнеева Н.Н., Никитин О.Р., Полушин П.А. Алгоритмы диагностики перфорированных кодов // Материалы докладов XXIII Международной научно-технической конференции, посвященной 100-летию НГТУ – Нижегородского политехнического института. – Нижний Новгород, 2017. – С. 1371–1375.

70. Корнеева Н.Н., Никитин О.Р. Алгоритм определения периода перфорации в процессе диагностики перфорированных кодов / Проектирование и технология электронных средств. – 2020, – № 2. – С. 54–57.

71. Ковалгин Ю.А. Цифровое радиовещание: системы и технологии : монография. – М. : Горячая линия – Телеком, 2021. – 580 с.

72. Barrett T. W. History of ultra wideband (UWB) radar & communications: Pioneers and innovators // In Proceedings of Progress in Electromagnetics Symposium 2000 (PIERS2000), July 2000.

73. Kalinin V., Karbolin V., Nosov V Features of Calculating the Noise Immunity of UWB Radiosystems in the INDOOR Mode // 2021 IEEE 22nd International Conference of Young Professionals in Electron Devices and Materials (EDM-2021). Souza, the Altai Republic, Russia, June 30-July 4, 2021. PP. 198-203, doi: 10.1109/EDM52169.2021.9507623.

74. Зюко А.Г., Кловский Д.Д., Назаров М.В., Финк Л.М. Теория передачи сигналов. – М. : Радио и связь, 1998. – 288 с.

75. Пятибратов А.П. Вычислительные системы, сети и телекоммуникации. – М.: Радио и связь, 1998. – 512 с.

76. Ратынский М.В. Основы сотовой связи. – М. : Радио и связь, 1998. – 392 с.

77. Повышение эффективности сверточных помехоустойчивых кодов с помощью конечных автоматов / И.А. Завадский // Управляющие системы и машины. – 2015. – № 1. – С. 25–31.

78. Мартышевская Д.А., Полушин П.А. Применение процедуры Витерби для борьбы с межсимвольной интерференцией / Наука и образование: проблемы, идеи, инновации. – 2015, – № 1. – С. 50–53.

79. Сайдахметов М., Оспанова Н., Балабатыров К. Исследование сверточного кодирования данных для проведения анализа эффективности помехоустойчивого кодирования / Вестник Казахской академии транспорта и коммуникаций. – 2020, № 1. – С. 316–320.



80. Горячкин О.В., Григоров И.В., Долгополов В.Н., Петров О.А., Суханов Д.В., Хабаров Е.О. Пути повышения эффективности бескабельных телеметрических забойных телесистем подземной связи / Инфокоммуникационные технологии – 2009, – № 4, – Т. 7, – С. 46–55.

81. Clark G.C. and Cain J.V. Error Correction Coding for Digital Communications. – Plenum Press, New York, 1981. – 144 p.

82. Коржик В.И., Финк Л.М. Помехоустойчивое кодирование дискретных сообщений в каналах со случайной структурой. – М. : Связь, 1979. – 272 с.

83. Банкет В.Л., Дорофеев В.М. Цифровые методы в спутниковой связи. – М. : Радио и связь, 1988. – 240 с.

84. Белоцерковский И.Л. Протоколы передачи файлов для модемов // Сети, 1995, – № 3 – С. 53–59.

85. Сидоренко А.А. Эффективность применения блочных и сверточных кодов для кодирования коротких информационных сообщений / Проектирование и технология электронных средств. – 2017, – № 4. – С. 31–36.

86. Яворски М., Зиаде Т. Python. Лучшие практики и инструменты. – СПб. : Питер, 2024. – 592 с.

87. Фёдоров Д.Ю. Программирование на языке высокого уровня Python. – М. : Юрайт, 2022. – 210 с.

88. David Beazley. Python Distilled. Sebastopol, O'Reilly Media, 2021. – 352 p.

89. Керниган Б.У., Ритчи Д.М. Язык программирования Си – М. : Вильямс. – 2013. – 251 с.

