

Федеральное агентство связи  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Сибирский государственный университет  
телекоммуникаций и информатики»  
ФГБОУ ВО «СибГУТИ»

**В.И. Носов**

## **МОДЕЛИРОВАНИЕ СИСТЕМ СВЯЗИ В MATLAB SIMULINK**

Учебное пособие

Новосибирск 2019 г.

**В.И. Носов Моделирование систем связи в среде MATLAB SIMULINK:**  
Учебное пособие/СибГУТИ. – г. Новосибирск, 2019 г. – 158 стр.

В учебном пособии излагаются:

- полосовая модуляция и демодуляция: критерии линейного кодирования сигналов; методы цифровой полосовой модуляции; сигналы MPSK, M-QAM и их векторное представление; замирания вследствие многолучевого распространения; реальные АЧХ и ФЧХ канала связи;
- помехоустойчивое кодирование в системах телекоммуникаций: основные понятия и определения теории кодирования; выявление ошибок; блочные коды с коррекцией ошибок; сверточные коды; скремблирование и чередование блоков;
- принципы компьютерного моделирования в среде MATLAB : описание исследуемой модели; особенности моделирования в SIMULINK;
- исследование влияния вида модуляции сигнала M-QAM и M-ФМ на коэффициент битовых ошибок и полосу, занимаемую модулированным сигналом;
- исследование влияния способов кодирования сигнала на помехоустойчивость канала радиосвязи.

Кафедра систем радиосвязи

Ил. 86, табл. 11, список лит. - 14 наименов.

Рецензенты: профессор В.П. Разинкин  
Профессор А.И. Фалько

Для студентов высших учебных заведений, обучающихся по направлению подготовки 11.03.02 – Инфокоммуникационные технологии и системы связи квалификации (степени) «бакалавр»

Утверждено редакционно-издательским советом СибГУТИ в качестве учебного пособия

© Сибирский государственный университет  
телекоммуникаций и информатики, 2019 г.

## СОДЕРЖАНИЕ

	стр
СПИСОК СОКРАЩЕНИЙ .....	6
ВВЕДЕНИЕ.....	8
1 Полосовая модуляция и демодуляция.....	11
1.1 Критерии линейного кодирования сигналов.....	11
1.2 Методы цифровой полосовой модуляции.....	14
1.3 Оценка качества приема в цифровых системах связи.....	16
1.4 Фазовая манипуляция PSK.....	17
1.5 Вероятность ошибки при когерентном обнаружении сигнала BPSK.....	18
1.6 Дифференциальная фазовая манипуляция DPSK.....	22
1.7 Сигналы MPSK и их векторное представление.....	24
1.8 Квадратурная амплитудная модуляция M-QAM.....	27
1.9 Замирания вследствие многолучевого распространения.....	28
Контрольные вопросы.....	31
2 Помехоустойчивое кодирование в системах телекоммуникаций.....	32
2.1 Роль и место помехоустойчивого кодирования в системах телекоммуникаций.....	32
2.2 Дискретный канал связи.....	34
2.3 Основные понятия и определения теории кодирования.....	35
2.4 Кодирование и защита от ошибок.....	36
2.4.1 Циклическая проверка четности с избыточностью.....	39
2.4.2 Блочные коды с коррекцией ошибок.....	43
2.4.3 Сверточные коды.....	61
2.4.4 Чередование битов.....	69
2.4.5 Скремблирование.....	72
Контрольные вопросы.....	74
3 Принципы компьютерного моделирования в среде MATLAB.....	75
3.1 История возникновения системы MATLAB.....	75
3.2 SIMULINK – общие сведения.....	77
3.3 Описание исследуемой модели.....	78
3.4 Особенности моделирования в SIMULINK.....	80
3.5 Генератор псевдослучайной двоичной последовательности.....	82
3.6 Модуляторы .....	83
3.6.1 Узкополосный модулятор M-PSK (M-PSK Modulator Baseband).....	83
3.6.2 Узкополосный модулятор M-DPSK (M-DPSK Modulator Baseband).....	85
3.6.3 Узкополосный модулятор M-QAM (Rectangular QAM Modulator Baseband).....	86
3.7 Цифровые фильтры.....	88

3.8	Имитаторы каналов .....	91
3.8.1	Канал с аддитивным белым Гауссовским шумом (AWGN Channel).....	91
3.8.2	Канал с замираниями Райса (Rician Fading Channel).....	92
3.8.3	Канал с замираниями Релея (Multipath Rayleigh Fading Channel).....	93
3.9	Канальное кодирование.....	94
3.9.1	Обнаружение и исправление ошибок (Error Detection and Correction).....	94
3.9.2	Перемежение (Interleaving).....	99
3.9.3	Скремблирование (Scrambling).....	102
3.10	Устройства графического отображения.....	103
3.10.1	Осциллограф (Scope).....	103
3.10.2	Анализатор спектра (Spectrum Analyzer).....	104
3.10.3	Вектограф (Constellation diagram).....	106
3.11	Устройства регистрации и подсчета ошибок (Error Rate Calculation).....	108
3.12	Анализатор результатов моделирования (Bit Error Rate Analysis).....	110
	Контрольные вопросы.....	114
4	Исследование влияния модуляции сигнала на коэффициент битовых ошибок и полосу для модуляций М-КАМ и М-ФМ.....	115
4.1	Цель исследования.....	115
4.2	Сравнение помехоустойчивости разных видов модуляции в условиях белого Гауссовского шума.....	115
4.2.1	Сравнение М-PSK и М-DPSK на примере BPSK и BDPSK.....	118
4.2.2	Сравнение М-PSK и М-QAM.....	120
4.3	Исследование эффективности использования частотного спектра при различных видах модуляции.....	121
4.3.1	Производительность.....	121
4.3.2	Плоскость "полоса-эффективность".....	122
4.3.3	Эффективность использования полосы при выборе схем MPSK и MQAM.....	124
4.4	Оценка влияния замираний на модулированный сигнал.....	124
	Контрольные вопросы.....	130
5	Исследование влияния способов кодирования сигнала на помехоустойчивость канала радиосвязи.....	131
5.1	Спецификация различных типов кодеров.....	131
5.2	Результаты исследования.....	133
	Контрольные вопросы.....	138
6	Задание и выполнение работы.....	139
6.1	Задание на работу.....	139

6.2	Выполнение работы.....	140
6.2.1	Основные сведения для работы с моделью.....	142
6.2.2	Порядок выполнения работы.....	142
6.3	Пример выполнения работы.....	144
	ЗАКЛЮЧЕНИЕ.....	154
	СПИСОК ЛИТЕРАТУРЫ.....	157

## СПИСОК СОКРАЩЕНИЙ

ACI (adjacent channel interference) – неприемлемых помех соседних каналов

ASK (amplitude shift keying) – амплитудная манипуляция со сдвигом

ARQ (automatic repeat request) – протоколы с автоматическим запросом повторной передачи

AWGN (additive white Gauss noise) – аддитивный белый Гауссовский шум

BER (bit error rate) – частота битовых ошибок

BPSK (binary phase shift keying) – двоичная фазовая манипуляция

CPM (continios phase modulation) – модуляция буз разрыва связи

CRC (cyclic redundancy check) – циклическая проверка четности с избыточностью

DPSK (difference phase shift keying) – дифференциальная двоичная фазовая манипуляция

DS (dopler shift) – доплеровский сдвиг

FCS (frame check sequence) – контрольная последовательность кадра

FDM (frequency division multiplexing) – уплотнение с частотным мультиплексированием

FEC (forward error correction) – коды с коррекцией ошибок

FSK (frequency shift keying) – частотная манипуляция со сдвигом

IEEE (The Institute of Electrical and Electronics Engineers, Inc., произносится как "ай-трипл-и") – Институт инженеров по электротехнике и радиоэлектронике, ИИЭР (США)

LFSR (linear feedback shift register) – линейных регистров сдвига с обратной связью

MSK (minimum shift keying) – манипуляция с минимальным сдвигом

NRZ (non-return zero) code – код без возвращения к нулю

OQPSK (offset quadrature phase shift keying) – офсетная квадратурная фазовая манипуляция

PSK (phase shift keying) – фазовая манипуляция

ROF (rolloff factor) – коэффициент сглаживания

RS codes (Reed-Solomon codes) – коды Рида-Соломона

QAPSK (quadrature amplitude-phase shift keying) – квадратурная амплитудно-фазовая модуляция

SEC-DED (single-error-correcting, double-error-detecting) – код исправления 1-битовых и обнаружения 2-битовых ошибок

SNR (signal to noise ratio) – отношение сигнал/шум

QAM (quadrature amplitude modulation) – квадратурная амплитудная модуляция

АБГШ – аддитивный белый Гауссовский шум

БЧХ – коды Боуза-Чоудхури-Хоквенгема

ДКС – дискретный канал связи

КАМ – квадратурная амплитудная модуляция

ОФМ – относительная фазовая модуляция

СПДС – система передачи дискретных сообщений

СКМ – системы компьютерной математики

ФАПЧ – фазовая автоподстройка частоты

ЦСРС – цифровая система радиосвязи

## ВВЕДЕНИЕ

Телекоммуникации как часть инфраструктуры общества служат одним из источников обеспечения функционирования и подъема экономики государства.

На рубеже XXI века телекоммуникации оказались одной из наиболее быстро развивающихся частей общества. При этом темпы развития мировой телекоммуникационной отрасли в 2 раза превышают темпы роста всей мировой экономики. В настоящее время объем доходов отрасли превышает триллион долларов, причем 70% этой суммы приходится на услуги, а 30% - на продажу оборудования связи.

Наступивший XXI век, по мнению большинства аналитиков многих стран, станет веком перехода от индустриального общества к информационному, в основе которого будет лежать информационная инфраструктура, во многом определяемая создаваемыми беспроводными телекоммуникационными системами.

Одной из наиболее быстро развивающихся отраслей связи сегодня является мобильная связь. Стремление обеспечить связь с любым абонентом, где бы он ни находился, является естественным и понятным. В настоящее время это становится возможным благодаря внедрению мобильных сетей, использующих наземные и спутниковые системы радиосвязи.

Современный период развития сотовой связи характеризуется повсеместным распространением систем сотовой связи четвертого поколения, и созданием первых фрагментов сетей пятого поколения.

Сети радиодоступа предназначены для беспроводного подключения абонентов к телефонной сети общего пользования. Они находят все большее применение, являясь в ряде случаев единственно разумным вариантом присоединения абонента к ТФОП.

Известно, что организация доступа составляет 60% затрат на инфраструктуру сети связи. Технология доступа определяет величину доходов и прибыли оператора. При ее выборе необходимо учитывать:

- требуемую для организации доступа ширину спектра частот;
- соотношение цены и качества;
- скорость установки и наладки;

Современные сети и системы передачи информации охватывают все стороны деятельности человека. В системах связи стали передавать цифровые сигналы с более высокими требованиями к достоверности передаваемой информации, а удовлетворить этим требованиям совершенствованием линий связи, увеличением излучаемой мощности, снижением собственного шума приемных устройств зачастую оказывается экономически невыгодно или же просто невозможно.