90. Стивен П. Язык программирования С. Лекции и упражнения. – М. : Вильямс, 2015. – 928 с.

91. Айвор Х. Visual C++ 2010 : полный курс. – М. : Диалектика, 2010. – 1216 с.

92. Страуструп Б. Программирование: принципы и практика использования C++. – М. : Вильямс, 2011. – 1248 с.

93. Сидоренко А.А. Разработка программной эмуляция мягкого решения на выходе детектора, для сравнительного исследования эффективности декодеров,

работающих с мягким и жестким решением детектора / Проектирование и технология электронных средств. – 2018, – № 3. – С. 44–49.

94. Нуньес-Иглесиас Х., Уолт Ш., Дэшноу Х. Элегантный SciPy. – ДМК Пресс, 2018. – 266 с.

95. Кольцов Д.М. Python. Полное руководство. – М. : Наука и техника, 2022. – 480 с.

96. Булинский А.В. Ширяев. А.Н. Теория случайных процессов. – М.: Физматлит, 2003.

97. Галкин А.П., Лапин А.Н., Самойлов А.Г. Моделирование каналов систем связи. – М.: Связь, 1979.

98. Коржик В.И., Финк Л.М., Щелкунов К.Н. Расчет помехоустойчивости систем передачи дискретных сообщений. – М. : Радио и связь, 1981. – 232 с.

99. Айвор Хортон. Visual C++ 2010 : полный курс. – М. : Диалектика, 2010. – 1206 с.

100. Сидоренко А.А. Механизм исправления множественных ошибок при декодировании сверточного кода / Радиотехнические и телекоммуникационные системы. – 2018, – № 2. 53–59.

101. Сидоренко А.А. Исправление множественных ошибок декодером сверточного кода // Перспективные технологии в средствах передачи информации: Материалы 12-ой международной научно – технической конференции. Том 1. – Владимир: ВлГУ. – 2017. – С. 156–159.

102. Сидоренко А.А. Сравнение эффективности декодирования мягкого и жесткого решения детектора на примере сверточного кода / Проектирование и технология электронных средств. – 2018, – № 2. – С. 51–56.

103. Сидоренко А.А. Исследование механизма декодирования перфорированных сверточных кодов / Проектирование и технология электронных средств. – 2019, – № 4. – С. 42–46.

104. Сидоренко А.А. Исследование эффективности декодирования мягких и жестких решений детектора в перфорированных сверточных кодах / Проектирование и технология электронных средств. – 2021, – № 3. – С. 31–36.

105. Сидоренко А.А. Адаптивные возможности блочных и сверточных кодов // Перспективные технологии в средствах передачи информации: Материалы 13-ой международной научно – технической конференции. Том 2. – Владимир : ВлГУ. – 2019. – С. 22–26.
106. Yasuda Y., Kashiki K. and Hirata Y., «High-Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding», IEEE Trans. Comm., vol. COM-32, no. 3, March 1984. – P. 325–328.
107. Chao Y.-L. Ultra-wideband Radios with Transmitted Reference Methods PhD thesis, University of Southern California, Los Angeles, CA, 2005. – 159 p.
108. Шахнович И.В. Современные технологии беспроводной связи: издание второе, исправленное и дополненное. – М. : Техносфера, 2006. – 288 с.
109. Носов В.И., Калинин В.О. Исследование методов повышения помехоустойчивости короткоимпульсных сверхширокополосных систем радиосвязи. Новосибирск, СибГУТИ, 2017. – 247 с.
110. Карболин В.А., Носов В.И. Исследование влияния скорости передачи и частоты дискретизации импульсной характеристики на помехоустойчивость КСШП-системы радиосвязи // Вестник СибГУТИ, 2018. – № 2. – С. 71–83.
111. Homier E.A. Synchronization of ultra-wideband signals in the dense multipath channel. PhD thesis, University of Southern California, Los Angeles, CA, 2004. – 220 p.
112. Карболин В.А., Носов В.И. Оценка влияния скорости передачи на помехоустойчивость КСШП системы радиосвязи при многолучевом распространении (INDOOR режим). // Современные проблемы телекоммуникаций: материалы Российской научно-технической конференции, Новосибирск, 26–27 апреля 2018 г. – Новосибирск : Изд-во СибГУТИ, 2018. – С. 337–341.
113. Перфилов О.Ю. Радиопомехи : учебн. пособие для вузов. –М. : Горячая линия – Телеком, 2016. – 110 с.
114. Sidorenko A.A. Method for changing the redundancy of sequential concatenated code // V International Conference on Applied Physics, Information

Technologies and Engineering – APITECH-V-2023, October, Uzbekistan, Bukhara. 2023.

115. Сидоренко А.А. Программный тренажер исследования эффективности последовательного каскадного кода, построенного на основе внешнего сверточного кода, при избирательном кодировании внутренним кодом // Свидетельство о государственной регистрации программы для ЭВМ. № 2023682349. 2023.

116. Золотарев В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы : Справочник / под ред. Зубарева Ю.Б. – М. : Горячая линия – Телеком, 2004. – 126 с.

117. Сидоренко А.А. Кодек для систем телеметрии, работающих в условиях сложной помеховой обстановки // Материалы XV Всероссийской научной конференции студентов-радиофизиков. – СПб., 2011. – С. 108–112.

118. Некоторые вопросы теории кодирования. Сборник переводов / под ред. Э.Л. Блоха и М.С. Пинскера. – М. : Мир, 1970. – 275 с.

119. Mayo-Wells The Origins of Space Telemetry, – Technology and Culture, 1963. – 344 p.

120. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. – М. : Мир, 1976. – 594 с.

121. Волков Л.Н., Немировский М.С., Шинаков Ю.С. Системы цифровой радиосвязи: базовые методы и характеристики : учебн. пособие. – М. : Эко-Трендз, 2005. – 390 с.

122. Коржик В.И., Финк Л.М. Помехоустойчивое кодирование дискретных сообщений в каналах со случайной структурой. – М. : Связь, 1976. – 272 с.

123. Злотник Б.М. Помехоустойчивые коды в системах связи. – М. : Радио и связь, 1989. – 232 с.

124. Сизоненко А.Б. Высокопроизводительная реализация линейных блочных кодов за счет выбора оптимальной полиномиальной формы представления / Сборник научных трудов SWORLD. – 2012, –№ 1. – С. 86–83.

125. Быков В.В. Помехоустойчивый блочный циклический код для цифровых телевизионных систем // X Международная отраслевая научно-техническая конференция : Сборник трудов. – М., 2016. – С. 88.
126. Алишери А. Модельное исследование блочного кода Хэмминга / Сборник избранных статей научной сессии ТУСУР. – 2018, – № 1. – С. 116–118.
127. Сорока Н.И., Кривинченко Г.А. Телемеханика : конспект лекций для студентов специальности «Автоматическое управление в технических системах». Ч. I : Сообщения и сигналы. – Мн. : БГУИР, 2000. – 127 с.
128. Бородин Л.Ф. Введение в теорию помехоустойчивого кодирования. – М.: Советское радио, 1968. – 408 с.
129. Советов Б.Я. Сравнительная оценка надежности передачи информации избыточными кодами // Автоматика и телемеханика, 1970, – № 2. – С. 41–44.
130. Мак-Вильямс Ф.Дж., Слоэн Н.Дж.А. Теория кодов, исправляющих ошибки. – М. : Связь, 1979. – 744 с.
131. Вернер М. Основы кодирования. – М. : Техносфера, 2006. – 286 с.
132. Блейхут Р. Теория и практика кодов, контролирующих ошибки. – М. : Мир, 1986. – 576 с.
133. Берлекэмп Э. Алгебраическая теория кодирования. – М. : Мир, 1971. – 477 с.
134. Кларк Дж. мл., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи. – М. : Радио и связь, 1987. – 391 с.
135. Прокис Д. Цифровая связь / под. ред. Д.Д. Кловского. – М. : Радио и связь, 2000. – 800 с.
136. Громаков Ю.А. Стандарты и системы подвижной радиосвязи. – М. : Эко-Трендз, 1998. – 239 с.
137. Massey J.L. «Shift Register and BCH Decoding», IEEE Trans. Info. Theory, vol. IT-15, no. 1, P. 122–127.
138. Касперский К. Техника защиты компакт-дисков от копирования. – СПб. : БХВ-Петербург, 2004. – 464 с.