В соответствии с теоремой Шеннона, для дискретных каналов связи (ДКС) с помехами, информация в таких каналах может передаваться со сколь угодно высокой степенью достоверности при условии, что скорость передачи информации не превышает пропускной способности канала. Помехоустойчивость (достоверность), как известно, определяется способностью системы противостоять вредному влиянию помех или, другими словами, это способность системы связи функционировать с требуемым качеством передачи информации в условиях неблагоприятных внутренних и внешних воздействий. Количественной мерой помехоустойчивости при передаче дискретных сообщений, как правило, является вероятность ошибки в принимаемой последовательности.

Основным средством обеспечения высокой помехоустойчивости сложной системы, при этом, является введение избыточности, необходимой для обнаружения и исправления ошибок, возникающих при работе системы и ее элементов. Теоретической базой эффективного использования вводимой избыточности является теория помехоустойчивого кодирования.

Помехоустойчивые (или корректирующие) коды имеют различные приложения. Так эти коды используются для защиты данных в вычислительных устройствах и сетях, системах хранения данных, спутниковых и навигационных системах, в сотовых телефонах и пейджерах, коды могут быть использованы для получения надежной связи даже тогда, когда мощность принимаемого сигнала близка к мощности шумов. В специальных приложениях коды используются для защиты информации против преднамеренно организованных помех.

В данной работе проводится исследование влияния способов кодирования сигнала на помехоустойчивость канала радиосвязи при изменении его характеристик.

Исследование базируется на блочной модели системы цифровой радиосвязи, реализованной с помощью программного пакета MATLAB. Пакет MATLAB является универсальной средой для моделирования любых систем, в том числе и телекоммуникационных, и предоставляет пользователю уникальную возможность экспериментально исследовать интересующую его систему без использования какого-либо оборудования, кроме персонального компьютера. Метод блочного моделирования, примененный в исследовании, позволяет не описывать математически всю модель целиком, используя языки программирования, а составить её, используя уже заложенные в программе SIMULINK (расширение программного пакета MATLAB) блоки, варьируя настройки которых, можно добиться нужных параметров системы.

В исследовании используется упрощенная модель цифровой системы радиосвязи (ЦРС), включающая в себя: генератор сигнала, кодер канала, модулятор, модель канала, демодулятор, декодер, а также ряд вспомогательных блоков.

Система MATLAB является научным инструментом высокой степени сложности. В связи с этим любое исследование, проводимое в среде MATLAB, требует не только превосходного знания предмета исследования, но и профессиональных навыков моделирования систем с помощью предлагаемых инструментов, что заметно сказывается на временных и трудозатратах при создании рабочей модели. На сегодняшний день в литературе нет достаточно подробных исследований моделей MATLAB SIMULINK для систем радиосвязи.

# 1 Полосовая модуляция и демодуляция

## 1.1 Критерии линейного кодирования сигналов

Прежде всего, имеет смысл дать определения некоторых терминов. Цифровой сигнал – это последовательность дискретных импульсов напряжения. Каждый импульс представляет собой сигнальную посылку. Для передачи двоичных данных каждый бит данных кодируется сигнальной посылкой. В простейшем случае существует взаимно однозначное соответствие между битами и сигнальными посылками (рисунок 1.1). Здесь двоичный «0» представляется более низким уровнем напряжения, а двоичная «1» – более высоким. Подобным образом цифровой поток данных можно закодировать в аналоговый сигнал как последовательность сигнальных посылок, причем каждая сигнальная посылка – это импульс постоянной частоты, фазы и амплитуды, соответствующий определенному двоичному числу. Для этого должно существовать взаимно однозначное соответствие между элементами данных (битами) и аналоговыми сигнальными посылками.

Скорость передачи данных  $R$  – это количество переданных бит за секунду. Длительностью или длиной бита называется время, затрачиваемое передатчиком на излучение этого бита; при скорости передачи данных  $R$  длина бита равна  $T_b = 1/R$  сек.

Цифровой сигнал в большинстве случаев передаётся по линиям связи не в двоичной форме, а после определённых преобразований, производимых с целью улучшения помехоустойчивости и спектральных характеристик, такие преобразования называют *линейным кодированием* или *модуляцией* [1, 14].

Модуляция – это процесс, посредством которого символы сообщений или каналные символы (если используется канальное кодирование) преобразуются в сигналы, совместимые с требованиями, налагаемыми каналом передачи данных. *Скоростью модуляции* называется скорость изменения уровня сигнала. Скорость модуляции измеряется в бодах, 1 бод соответствует одной сигнальной посылке в одну секунду. В случае полосовой модуляции (passband modulation) импульсы заданной формы модулируют синусоиду, называемую несущей частотой (carrier frequency), или просто несущей (carrier), для радиопередачи на нужное расстояние [2, 12].

Полосовая модуляция имеет и другие важные преимущества при передаче сигналов. При использовании одного канала более чем одним сигналом, модуляция может применяться для разделения различных сигналов. Один из таких методов – уплотнение с частотным разделением (frequency-division multiplexing – FDM). Модуляция может использоваться и

для минимизации последствий интерференции. Класс схем модуляции, известный как модуляция расширенным спектром, требует полосы, значительно превышающей минимальную полосу, необходимую для передачи сообщения. Кроме того, модуляция может использоваться для перемещения сигнала в диапазон частот, в котором легко удовлетворяются специфические конструктивные требования, например, относящиеся к фильтрации и усилению. Примером такого применения модуляции является преобразование в приемнике радиочастотных сигналов в сигналы промежуточной частоты.

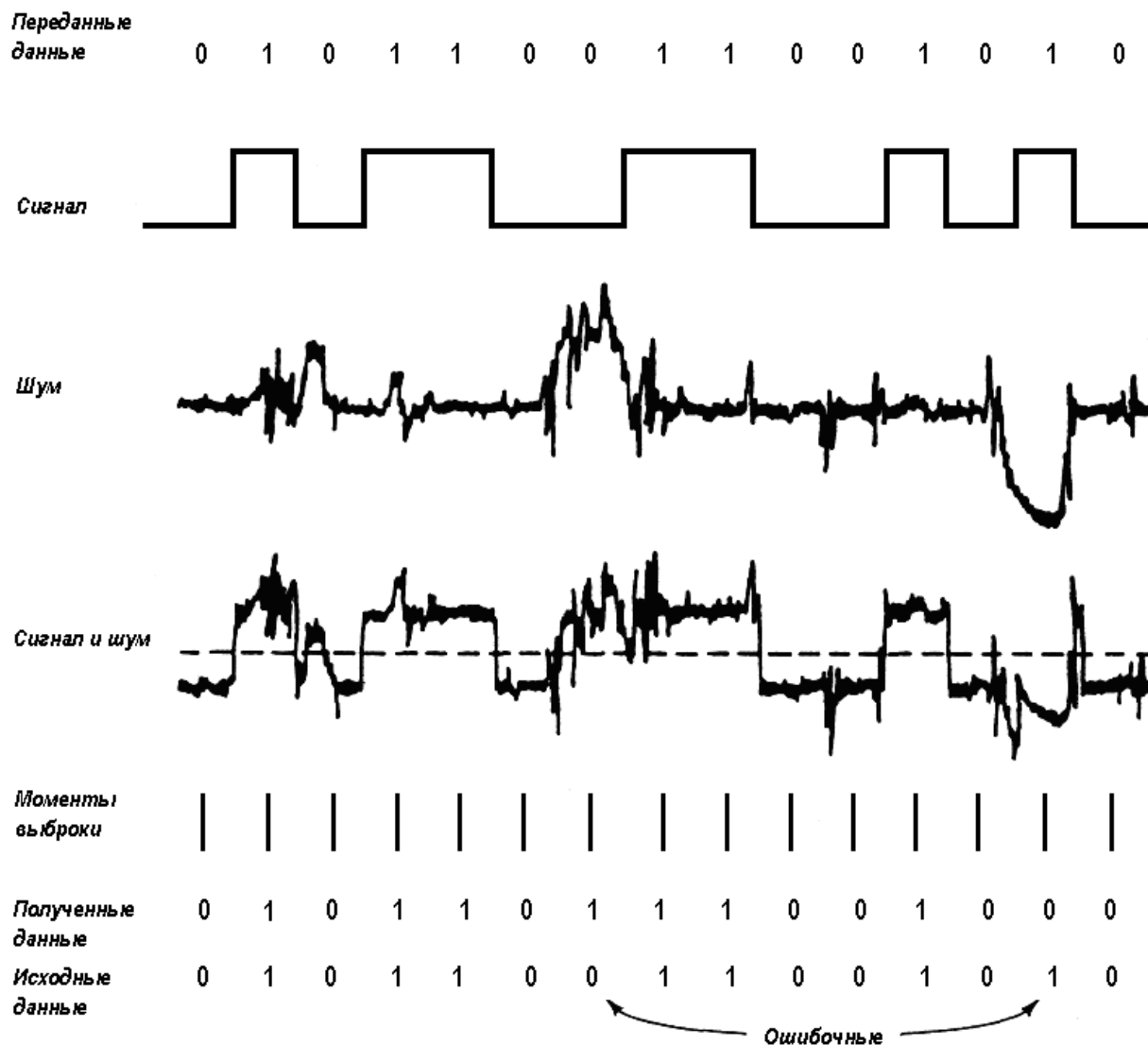


Рисунок 1.1 – Воздействие шума на цифровой сигнал

Задачи, входящие в интерпретацию цифровых сигналов в приемнике, можно обобщить, снова обратившись к рисунку 1.1. Во-первых, приемник должен знать распределение сигналов во времени. Иначе говоря, приемник должен, с определенной точностью, знать, где бит начинается и где –

заканчивается. Во-вторых, приемник должен определить уровень каждого двоичного разряда: высокий он «1» или низкий «0». На рисунке 1.1 для выполнения этих задач в середине интервала каждого двоичного разряда производится выборка, после чего полученная величина сравнивается с пороговым значением. Как можно видеть на том же рисунке, помехи и другие искажения могут приводить к ошибочным результатам.

То, насколько успешной будет интерпретация сигнала, поступившего в приемник, главным образом зависит от трёх факторов: отношения сигнал/шум, скорости передачи данных и ширины полосы. При фиксированных остальных параметрах справедливы следующие утверждения:

- увеличение скорости передачи данных приводит к увеличению скорости появления ошибочных битов, или, что тоже самое, к увеличению частоты битовых ошибок (BER – bit error rate);
- увеличение отношения сигнал/шум уменьшает скорость появления ошибочных битов;
- увеличение ширины полосы позволяет увеличить скорость передачи данных.

Существует еще один фактор, позволяющий увеличить производительность – выбор схемы линейного кодирования. Схема кодирования представляет собой простое отображение информационных битов в сигнальные посылки. При выборе конкретной схемы в первую очередь обращают внимание на следующие параметры и показатели [3, 4, 12].

- **Спектр сигнала.** Некоторые особенности спектра сигнала имеют важное значение. Отсутствие высокочастотных компонентов означает, что для передачи требуется более узкая полоса. Кроме того, желательно также, чтобы отсутствовала постоянная составляющая. Если в сигнале она присутствует, то должно существовать прямое физическое соединение передающего и приемного устройства (если речь идет о проводных системах связи). Если же постоянная составляющая отсутствует, то можно использовать соединение по переменному току через трансформатор; это дает превосходную электрическую изоляцию и снижает интерференцию. И, наконец, уровни искажения сигнала и интерференции зависят от спектральных свойств передаваемого сигнала. На практике наибольшие искажения сигнал претерпевает на краях полосы пропускания. Следовательно, при хорошей структуре сигнала основная передаваемая мощность должна быть сосредоточена в середине полосы передачи. Тогда принимаемый сигнал будет менее искаженным. Этого можно добиться с помощью методов модуляции, позволяющих формировать спектр передаваемого сигнала.

- Синхронизация. Приемник должен уметь определять начало и конец каждого двоичного разряда. Это совсем не легко. Один из довольно дорогостоящих подходов состоит в выделении специального канала для синхронизации приемника и передатчика. Альтернативный вариант – создание некоторого механизма синхронизации на базе переданного сигнала.

- Интерференция сигналов и помехоустойчивость. Отдельные кодировки позволяют добиваться лучшей производительности при наличии помех, чем другие. Выражается это, как правило, через скорость появления ошибок.

- Стоимость и сложность. Хотя цифровые логические схемы продолжают падать в цене, этот фактор игнорировать не стоит. В частности, чем выше скорость передачи сигналов, позволяющая получить определенную скорость передачи данных, тем выше цена.

## 1.2 Методы цифровой полосовой модуляции

Полосовая модуляция [3, 4, 12] (аналоговая или цифровая) – это процесс преобразования информационного сигнала в синусоидальный сигнал; при цифровой модуляции синусоида на интервале  $T$  называется цифровым символом. Синусоиды могут отличаться по амплитуде, частоте и фазе. Таким образом, полосовую модуляцию можно определить как процесс варьирования амплитуды, частоты или фазы (или их комбинаций) радиочастотной несущей согласно передаваемой информации. В общем виде несущая записывается следующим образом

$$s(t) = A(t) \cos \theta(t) \quad (1.1)$$

Здесь  $A(t)$  – переменная во времени амплитуда, а  $\theta(t)$  – переменный во времени угол.

Угол удобно записывать в виде

$$\theta(t) = \omega_0 t + \varphi(t) \quad (1.2)$$

так что

$$s(t) = A(t) \cos[\omega_0 t + \varphi(t)] \quad (1.3)$$

где  $\omega$ , – угловая частота несущей,  $\varphi(t)$  – ее фаза.

Частота может записываться как переменная  $f$  или как переменная  $\omega$ . В первом случае частота измеряется в герцах (Гц), во втором – в радианах в секунду (рад/с). Эти параметры связаны следующим соотношением  $\omega = 2\pi f$ .

Таблица 1.1 – Основные типы полосовой модуляции/демодуляции

Когерентные схемы	Некогерентные схемы
<ul style="list-style-type: none"> <li>• Фазовая манипуляция (PSK)</li> <li>• Частотная манипуляция (FSK)</li> <li>• Амплитудная манипуляция (ASK)</li> <li>• Модуляция без разрыва фазы (CPM)</li> <li>• Смешанные комбинации</li> </ul>	<ul style="list-style-type: none"> <li>• Дифференциальная фазовая манипуляция (DPSK)</li> <li>• Частотная манипуляция (FSK)</li> <li>• Амплитудная манипуляция (ASK)</li> <li>• Модуляция без разрыва фазы (CPM)</li> <li>• Смешанные комбинации</li> </ul>

Если для детектирования сигналов приемник использует информацию о фазе несущей, процесс называется когерентным детектированием (coherent detection); если подобная информация не используется, процесс именуется некогерентным детектированием (noncoherent detection). При идеальном когерентном детектировании приемник содержит прототипы каждого возможного сигнала. Эти сигналы-прототипы дублируют алфавит переданных сигналов по всем параметрам, даже по радиочастотной фазе. В этом случае говорят, что приемник автоматически подстраивается под фазу входного сигнала. В процессе демодуляции приемник перемножает и интегрирует входной сигнал с каждым прототипом (определяет корреляцию). В таблице 1.1 под общим заголовком когерентной модуляции/демодуляции перечислены: фазовая манипуляция, частотная манипуляция (frequency shift keying – FSK), амплитудная манипуляция (amplitude shift keying – ASK), модуляция без разрыва фазы (continuous phase modulation – CPM) и смешанные комбинации этих модуляций. Некоторые специализированные форматы, такие как квадратурная фазовая манипуляция со сдвигом (offset quadrature PSK – OQPSK), манипуляция с минимальным сдвигом (minimum shift keying – MSK), принадлежащие к классу модуляций CPM, и квадратурная амплитудная модуляция (quadrature amplitude modulation – QAM) в таблице не отображены.

Некогерентная демодуляция относится к системам, использующим демодуляторы, спроектированные для работы без знания абсолютной величины фазы входного сигнала; следовательно, определение фазы в этом случае не требуется. Таким образом, преимуществом некогерентных систем перед когерентными является простота, а недостатком – большая вероятность ошибки. В таблице 1.1 под заголовком некогерентной передачи сигналов перечислены модуляции, подобные используемым при когерентной

передаче: DPSK, FSK, ASK, CPM и смешанные их комбинации. Подразумевается, что для некогерентного приема информация о фазе не используется; но, тем не менее, под заголовком "некогерентная передача" указана одна из форм фазовой манипуляции. Это вызвано тем, что одну из важных форм PSK можно отнести к некогерентной (или дифференциально когерентной), поскольку она не требует согласования по фазе с принятой несущей. При использовании этой модуляции, называемой дифференциальной фазовой манипуляцией (differential PSK – DPSK), в процессе детектирования текущего символа в качестве опорной фазы применяется фаза предыдущего символа [2, 12].

### 1.3 Оценка качества приема в цифровых системах связи

Любой, кто изучал аналоговую связь, знаком с критерием качества, именуемым отношением средней мощности сигнала к средней мощности шума ( $S/N$  или  $SNR$ ). В цифровой связи для оценки качества чаще используется нормированная версия  $SNR$  – отношение  $E_b/N_0$ .  $E_b$  – это энергия бита, и ее можно описать как мощность сигнала  $S$ , умноженную на время передачи бита  $T_b$ .  $N_0$  – это спектральная плотность мощности шума, и ее можно выразить как мощность шума  $N$ , деленную на ширину полосы  $W$ . Поскольку время передачи бита и скорость передачи битов  $R_b$  взаимно обратны,  $T_b$  можно заменить на  $1/R_b$ .

$$\frac{E_b}{N_0} = \frac{S \cdot T_b}{N/W} = \frac{S/R_b}{N/W} \quad (1.4)$$

Еще одним параметром, часто используемым в цифровой связи, является скорость передачи данных в битах в секунду. В целях упрощения выражений, встречающихся в данной работе, для представления скорости передачи битов вместо записи  $R_b$  используется знак  $R$ . С учетом сказанного перепишем, выражение (1.4) так, чтобы было явно видно, что отношение  $E_b/N_0$  представляет собой отношение  $S/N$ , нормированное на ширину полосы и скорость передачи битов.

$$\frac{E_b}{N_0} = \frac{S}{N} \left( \frac{W}{R} \right) \quad (1.5)$$



Одной из важнейших метрик производительности в системах цифровой связи является график зависимости вероятности появления ошибочного бита  $P_b$  от  $E_b/N_0$ . На рисунке 1.2 показан общий вид большинства подобных кривых. При  $E_b/N_0 \geq x_0$ ,  $P_b \leq P_0$ . Безразмерное отношение  $E_b/N_0$  – это стандартная качественная мера производительности систем цифровой связи. Следовательно, необходимое отношение  $E_b/N_0$  можно рассматривать как метрику, позволяющую сравнивать производительность различных систем; чем меньше требуемое отношение  $E_b/N_0$ , тем эффективнее процесс регистрации при данной вероятности ошибки.

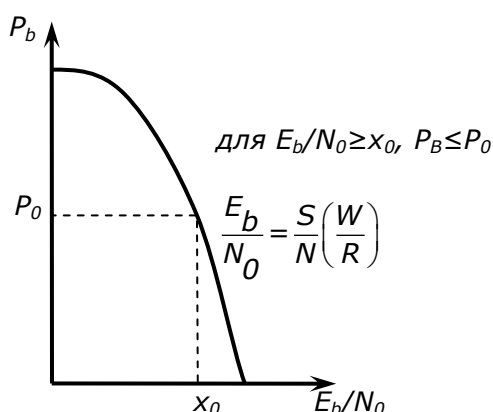


Рисунок 1.2 – Общий вид зависимости  $P_b$  от  $E_b/N_0$

#### 1.4 Фазовая манипуляция PSK

Фазовая манипуляция (phase shift keying – PSK) была разработана в начале развития программы исследования дальнего космоса; сейчас схема PSK широко используется в коммерческих и военных системах связи. Сигнал в модуляции PSK имеет следующий вид.

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos[\omega_0 t + \phi_i(t)] \quad 0 \leq t \leq T; i = 1, \dots, M, \quad (1.8)$$

где:  $E$  – энергия символа;

$T$  – время передачи символа,  $0 \leq t \leq T$ ;

$\omega_0$  – угловая частота;

$\phi_i(t)$  – фазовый член, который может принимать  $M$  дискретных значений, обычно определяемых следующим образом

$$\varphi_i(t) = \frac{2\pi i}{M} \quad i = 1, \dots, M \quad (1.9)$$

Типичный вид сигнала в модуляции BPSK (binary PSK – BPSK) приведен на рисунке 1.3, где явно видны характерные резкие изменения фазы при переходе между символами; если модулирующий поток данных состоит из чередующихся нулей и единиц, то такие изменения будут происходить при каждом переходе. Работа схемы модуляции заключается в смещении фазы модулируемого сигнала  $s_i(t)$  на одно из двух значений, нуль или  $\pi$  ( $180^\circ$ ).