139. Боуз Р.Ч., Рой-Чоудхури Д.К. Об одном классе двоичных групповых кодов с исправлением ошибок / Кибернетический сборник. – М., 1961. – Вып. 2.– С. 83–94.
140. Рид-Соломон Полиномиальные коды над некоторыми конечными полями / Кибернетический сборник. – М., 1963. – Вып. 7.
141. Чеботарев Н.Г. Основы теории Галуа. Часть 1. – М. : Едиториал УРСС, 2004. – 219 с.
142. Сидоренко А.А. Построение кодов, исправляющих ошибки с использованием арифметики полей Галуа // II международная заочная научно-техническая конференция «Информационные технологии. Радиоэлектроника. Телекоммуникации (ITRT-2012)». – Тольятти, 2012. – Ч. 3. – С. 230–236.
143. Каганцов С.М., Полушин П.А., Самойлов А.Г., Самойлов С.А. Кодек для цифровых радиорелейных станций // Пятая Российская конференция по атмосферному электричеству. Том 2, изд. «Транзит ИКС», Владимир. 2003. – С. 148–149.
144. Осмоловский С.А. Сравнительный анализ некоторых свойств стохастических кодов и кодов Рида-Соломона // Электросвязь, 1991. – № 1.
145. Сидоренко А.А. Использование арифметики полей Галуа при построении кодов Рида-Соломона / Проектирование и технология электронных средств. – 2011, – № 3. – С. 13–17.
146. Сидоренко А.А. Энергетический выигрыш от применения помехоустойчивого кодирования / Проектирование и технология электронных средств. – 2016, – № 4. – С. 8–14.
147. Гмурман В.Е. Теория вероятностей и математическая статистика : учебн. пособие. – М. : Высшее образование, 2006. – 479 с.
148. Вентцель Е.С. Теория вероятностей : учебник. – М. : Высшая школа, 2006. – 575 с.
149. Гнеденко Б.В. Курс теории вероятностей. – М. : Едиториал УРСС, 2005. – 448 с.
150. Баврин И.И. Высшая математика. – М. : Наука, 2000. – 611 с.

151. Сидоренко А.А. Анализ эффективности кодов Рида-Соломона в борьбе с независимыми ошибками и пакетами ошибок // 21 Всероссийская межвузовская научно-техническая конференция студентов и аспирантов «Микроэлектроника и информатика-2014». – М., 2014. – С. 196.

152. Sidorenko A.A. Overview of the III International Conference on Advanced Technologies in Materials Science, Mechanical and Automation Engineering – MIP: Engineering-III – 2021 // AIP Conf. Proc. 2021. Vol. 2402 (1). P. 050038(1-7). doi:10.1063/5.0071456.

153. Сидоренко А.А. Программный тренажер исследования эффективности каскадного кодирования при применении внешнего кода Рида-Соломона // Свидетельство о государственной регистрации программы для ЭВМ № 2023668233. 2023.

154. Сидоренко А.А. Построение универсального алгоритма Берлекемпа-Месси для кодов Боуза-Чоудхури-Хоквенгема // Материалы X международной научно-технической конференции «Перспективные технологии в средствах передачи информации». – Владимир-Суздаль, 2013. – Т. 1. – С. 155–157.