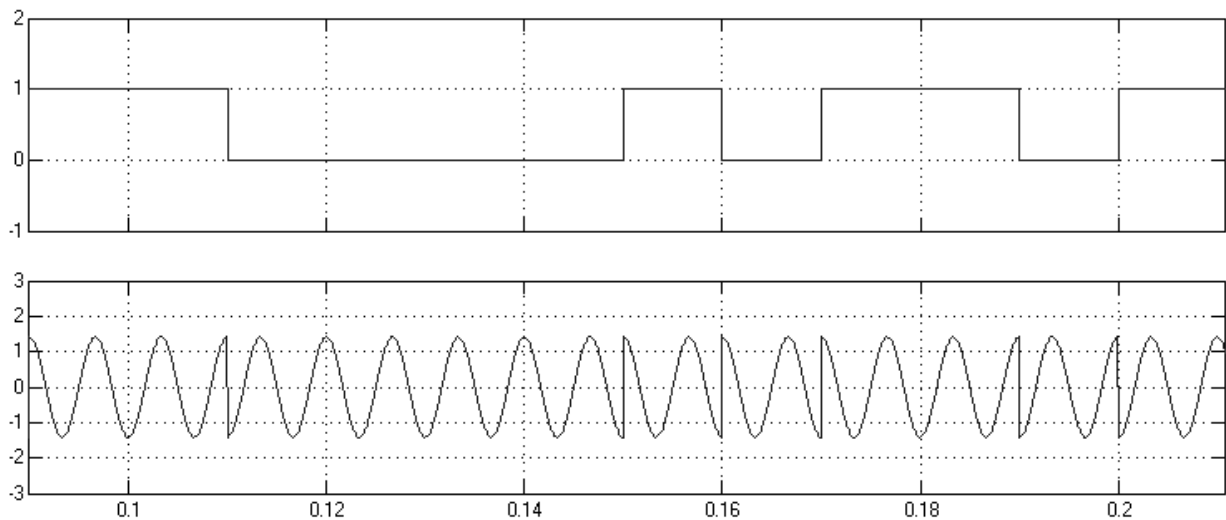


Рисунок 1.3 – Типичный вид сигнала BPSK

Модулированный сигнал можно представить как вектор на графике в полярной системе координат, где длина вектора соответствует амплитуде сигнала, а его ориентация в общем  $M$ -арном случае – фазе сигнала относительно других  $M-1$  сигналов набора. При модуляции BPSK векторное представление дает два противофазных ( $180^\circ$ ) вектора. Наборы сигналов, которые могут быть представлены подобными противофазными векторами, называется антиподными [3, 12].

### 1.5 Вероятность ошибки при когерентном обнаружении сигнала BPSK

Двоичное детектирование. Допустим, что передаваемый сигнал искажается только вследствие воздействия шума (additive white Gaussian noise – AWGN). Принятый сигнал в этом случае описывается как сумма переданного сигнала  $s_i(t)$  и случайного шумового процесса  $n(t)$ :

$$r(t) = s_i(t) + n(t) \quad 0 \leq t \leq T, \quad i = 1, \dots, M \quad (1.10)$$

В приемнике используется коррелятор принятого сигнала  $r(t)$  с сигналами прототипами  $s_i(t)$ . Глагол "коррелировать" означает "совпадать", "согласовываться", следовательно, коррелятор упрощенно можно определить как устройство, которое пытается найти соответствие принятого сигнала  $r(t)$  с каждым возможным сигналом-прототипом  $s_i(t)$ , известным приемнику априори. Разумное правило принятия решения звучит так: выбирать сигнал  $s_i(t)$ , лучше всего согласующийся (или имеющий наибольшую корреляцию) с  $r(t)$ . Другими словами, правило принятия решения выглядит следующим образом рисунок 1.4

$$\text{выбрать сигнал } s_i(i), \text{ индекс которого} \quad (1.11) \\ \text{соответствует максимальной } z_i(T)$$

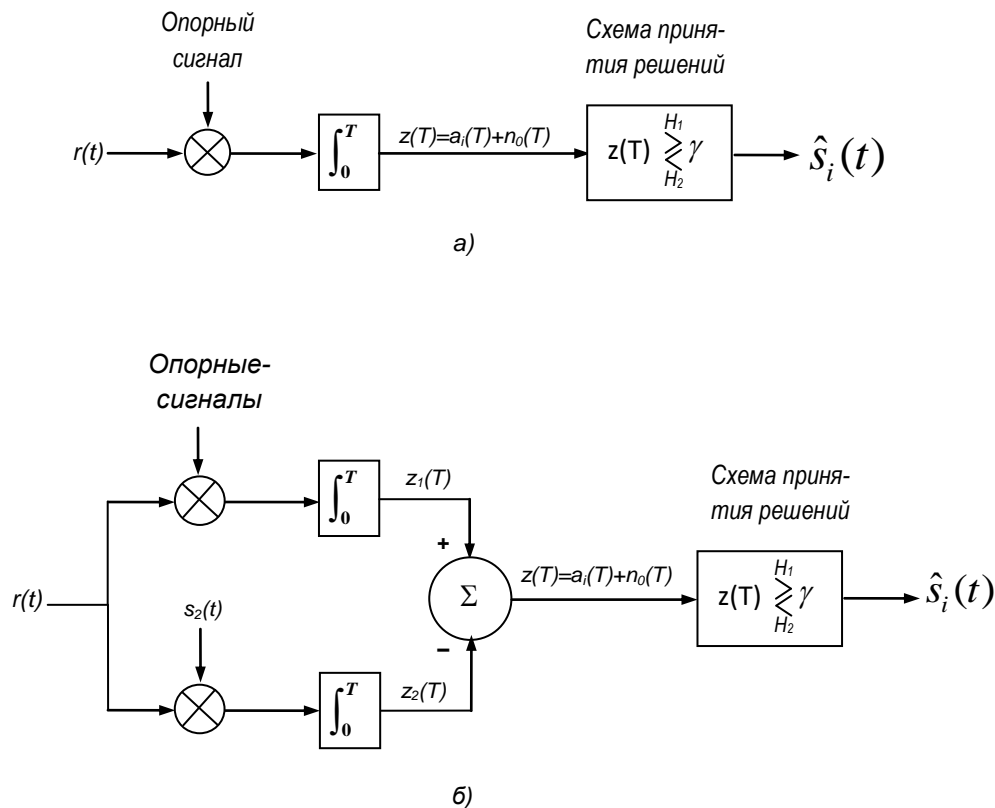


Рисунок 1.4 – Двоичный корреляционный приемник: а) использование одного коррелятора; б) применение двух корреляторов

В случае двоичного детектирования корреляционный приемник, как показано на рисунке 1.4, можно построить как согласованный фильтр или интегратор произведений с опорным сигналом, равным разности двоичных

сигналов-прототипов  $s_1(t)$ - $s_2(t)$ . Выход коррелятора  $z(T)$  используется непосредственно в процессе принятия решения.

Корреляционный приемник при двоичном детектировании также можно изобразить как два согласованных фильтра или интегратора произведений, один из которых согласовывается с  $s_1(t)$ , а второй – с  $s_2(t)$  (рисунок 1.4, б). На этапе принятия решения теперь может использоваться правило, приведенное в формуле (1.11), или же из выхода одного коррелятора можно вычесть выход другого и на этапе принятия решения использовать разность

$$z(T) = z_1(T) - z_2(T) \quad (1.12),$$

как показано на рисунке 1.4 б.

Здесь  $z(T)$ , называемое тестовой статистикой, подается в схему принятия решения, как и в случае только одного коррелятора. В отсутствие шума на выходе мы получаем  $z(T) = a_i(T)$ , где  $a_i(T)$  – сигнальный компонент. Входной шум  $n(T)$  и выходной шум  $n_0(T)$  при этом являются случайными гауссовыми процессами. Таким образом, можно записать выражение сигнала на выходе коррелятора в момент взятия выборки  $t = T$

$$z(T) = a_i(T) + n_0(T), \quad i=1,2,... \quad (1.13)$$

где  $n_0(T)$  – компонент шума, это гауссова случайная переменная с нулевым средним;

$z(T)$  – это гауссова случайная переменная со средним  $a_1$  или  $a_2$ , в зависимости от того, была передана двоичная единица или двоичный ноль.

Порог двоичного решения. Критерий минимума ошибки для равновероятных двоичных сигналов, искаженных гауссовым шумом, можно сформулировать следующим образом:

$$z(T) \underset{H_2}{\overset{H_1}{\gtrless}} \gamma_0 \quad (1.14)$$

здесь  $a_1$  — сигнальный компонент  $z(T)$  при передаче  $s_1(t)$ , а  $a_2$  — сигнальный компонент  $z(T)$  при передаче  $s_2(t)$ .

Порог  $\gamma_0$ , равный  $(a_1 + a_2)/2$ , — это оптимальный порог для минимизации вероятности принятия неверного решения при равновероятных сигналах и симметричных функциях правдоподобия. Правило принятия решения, приведенное в формуле (1.14), указывает, что гипотеза  $H_1$  (решение, что переданный сигнал – это  $s_1(t)$ ) выбирается при  $z(T) > \gamma_0$ , а гипотеза  $H_2$  (решение, что переданный сигнал – это  $s_2(t)$ ) – при  $z(T) < \gamma_0$ . Если  $z(T) = \gamma$ , решение может быть любым. При равновероятных антиподных

сигналах с равными энергиями, где  $s_1(t) = -s_2(t)$  и  $a_1 = -a_2$ , оптимальное правило принятия решения принимает следующий вид

$$z(T) \underset{H_2}{\overset{H_1}{\gtrless}} \gamma_0 = 0 \quad (1.15)$$

что можно сформулировать следующим образом

*выбрать сигнал  $s_1(t)$ , если  $z_1(T) > z_2(T)$   
выбрать сигнал  $s_2(t)$  в противном случае*

Вероятность битовой ошибки. Вероятность принятия детектором неверного решения называется вероятностью символьной ошибки,  $P_s$ . Несмотря на то, что решения принимаются на символьном уровне, производительность системы часто удобнее задавать через вероятность битовой ошибки  $P_b$ .

Формула (1.16) описывает вероятность битовой ошибки  $P_b$  для детектора, работающего по принципу минимальной вероятности ошибки

$$P_B = \int_{(a_1 - a_2)/2\sigma_0}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du = Q\left(\frac{a_1 - a_2}{2\sigma_0}\right) \quad (1.16)$$

Здесь  $\sigma_0$  – среднеквадратическое отклонение шума вне коррелятора. Функция  $Q(x)$ , называемая гауссовым интегралом ошибок, определяется следующим образом

$$Q(X) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du \quad (1.17)$$

Для передачи антиподных сигналов с равными энергиями, таких как сигналы в формате BPSK, на выход приемника поступают следующие компоненты:  $a_1 = \sqrt{E_b}$ , при переданном сигнале  $s_1(t)$ , и  $a_2 = -\sqrt{E_b}$ , при переданном сигнале  $s_2(t)$ , где  $E_b$  – энергия сигнала, приходящаяся на двоичный символ. Для процесса AWGN дисперсию шума  $\sigma_0^2$  вне коррелятора можно заменить на  $N_0/2$ , так что формулу (1.16) можно переписать следующим образом

$$P_B = \int_{\sqrt{2E_b/N_0}}^{\infty} \frac{1}{\sqrt{1\pi}} \exp\left(-\frac{u^2}{2}\right) du = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (1.18)$$

## 1.6 Дифференциальная фазовая манипуляция DPSK

Альтернативной формой двухуровневой PSK является дифференциальная PSK (differential phase-shift keying – DPSK), пример которой приведен на рисунке 1.5. В данной системе двоичный ноль представляется сигнальным пакетом, фаза которого совпадает с фазой предыдущего посланного пакета, а двоичная единица представляется сигнальным пакетом с фазой, противоположной фазе предыдущего пакета. Такая схема называется дифференциальной, поскольку сдвиг фаз выполняется относительно предыдущего переданного бита, а не относительно какого-то эталонного сигнала.

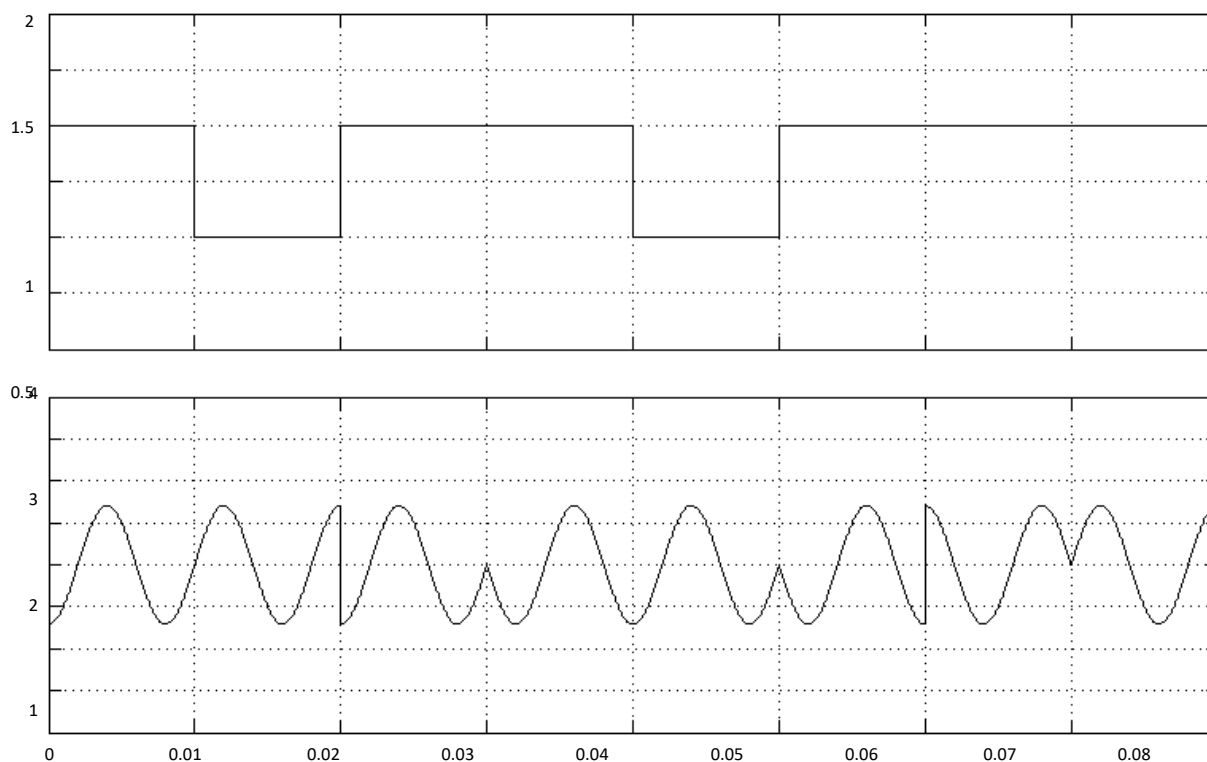


Рисунок 1.5 – Пример дифференциальная фазовой модуляции

Само название – дифференциальная фазовая манипуляция – требует некоторого пояснения, поскольку со словом «дифференциальный» связано два различных аспекта процесса модуляции/демодуляции: процедура кодирования и процедура обнаружения. Термин «дифференциальное кодирование» употребляется в том случае, когда передаваемая информация представляется не сигнальными посылками, а изменениями между последовательными сигнальными посылками. Термин «дифференциальное обнаружение» сигналов в дифференциальной модуляции PSK (именно в этом значении обычно используется название DPSK) связан со схемой

обнаружения, которая зачастую относится к некогерентным схемам, поскольку не требует согласования по фазе с принятой несущей. Стоит отметить, что дифференциально кодированные сигналы иногда обнаруживаются когерентно.

В некогерентных системах не предпринимаются попытки определить действительное значение фазы поступающего сигнала. Следовательно, если переданный сигнал имеет вид

$$s_i = \sqrt{\frac{2E}{T}} (\cos \omega_0 t + \varphi), 0 \leq t \leq T \quad (1.19)$$

то принятый сигнал можно описать следующим образом

$$r(t) = \sqrt{\frac{2E}{T}} \cos[\omega_0 t + \theta_i(t) + \alpha] + n(t), 0 \leq t \leq T, i = 1, \dots, M \quad (1.20)$$

Здесь  $\alpha$  – произвольная константа, обычно предполагаемая случайной переменной, равномерно распределенной между нулём и  $2\pi$ , а  $n(t)$  – процесс AWGN.

С точки зрения помехозащищенности, схема DPSK менее эффективна, чем PSK, поскольку в первом случае, вследствие корреляции между сигналами, ошибки имеют тенденцию к распространению (на соседние времена передачи символов). Стоит помнить, что схемы PSK и DPSK отличаются тем, что в первом случае сравнивается принятый сигнал с идеальным опорным, а во втором – два зашумленных сигнала; из-за чего при использовании DPSK следует ожидать вдвое (на 3 дБ) большей вероятности ошибки, чем в случае PSK. Ухудшение качества передачи происходит довольно быстро с уменьшением отношения сигнал/шум. Преимуществом схемы DPSK можно назвать меньшую сложность системы.

Сигналы в канале иногда инвертируются; например, при использовании когерентного опорного сигнала, генерируемого контуром ФАПЧ, фаза может быть неоднозначной. Если фаза несущей была инвертирована при использовании схемы DPSK это скажется на сообщении следующим образом: поскольку информация сообщения кодируется подобием или отличием соседних символов, единственным следствием может быть ошибка в бите, который инвертируется, или в бите, непосредственно следующим за инвертированным. Точность определения подобия или отличия символов не меняется при инвертировании несущей.

Вероятность появления ошибочного бита при некогерентном обнаружении сигналов в дифференциальной модуляции PSK (DPSK) дается выражением (1.21) [1, 12].

$$P_b = \frac{1}{2} \exp\left(-\frac{E_b}{N_0}\right) \quad (1.21)$$

## 1.7 Сигналы MPSK и их векторное представление

Когерентная М-позиционная фазовая манипуляция (M-ary phase shift keying – MPSK) – это хорошо известный метод, позволяющий сузить полосу частот, занимаемую передаваемым сигналом. Здесь используется не бинарный алфавит с передачей одного информационного бита за период передачи канального символа, а алфавит из М символов, что позволяет передавать  $I = \log_2 M$  битов за каждый символьный интервал. Поскольку использование М-позиционных символов в  $k$  раз повышает скорость передачи информации при той же полосе пропускания, то при фиксированной скорости применение М-позиционной PSK (М-PSK) сужает необходимую полосу пропускания в  $k$  раз.

На рисунке 1.6 показаны наборы сигналов MPSK для  $M = 2, 4, 8$  и 16. На рисунке 1.6, *а* видим бинарные ( $k=1, M=2$ ) антиподные векторы  $s_1$  и  $s_2$ , угол между которыми равен  $180^\circ$ . Граница областей решений разделяет сигнальное пространство на две области. На рисунке также показан вектор шума  $n$ , равный по амплитуде сигналу  $s$ . При указанных направлении и амплитуде вектора шума его амплитуда является минимальной, когда детектор может допустить символьную ошибку.

На рисунке 1.6, *б* видим четырех позиционные ( $k = 2, M = 4$ ) векторы, расположенные друг к другу под углом  $90^\circ$ . Границы областей решений (на рисунке изображена только одна) делят сигнальное пространство на четыре области.

Здесь также изображен вектор шума  $n$  (начало – в вершине вектора сигнала, направление перпендикулярно ближайшей границе областей решений), являющийся вектором минимальной энергии, достаточной, чтобы детектор допустил символьную ошибку. Отметим, что вектор шума минимальной энергии на рисунке 1.6, *б* меньше вектора шума на рисунке 1.6, *а*, что свидетельствует о большей уязвимости 4-х позиционной системы к шуму, по сравнению с бинарной (энергии сигналов в обоих случаях взяты равными).

Изучая рисунок 1.6, *в, г*, можно отметить следующую закономерность. При многофазной передаче сигналов по мере роста величины  $M$  на сигнальную плоскость помещается все больше сигнальных векторов. По мере того как векторы располагаются плотнее, для появления ошибки из-за воздействия шума требуется все меньшая его энергия. Кроме того, рисунок



позволяет взглянуть на природу компромиссов при многофазной передаче сигналов.

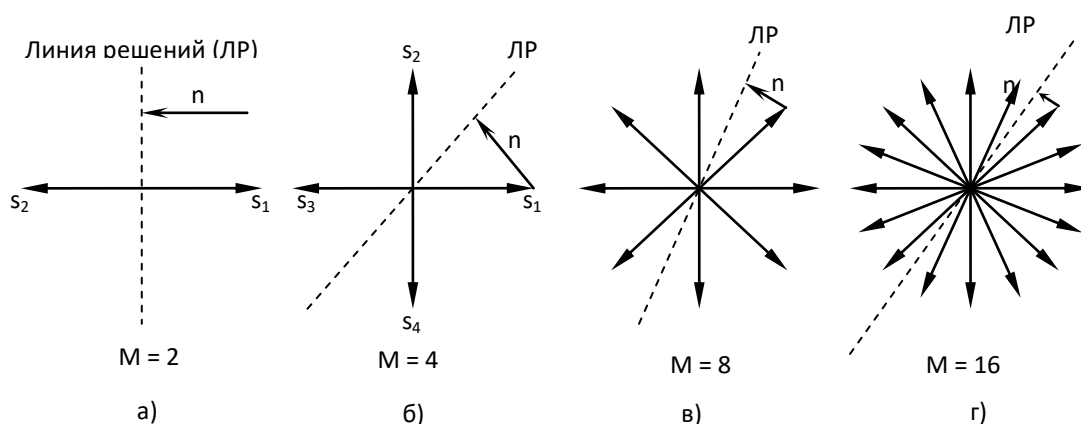


Рисунок 1.6 – Наборы сигналов MPSK для  $M = 2, 4, 8, 16$

Размещение большего числа векторов сигналов в сигнальном пространстве эквивалентно повышению скорости передачи данных без увеличения системной ширины полосы (все векторы ограничиваются одной и той же плоскостью). Другими словами, повышается эффективность использования полосы за счет увеличения вероятности ошибки. На рисунке 1.6, а – г наибольшая вероятность ошибки будет при  $M = 16$ .