155. Майстренко В.А., Седунов Д.П. Сравнительный анализ декодирования каскадных укороченных кодов Рида-Соломона в адаптивных системах обмена данными с обычным декодированием / Техника радиосвязи, 2019. – № 3. – С. 60–68.

156. Самойлов А.Г., Сидоренко А.А. Применения кодов РС в каскаде с двоичными кодами с целью повышения эффективности борьбы с независимыми ошибками / Проектирование и технология электронных средств, 2014. – № 3. – С. 2–7.

157. Комаров В.Н. Об исправлении многократных пакетов ошибок / Техника средств связи, 1980. – Вып. 3.

158. Мелентьев О.Г. Теоретические аспекты передачи данных по каналам с группирующимися ошибками / под ред. В.П. Шувалова. – М. : Горячая линия – Телеком, 2007. – 232 с.

159. Бородич С.В. Искажения и помехи в многоканальных системах радиосвязи с частотной модуляцией. – М. : Связь, 1976. – 256 с.
160. Альшрайдех А.М., Гомес Ж.Л., Самойлов С.А., Сидоренко А.А. Исследование «мягкого» декодирования кода Рида-Соломона / Проектирование и технология электронных средств, 2014. – № 1. – С. 8–11.
161. Samoilo S. Reed – Solomon Codes Communication System // 5 – th Workshop Digital Broadcasting, Erlangen, Germany, 2004. – P. 85–86.
162. Самойлов С.А., Даниленко А.М. Оптимизированный алгоритм «мягкого» декодирования кода Рида-Соломона // Сборник трудов международной научно-технической и научно-методической конференции «Современные технологии в науке и образовании – СТНО-2016». – Рязань, 2016. – Т. 1. – С. 89–92.
163. Сидоренко А.А. Мягкое декодирование блочных кодов с использованием синдромной решетки // Перспективные технологии в средствах передачи информации: Материалы 14-ой международной научно – технической конференции. – Владимир : ВлГУ. – 2021. – С. 184–187.
164. Клюев Л.Л. Теория электрической связи: учебник. – М. : НИЦ ИНФРА-М, 2019. – 448 с.
165. Сидоренко А.А. Метод изменения избыточности последовательного каскадного кода, построенного на основе внешнего блочного кода / Информационно-измерительные и управляющие системы, 2024. – № 2.
166. Сидоренко А. А. Программный имитатор функционирования кодера-декодера последовательного каскадного кода на основе внешнего кода Хемминга, при избирательном кодировании внутренним кодом // Свидетельство о государственной регистрации программы для ЭВМ. № 20247255124. 2024.
167. Berrou C., Glavieux A, Thitimajshima P. Near Shannon Limit Error-Correcting Coding and Decoding: «Turbo-Codes», Proceedings of International Conference on Communications, Geneva, Switzerland, 1993. – P. 1064–1070.
168. Красносельский И.Н. Турбокоды : принципы и перспективы // Электросвязь, 2001. – № 1. – С. 17–20.