Если увеличивать интенсивность сигнала, показанного на рисунке 1.6 г минимальное расстояние от вершины вектора сигнала до линии решений станет равным размеру вектора шума на рисунке 1.6, а. Таким образом, для многофазной системы по мере роста  $M$  можно увеличивать скорость передачи в фиксированной полосе частот либо за счет повышения вероятности ошибки, либо за счет увеличения отношения  $E_b/N_0$ .

На векторных диаграммах, изображенных на рисунке 1.6 для различных значений  $M$ , все векторы имеют одинаковую амплитуду. Это равносильно утверждению, что сопоставление различных схем выполняется при фиксированном отношении  $E_s/N_0$ , где  $E_s$  – энергия символа. Сравнительные схемы можно сделать и при фиксированном отношении  $E_b/N_0$ , в этом случае амплитуды векторов будут увеличиваться с ростом  $M$ . При  $M = 4, 8$  и  $16$  амплитуды векторов будут, соответственно, в  $\sqrt{2}$ ,  $\sqrt{3}$  и  $2$  раза больше векторов для случая  $M = 2$ . Как и в предыдущем случае, с ростом  $M$  будет усиливаться восприимчивость к шуму, но она не будет такой явной, как на рисунке 1.6.

Для больших отношений сигнал/шум вероятность символьной ошибки  $P_s(M)$  для равновероятных сигналов в  $M$ -позиционной модуляции PSK с когерентным обнаружением можно выразить как [4, 12]

$$P_s(M) \approx 2Q\left(\sqrt{\frac{2E_s}{N_0}} \sin \frac{\pi}{M}\right) \quad (1.22)$$

где  $P_s(M)$  — вероятность символьной ошибки,  
 $E_s = E_b(\log_2 M)$  — энергия, приходящаяся на символ,  
 $M = 2^k$  — размер множества символов.

Зависимость  $P_s(M)$  от  $E_b/N_0$  для передачи сигналов MPSK с когерентным обнаружением показана на рисунке 1.7.

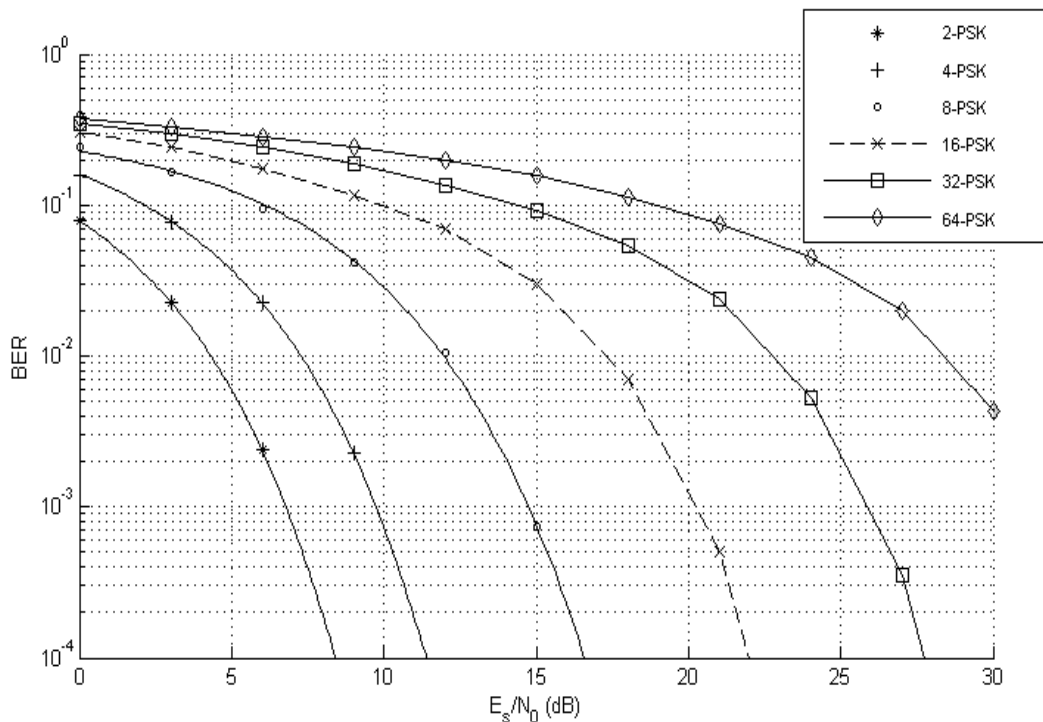


Рисунок 1.7 – Вероятность символьной ошибки для многофазной передачи сигналов с когерентным обнаружением.

Вероятность символьной ошибки для дифференциального когерентного обнаружения  $M$ -позиционной схемы DPSK (для больших значений  $E_b/N_0$ ) выражается подобно тому, как это было приведено выше

$$P_s(M) = 2Q\left(\sqrt{\frac{2E_s}{N_0}} \sin \frac{\pi}{\sqrt{2}M}\right) \quad (1.23)$$

## 1.8 Квадратурная амплитудная модуляция M-QAM

Квадратурную амплитудную модуляцию (quadrature amplitude modulation – QAM) можно считать логическим продолжением QPSK, поскольку сигнал QAM также состоит из двух независимых амплитудно-модулированных несущих (рисунок 1.8). Каждый блок из  $k$  бит можно разделить на два блока из  $k/2$  бит, подаваемых на цифро-аналоговые преобразователи (ЦАП), которые обеспечивают требуемое модулирующее напряжение для несущих. В приемнике оба сигнала детектируются независимо с помощью согласованных фильтров. Передачу сигналов, модулированных QAM, можно также рассматривать как комбинацию амплитудной (amplitude shift keying – ASK) и фазовой (phase shift keying – PSK) манипуляций, откуда альтернативное название амплитудно-фазовая манипуляция (amplitude phase keying – APK). И наконец, ее можно считать двумерной амплитудной манипуляцией, откуда еще одно название – квадратурная амплитудно-фазовая манипуляция (quadrature amplitude-phase shift keying – QAPSK).

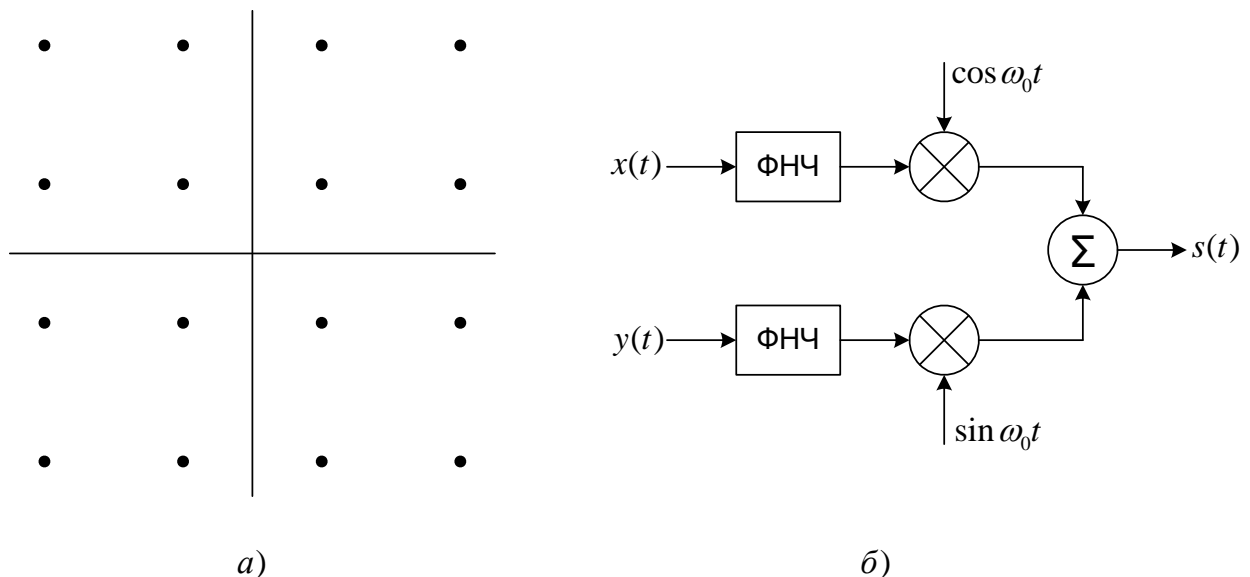


Рисунок 1.8 – Схема модуляции QAM: а) 16-ричное пространство сигналов; б) канонический модулятор QAM

На рисунке 1.8, а показано двумерное пространство сигналов и набор векторов сигналов, модулированных 16-ричной QAM и изображенных точками, которые расположены в виде прямоугольного множества. На рисунке 1.8, б показан канонический модулятор QAM [8, 12].

Вероятность появления битовой ошибки при модуляции M-QAM, где  $M = 2^k$  и  $k$  – четное, выражается следующим образом [2, 4, 12]

$$P_b = \frac{2(1-L^{-1})}{\log_2 L} Q \left[ \sqrt{\left( \frac{3 \cdot \log_2 L}{L^2 - 1} \right) \frac{2E_b}{N_0}} \right] \quad (1.24)$$

Здесь  $Q(x)$  определено в формуле (1.13), а  $L$  представляет количество уровней амплитуды в одном измерении. Предполагается, что при отображении последовательности  $\log_2 L$  бит в  $L$ -позиционный символ используется код Грея.

Для неортогональных схем, таких как MPSK, часто используется код преобразования бинарных символов в  $M$ -позиционные, такие, что двоичные последовательности, соответствующие соседним символам (сдвигам фаз), отличаются единственной битовой позицией; таким образом, при появлении ошибки в  $M$ -позиционном символе высока вероятность того, что ошибочным является только один из  $k$  прибывших битов.