169. Панышо С.П., Югай В.В. Турбокодирование / Успехи современной радиоэлектроники, 2004. – № 2. – С. 3–16.
170. Ziemer R. and Peterson R. Introduction to Digital Communication, 2 ded., Prentice Hall, 2001. – 345 p.
171. Григорьев А.С., Дронов А.Е. Турбокодирование в системах однонаправленной передачи цифровой информации // Материалы 12 межрегиональной конференции «Обработка сигналов в системах телефонной связи и вещания», – Пушкинские горы – М. : МТУСИ, 2003. – 103 с.
172. Галкин В.А. Цифровая мобильная радиосвязь : учебн. пособие. – М. : Горячая линия – Телеком, 2021. – 518 с.
173. Волков А.А., Карпова Г.В., Журавлев О.Е. Повышение помехоустойчивости радиосвязи / А.А. Волков, Г.В. Карпова, О.Е. Журавлев // Мир транспорта, 2012. – №3. – С. 31–33.
174. Андрианов В.И., Соколов А.В. Сотовые, пейджинговые и спутниковые средства связи. – СПб. : БХВ-Петербург, 2001. – 400 с.
175. Телекоммуникационные системы и сети, Т. 2. – Радиосвязь, радиовещание и телевидение / под ред. В.П. Шувалова. – М. : Горячая линия – Телеком, 2004. – 672 с.
176. Sidorenko A.A. Overview of the International Conference on Applied Physics, Information Technologies and Engineering – APITECH – 2019 // Journal of Physics: Conference Series. 2019. Vol. 1399(1). P. 033011(1-8). doi:10.1088/1742-6596/1399/3/033011.
177. Marvin K. Simon, Mohamed-Slim Alouini. Digital Communication over Fading Channels: A Unified Approach to Performance Analysis 1st Edition. John Wiley and Sons, 2000. – 544 p.
178. Bertoni H.L. Radio Propagation for Modern Wireless Systems. Prentice hall, New Jersey, 2001. – 340 p.
179. Зяблов В.В., Цветков М.А. Метод обнаружения ошибочного декодирования с использованием списков / Информационные процессы, 2004. – № 2. – Том 4. – С. 188–201.

180. Санников В.Г. Теория информации и кодирования : учебн. пособие / Санников В.Г. – М. : Московский технический университет связи и информатики, 2015. – 95 с.

181. Proakis J.G., Salehi M. Digital communications. 5th ed., McGraw-Hill, New York, 2008. – 1150 p.

182. Березкин Е.Ф. Основы теории кодирования: учебное пособие. – СПб. : Издательство «Лань», 2019. – 320 с.

183. Sidorenko A.A. Overview of the International Conference on Applied Physics, Information Technologies and Engineering – APITECH III 2021//Journal of Physics: Conference Series. 2021. Vol. 2094(1). P. 032061(1-9) doi:10.1088/1742-6596/2094/3/032061.

184. Сидоренко А.А. Программный тренажер исследования эффективности турбокода с изменяемыми коэффициентами значимости декодируемых символов // Свидетельство о государственной регистрации программы для ЭВМ № 2023619022. 2023.

185. Сидоренко А.А. Исследование эффективности декодирования блочного и сверточного кода с использованием декодера турбокода / Проектирование и технология электронных средств, 2023, –№ 4. – С. 44–49.

186. Sidorenko A.A. Study of the block and convolutional code decoding efficiency using a turbo code decoder // V International Conference on Applied Physics, Information Technologies and Engineering – APITECH-V-2023, October, Uzbekistan, Bukhara. 2023.

187. Сидоренко А.А. Программный комплекс исследования эффективности комбинированного кода, построенного на основе сверточного кода при повторной передаче информационных бит и декодировании декодером турбокода // Свидетельство о государственной регистрации программы для ЭВМ № 2024611219. 2024.

188. Сидоренко А.А. Программный тренажер исследования эффективности упрощенного турбокода, построенного на основе кода Хемминга при повторной

передаче информационных бит // Свидетельство о государственной регистрации программы для ЭВМ. № 2023664344. 2023.

189. Sungsik Y. Byungkyu A. Jun H. An advanced low-complexity decoding algorithm for turbo product codes based on the syndrome // EURASIP Journal on Wireless Communications and Networking 2020, Article number: (2020) 2020:126.

190. Каданцев С.М., Казак П.А., Корж Н.С., Реализация алгоритма итеративного декодирования Чейза-Пиндиаха для блочных турбокодов на основе кодов Хэмминга / Вестник РГРТУ, 2019. – № 70. – С. 98–106.

191. Кузнецов В.А. Ускоренный расчет скалярных произведений для второго алгоритма Чейза при декодировании блочных турбокодов на основе паритетного кода // Международная научная конференция Обработка, передача и защита информации в компьютерных системах : Сборник докладов. Санкт-Петербург, 2021. – С. 233–237.