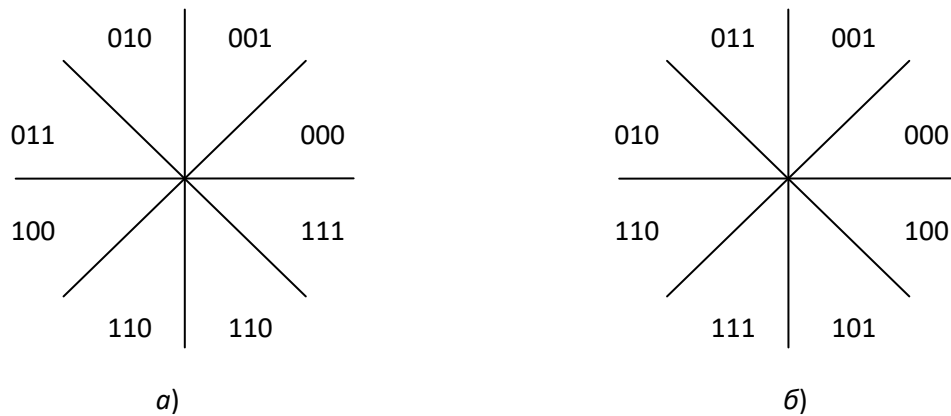


Рисунок 1.9 – Области решения в пространстве *MPSK*:  
а) в бинарной кодировке; б) в кодировке Грея

Кодом, обеспечивающим подобное свойство, является код Грея (Gray code). На рисунке 1.9 показано распределение битов по символам с использованием кода Грея для восьмеричной схемы *PSK*. Можно видеть, что соседние символы отличаются одним двоичным разрядом. Следовательно, вероятность появления многобитовой ошибки при ошибке в одном символе значительно меньше, по сравнению с некодированным распределением битов, показанным на рисунке 1.9. Реализация кода Грея представляет один из редких случаев в цифровой связи, когда определенная выгода может быть получена без сопутствующих недостатков.

## 1.9 Замирания вследствие многолучевого распространения

В системах мобильной радиосвязи сигнал может передаваться от передатчика к приемнику по множеству путей. Это явление, называемое

многолучевым распространением (multipath propagation), может вызвать флуктуации амплитуды, фазы и угла прибытия полученного сигнала, что определило название замирание вследствие многолучевого распространения. Если преобладает незамирающий компонент сигнала, распространяющийся вдоль линии прямой видимости между антеннами, огибающая замирания описывается функцией плотности вероятности Райса

$$p(r_0) = \begin{cases} \frac{r_0}{\sigma^2} \exp\left[-\frac{(r_0^2 + A^2)}{2\sigma^2}\right] I_0\left(\frac{r_0 A}{\sigma^2}\right) & \text{для } r_0 \geq 0, A \geq 0 \\ 0 & \text{для других } r_0, A \end{cases} \quad (1.25)$$

где:  $r_0$  – амплитуда огибающей релеевского замирания;

$\sigma^2$  – средняя мощность многолучевого сигнала до обнаружения;

$A$  – максимальное значение незамирающего компонента сигнала, называемого зеркальным компонентом;

$I_0(f)$  – модифицированная функции Бесселя первого рода нулевого порядка.

Распределение Райса часто записывают через параметр  $K$ , который определяется как отношение мощности зеркального компонента к мощности многолучевого сигнала. Математически это записывается как

$$K = \frac{A^2}{2\sigma^2} \quad (1.26)$$

При приближении к нулю амплитуды зеркального компонента функция плотности вероятностей Райса стремится к функции плотности вероятности Релея, имеющей следующий вид

$$p(r_0) = \begin{cases} \frac{r_0}{\sigma^2} \exp\left[-\frac{r_0^2}{2\sigma^2}\right] & \text{для } r_0 \geq 0 \\ 0 & \text{для других } r_0 \end{cases} \quad (1.27)$$

Замирание называется Релеевским, если имеется большое число многократно отражающихся лучей и нет компонента сигнала вдоль луча обзора; огибающая такого сигнала статистически описывается с помощью релеевской функции плотности вероятности. Иными словами, статистики замирания распределены по Релею, если путь распространения вдоль луча обзора блокирован, в противном случае имеем распределение Райса.

Помимо параметра  $K$  замирания характеризуются задержкой распространения и доплеровским смещением частоты. Эффект Доплера –

отклонение частоты, воспринимаемой приемником, от частоты, переданной передатчиком, которое возникает вследствие относительного движения передатчика и приемника. Если пренебречь эффектами второго и более высоких порядков, смещение частоты равно

$$\Delta f = \frac{Vf_0}{c} \quad (1.28)$$

где:  $V$  – относительная скорость (положительная, если расстояние между приемником и передатчиком сокращается);

$f_0$  – номинальная частота передачи;

$c$  – скорость света.

## Контрольные вопросы

1. Поясните, для чего при цифровой передаче сигналов используют модуляцию.
2. Какие три основных фактора влияют на помехоустойчивость приёма, и как они взаимосвязаны между собой?
3. В чем отличие когерентного и некогерентного детектирования?
4. По каким критериям оценивается качество в цифровых системах связи?
5. В чем различие двоичной фазовой модуляции и дифференциальной двоичной фазовой модуляции? Каковы положительные и отрицательные стороны этих двух схем?
6. В какой зависимости между собой находятся скорость передачи, ширина спектральной полосы и помехоустойчивость при многофазной передаче сигналов?
7. Дайте понятие вероятности битовой и символьной ошибки.
8. Поясните суть многопозиционной квадратурной амплитудной модуляции. Какие параметры сигнала используются для передачи информации при QAM?
9. Какие принципы используются при формировании сигнала в кодировке Грея?
10. Какие параметры характеризуют замирания вследствие многолучевого распространения? В чем отличие замираний Релея и Райса?

## 2 Помехоустойчивое кодирование в системах телекоммуникаций

### 2.1 Роль и место помехоустойчивого кодирования в системах телекоммуникаций

Система связи соединяет источник данных с получателем посредством канала связи. Структурная схема системы связи изображена на рисунке 2.1.

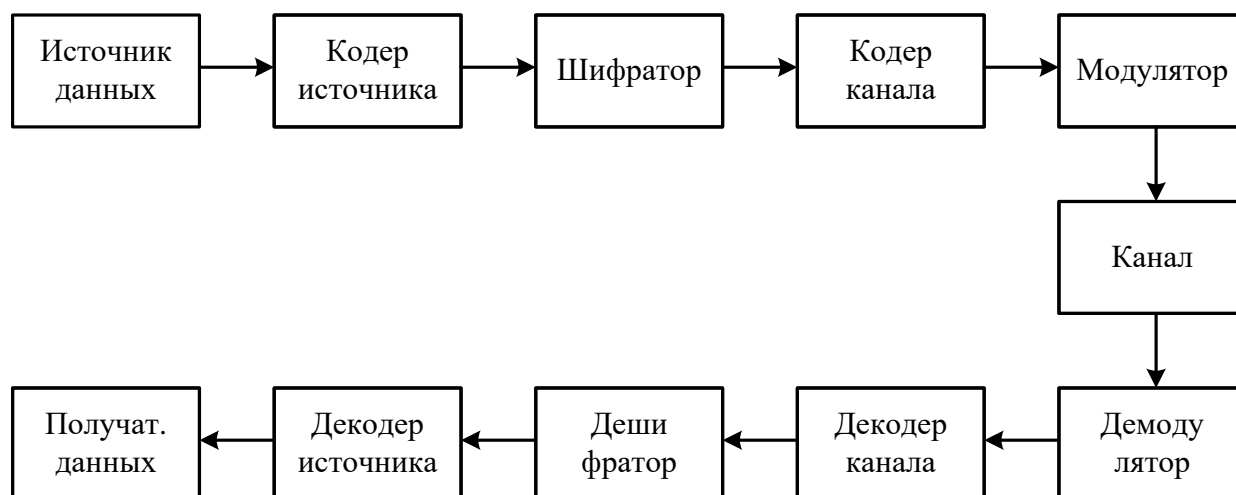


Рисунок 2.1 - Обобщенная структурная схема системы связи

Статистическое кодирование используется для уменьшения первичной избыточности передаваемой информации. Для описания этого процесса широко используется понятие "сжатие данных". В зависимости от области применения различают два основных типа схем сжатия:

1) «Без потерь информации» – коды Шеннона-Фано, Хаффмена, арифметическое кодирование, коды Лемпела-Зива (используемые, например, при архивировании данных в ЭВМ - архиваторы Zip, Rar и др.);

2) «С потерями информации» – кодирование с преобразованием и оценкой движения (например, сжатие статических изображений и видео).

Криптографическая защита используется для предотвращения несанкционированного доступа к информации. Широкое применение нашли системы криптографической защиты в среде Интернет (финансовые операции и операции аутентификации пользователей), в различных программных продуктах (операционные системы и программные приложения) и др. Архиватор Rar использует шифрование, Zip - нет. Популярный формат электронных документов pdf (portable document format), предназначенный для межмашинного обмена готовыми документами и используемый в издательствах, также использует шифрование (в отличие от аналогичного формата ps – post script). Следует отметить, что шифрование



наиболее эффективно при работе с данными, имеющими малую избыточность, поэтому используется предварительное сжатие данных.

Помехоустойчивое кодирование предназначено для защиты данных от ошибок и применяется в системах передачи и хранения данных либо только для обнаружения ошибок, либо для обнаружения и исправления ошибок.

Последовательность действий при передаче дискретных сообщений может быть пояснена следующим образом. Поток данных, поступающих на вход кодера помехоустойчивого кода (кодер ПК), преобразуется в новый поток данных путем разбиения на кодируемые группы и добавления избыточности, необходимой для борьбы с помехами, имеющими место в канале связи [1. 2. 4].

Основная задача перемежителя состоит в перестановке элементов потока данных с выхода кодера помехоустойчивого кода таким образом, чтобы деперемежитель на приемной стороне декоррелировал помехи, т.е. преобразовал пакеты ошибок, проявляющихся в реальных дискретных каналах связи (ДКС) в поток независимых ошибок.

Модулятор преобразует кодовые символы с выхода перемежителя в соответствующие аналоговые символы. Так как в канале связи возникают различного типа шумы, искажения и интерференция, то сигнал на входе демодулятора (первое решающее устройство) отличается от сигнала на выходе модулятора. При этом сигнал на выходе демодулятора является оценкой соответствующего переданного символа, а из-за шумов в канале возникают ошибки. В зависимости от характера оценки доверия к сигналу на выходе демодулятора различают мягкое и жесткое решения. В соответствии с правилом жесткого решения сигнал на выходе демодулятора определен однозначно для каждого тактового интервала ("0" или "1" для двоичного канала). Демодулятор с мягким решением обычно имеет два выхода: один из них представляет собой жесткое решение, на втором выходе формируется оценка качества этого решения в виде веса  $w_i$ , пропорционального отношению правдоподобия на выходе демодулятора. Мягкое решение демодулятора может быть использовано для дальнейшего повышения помехоустойчивости приема (например, мягкого декодирования кодовых слов помехоустойчивого кода). Наиболее часто используется восьмиуровневая оценка качества (три двоичных символа).