192. Кузнецов В.А. Векторизация алгоритма декодирования блочного турбокода произведения путем полного перебора кодовых слов // Международная научная конференция Обработка, передача и защита информации в компьютерных системах : Сборник докладов. СПб., 2022. – С. 317–324.

193. Воробьев К.А., Аджемов С.С. Сравнение и анализ характеристик блочных и сверточных турбокодов / T-Comm, 2011. – № 11. – С. 17–19.

194. Sidorenko A.A. Overview of the International Conference on Applied Physics, Information Technologies and Engineering – APITECH-IV - 2022 // Journal of Physics: Conference Series. 2022. Vol. 2388(1). P. 012029 (1-7). doi:10.1088/1742-6596/2388/1/012029.

195. Сидоренко А.А. Декодирование турбокода, созданного на основе блочного кода, с использованием синдромной решетки / Проектирование и технология электронных средств, 2022. – № 4. – С. 46–52.

196. Сидоренко А.А. Сравнение эффективности декодирования турбокодов, созданных на основе сверточных и блочных кодов / Проектирование и технология электронных средств, 2023, – № 1. – С. 3–9.

197. Сидоренко А.А. Оценка эффективности турбокода, построенного на основе блочного кода, при декодировании в соответствии с критерием максимума апостериорной вероятности / Системы управления, связи и безопасности, 2023. – № 3. – С. 29–43.

198. Сидоренко А. А. Программный тренажер исследования эффективности турбокода, построенного на основе кода Хемминга // Свидетельство о государственной регистрации программы для ЭВМ. № 2023614197. 2023.

199. Sidorenko A.A. Overview of the International Conference on Applied Physics, Information Technologies and Engineering - APITECH-IV - 2022 // Journal of Physics: Conference Series. 2022. Vol. 2388(1). P. 012032 (1-9). doi:10.1088/1742-6596/2388/1/012032.

200. Сидоренко А.А. Построение комбинированного кода на основе турбокода с дополнительным кодированием бит / Научные технологии в космических исследованиях Земли, 2024. – № 1. – С. 53–58.

# Приложение 1. Акты использования результатов диссертационного исследования

УТВЕРЖДАЮ



Генеральный директор  
компании «Стилсофт»

Ю.П. Стоянов

декабря 2023 г.

## АКТ

Об использовании результатов диссертационного исследования  
Сидоренко Александра Анатольевича на тему «Повышение эффективности  
комбинированных помехоустойчивых кодов», подготовленного на соискание  
ученой степени доктора технических наук по специальности 2.2.15 –  
Системы, сети и устройства телекоммуникаций

Настоящим актом подтверждаем, что результаты диссертационного исследования Сидоренко А.А. на тему «Повышение эффективности комбинированных помехоустойчивых кодов» обладают актуальностью, научной новизной и представляют практический интерес. Теоретические положения диссертационного исследования Сидоренко А.А. в части, касающейся направлений повышения эффективности комбинированных помехоустойчивых кодов, используются научными подразделениями компании. Фрагменты алгоритмов кодирования-декодирования, описанные в диссертации, использовались при разработке каналов управления беспилотными летательными аппаратами квадрокоптерного типа.

Технический директор

А.В. Шипулин

Главный конструктор

Ю.П. Стоянов



УТВЕРЖДАЮ



Генеральный директор  
ООО «Аргун-Софт»  
О.Б. Неилко

«...» января 2024 г.

## АКТ

внедрения результатов диссертационного исследования  
Сидоренко Александра Анатольевича  
«Повышение эффективности комбинированных помехоустойчивых кодов»  
подготовленного на соискание ученой степени доктора технических наук по  
специальности 2.2.15 (системы, сети и устройства телекоммуникаций)

Настоящий акт удостоверяет то, что результаты, полученные Сидоренко А.А. в диссертационном исследовании на тему «Повышение эффективности комбинированных помехоустойчивых кодов», а именно:

- предложенный вариант построения комбинированного кода, на основе блочного или сверточного кода при повторной передаче информационных бит кодового слова;
- предложения по повышению корректирующей способности комбинированных помехоустойчивых кодов путем модификации методов декодирования по максимуму правдоподобия и максимизации апостериорной вероятности, с использованием коэффициентов, пропорциональных достоверности декодируемых символов;
- предложенная модификация метода последовательного кодирования путем реализации избирательного кодирования кодовых бит внутренним кодом;

использованы при проектировании и разработке каналов связи автоматизированных систем управления, в том числе в рамках СЧ ОКР «Магистраль-МПСН/АС по заказу ФГУП «РНИИРС». Использование предложенных в диссертации научно-технических решений позволило повысить устойчивость передачи данных.

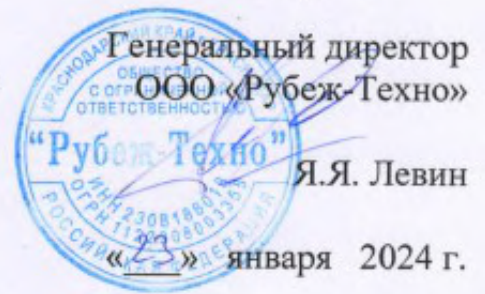
Заместитель генерального директора  
По разработке информационного обеспечения

Д.В. Агеев

Главный специалист  
отдела системного проектирования

В.Д. Богомолов

УТВЕРЖДАЮ

Генеральный директор  
ООО «Рубеж-Техно»

Я.Я. Левин

января 2024 г.

## АКТ

об использовании результатов, полученных в ходе диссертационного исследования Сидоренко Александра Анатольевича  
«Повышение эффективности комбинированных помехоустойчивых кодов»  
на соискание ученой степени доктора технических наук

Актом удостоверяется, что результаты диссертационной работы Сидоренко А.А. «Повышение эффективности комбинированных помехоустойчивых кодов» (специальности 2.2.15 – системы, сети и устройства телекоммуникаций) были внедрены в научно-исследовательской и производственной деятельности ООО «Рубеж-Техно» (г. Краснодар). В частности, использованы:

- предложенный вариант построения турбокода на основе блочного кода с декодированием по максимуму апостериорной вероятности;
- комбинированный код на основе турбокода с дополнительным кодированием информационных бит.

Внедрение результатов диссертационного исследования Сидоренко А.А. позволило повысить помехоустойчивость радиоканалов передачи данных между компонентами охранных и сигнализационных систем.

Начальник Технического центра

С.С. Варава



Экз. № 7

УТВЕРЖДАЮ

Генеральный директор  
АО «МВП «СВЕМЕЛ»А.В. Неверко  
января 2024 г.

## АКТ

об использовании результатов диссертационной работы  
Сидоренко Александра Анатольевича  
«Повышение эффективности комбинированных помехоустойчивых кодов»

Настоящим удостоверяем, что результаты диссертационной работы Сидоренко А.А. «Повышение эффективности комбинированных помехоустойчивых кодов», подготовленной на соискание ученой степени доктора технических наук по специальности 2.2.15 (системы, сети и устройства телекоммуникаций), обладают научной новизной и практической актуальностью. Наибольший интерес с точки зрения использования при создании автоматизированных систем у АО «МВП «СВЕМЕЛ» вызвали предложенные автором новые конструкции комбинированных кодов.

Элементы описанных в исследовании Сидоренко А.А. алгоритмов, обеспечивающих повышение корректирующей способности помехоустойчивых кодов путем применения коэффициентов к декодируемым битам, использованы при разработке протоколов передачи данных в автоматизированной системе в защищенном исполнении в ходе ОКР «Рубеж-Арктика», выполняемой по заказу ФГКУ «Войсковая часть 43753».

Директор по созданию автоматизированных систем  
в защищенном исполнении  
лауреат премии Правительства РФ в области науки и техники

 С.К. Гузь

Начальник аналитического отдела  
к.п.н, доцент  
лауреат премии Правительства РФ в области науки и техники

 К.С. Лахтионов