Декодер помехоустойчивого кода (второе решающее устройство) использует избыточность кодового слова для того, чтобы обнаружить или обнаружить и исправить ошибки в принятом слове, и затем выдает оценку кодового слова источника сигнала потребителю. Если все ошибки исправлены, то оценка кодового слова совпадает с исходным кодовым словом источника, следовательно, информация достигла получателя без искажений.

## 2.2 Дискретный канал связи

Если входные и выходные сигналы канала являются дискретными, то и канал называется дискретным (ДКС). На рисунке 2.2 приведена обобщенная структурная схема системы передачи дискретных сообщений (СПДС).

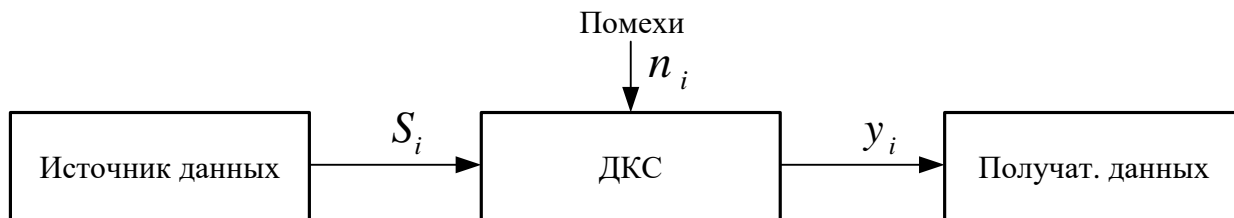


Рисунок 2.2 – Обобщенная структурная схема СПДС

Математическая модель ДКС требует описания следующих параметров:

- 1) алфавитов входных и выходных сообщений;
- 2) скорости передачи элементов алфавита;
- 3) переходных вероятностей.

Диаграмма состояний и переходов для двоичного дискретного канала связи со стираниями показана на рисунке 2.3, где  $S_0, S_1$  - элементы алфавита источника;  $y_0, y_1$  - элементы алфавита на выходе канала;  $q$  - символ стирания;  $p(y_i/S_j)$  и  $p(q/S_j)$  - переходные вероятности, где  $i, j \in \{0,1\}$ .

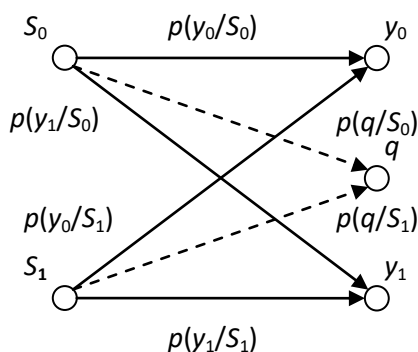


Рисунок 2.3 – Диаграмма состояний и переходов для двоичного ДКС

Характеристики непрерывного канала (в том числе характер действия помех в линии связи) проявляются в свойствах переходных вероятностей ДКС. В результате этого ДКС могут быть:

- 1) симметричными, когда переходные вероятности  $p(y_i/S_j)$  одинаковы для всех  $i \neq j$  и, соответственно, несимметричными в противном случае;

2) без памяти, когда переходные вероятности  $p(y_i/S_i)$  не зависят от того, какие символы и с каким качеством передавались до данного символа  $S_i$  и с памятью в противном случае;

3) без стирания, когда алфавиты на входе канала и выходе демодулятора совпадают, в канале со стиранием алфавит на выходе демодулятора имеет дополнительный символ стирания  $q$ , формируемый тогда, когда демодулятор не может с заданной надежностью опознать переданный символ.

Ошибки в симметричных каналах без памяти независимы и характеризуются вероятностью ошибки  $P$  - примером является биномиальный канал.

В каналах с памятью ошибки пакетируются (группируются) – примером является Марковский канал.

## 2.3 Основные понятия и определения теории кодирования

Избыточность кода

$$g = \frac{n - k}{m} = \frac{r}{n}, \quad (2.1)$$

где  $n$  – количество элементов (символов) в кодовом слове;  $k$  и  $r$  – количество информационных и проверочных символов, соответственно.

Скорость кода

$$R = \frac{k}{n} \quad (2.2)$$

Чем больше избыточность кода, тем меньше скорость кода, и наоборот. Расстояние Хэмминга между двумя кодовыми словами  $d_{ij}$

$$d_{ij} = \sum_{k=1}^n |x_{ik} - x_{jk}|, \quad (2.3)$$

где  $x_{ik}$ ,  $x_{jk}$  - координаты слов  $A_i$ ,  $A_j$  в  $n$ -мерном неевклидовом пространстве.

Если код является двоичным, под расстоянием между парой кодовых слов понимается количество символов, в которых они отличаются между собой. Оно определяется сложением этих двух слов по  $\text{mod } 2$  и равно числу единиц в этой сумме

$$d_{ij} = [(A_i + A_j) \bmod 2]_{\sum "1"} \quad (2.4)$$

где  $(..+..) \bmod 2$  – сложение по mod2;  $\sum "1"$  – означает, что после операции сложения по модулю два необходимо подсчитать количество единиц в полученном результате.

Таким образом, расстоянием Хэмминга между двумя двоичными последовательностями, называется число позиций, в которых они различны. Минимальное расстояние Хэмминга называется кодовым расстоянием

$$d = \min d_{ij} \quad (2.5)$$

Обнаруживающая и исправляющая способности корректирующих кодов тесно связаны с расстояниями между разрешенными кодовыми словами. Число ошибочных символов в принятом кодовом слове называется кратностью ошибки  $t$ , при длине кодового слова  $n$  символов она изменяется в пределах от 0 до  $n$ . Так как кратность ошибки  $t$  в геометрическом представлении является расстоянием между переданным словом и принятым, то для обнаружения ошибок кратности  $t_0$  требуется кодовое расстояние

$$d \geq t_0 + 1, \quad (2.6)$$

Для исправления ошибок кратности  $t_{\text{и}}$ , требуется кодовое расстояние

$$d \geq t_{\text{и}} + 1, \quad (2.7)$$

Это означает, что для исправления ошибок искаженное кодовое слово должно располагаться ближе всего к соответствующему правильному слову. Кратность исправления  $t_{\text{и}}$  определяет границу гарантированного исправления ошибок. В случае исправления  $t_{\text{и}}$  ошибок и  $t_q$  стираний кодовое расстояние  $d$ , [1]

$$d \geq t_{\text{и}} + 1 + t_q, \quad (2.8)$$

## 2.4 Кодирование и защита от ошибок

Существуют три наиболее распространенных способа борьбы с ошибками в процессе передачи данных:

- коды с обнаружением ошибок;

- коды с коррекцией ошибок, называемые также схемами прямого исправления ошибок (forward error correction – FEC);
- протоколы с автоматическим запросом повторной передачи (automatic repeat request – ARQ).

Код с обнаружением ошибок позволяет довольно легко установить наличие ошибки. Как правило, подобные коды используются совместно с определенными протоколами канального или транспортного уровня, имеющими схему ARQ. В схеме ARQ приемник попросту отклоняет блок данных, в котором была обнаружена ошибка, после чего передатчик передает этот блок повторно. Коды прямого исправления ошибок позволяют не только обнаружить ошибки, но и исправить их, не прибегая к повторной передаче. Схемы FEC часто используются в беспроводной передаче, где повторная передача крайне неэффективна, а уровень ошибок довольно высок.

Выявление ошибок. Далее будем считать, что данные передаются как одна или несколько непрерывных последовательностей битов, которые называют кадрами. Определим вероятности, связанные с возникновением ошибок в переданных кадрах:

$P_B$  – вероятность появления единичного ошибочного бита; именуется также частотой появления ошибочных битов (bit error rate – BER);

$P_1$  – вероятность безошибочного приема кадра;

$P_2$  – вероятность того, что используемый алгоритм выявления ошибок не позволяет обнаружить ошибку в кадре;

$P_3$  – вероятность того, что используемый алгоритм выявления ошибок позволяет обнаружить все ошибки в кадре.

Рассмотрим для начала пример, когда при передаче данных схемы выявления ошибок не используются. В этом случае вероятность обнаружения всех ошибок ( $P_3$ ) равна нулю. Чтобы найти значения остальных вероятностей, предположим, что каждый бит может быть ошибочным с равной вероятностью, т.е.  $P_B$  – постоянная независимая величина для каждого бита. Тогда можно записать:

$$\begin{aligned} P_1 &= (1 - P_B)^F, \\ P_2 &= 1 - P_1, \end{aligned} \tag{2.9}$$

где  $F$  – число битов в кадре.

Иными словами, вероятность получения кадра без ошибок уменьшается с ростом вероятности битовой ошибки. Кроме того, вероятность отсутствия ошибок в полученном кадре уменьшается с увеличением длины кадра; чем длиннее кадр, тем больше в нем битов и тем больше вероятность ошибочности одного из них.

Чтобы проиллюстрировать приведенные соотношения, рассмотрим простой пример. Для цифровых сетей с интеграцией услуг (Integrated Services Digital Network – ISDN) стандартной задачей является поддержание частоты появления ошибочных битов в канале 64 Кбит/с ниже  $10^{-6}$ , по крайней мере для 90% интервалов длительностью 1 мин. Теперь предположим, что требования пользователя являются сравнительно скромными – в течение суток в среднем должно появляться не более одного кадра с необнаруженной битовой ошибкой. Длину кадров примем равной 1000 бит. В течение суток может быть передано  $5,529 \times 10^6$  кадров. Вычислим максимально допустимую частоту появления ошибочных кадров.  $P_2 = 1/(5,529 \times 10^6) = 0,18 \times 10^{-6}$ . Для принятого нами значения  $P_B = 10^{-6}$   $P_1 = (0,999999)^{1000} = 0,999$ . Следовательно  $P_2 = 10^{-3}$ , что на три порядка превышает требования.

Полученный результат свидетельствует о необходимости применения схем обнаружения ошибок. Работа всех методов обнаружения ошибок основывается на следующем принципе: к информационному кадру передатчиком добавляется последовательность битов, которые составляют код обнаружения ошибок (рисунок 2.4). Этот код вычисляется как функция переданных битов. Обычно для информационного блока из  $k$  бит алгоритм обнаружения ошибок дает код, имеющий  $n-k$  бит, причем  $(n-k) < k$ . Код обнаружения ошибок (иногда называемый контрольными битами) присоединяется к блоку данных, в результате чего получается последовательность из  $n$  бит, которая и передается. Приемник разделяет полученную последовательность на  $k$  бит данных и  $(n-k)$  бит кода обнаружения ошибок. Основываясь на битах данных, приемник вычисляет код, после чего сверяет результат с принятым кодом обнаружения ошибок. Если два кода не совпадают, имеется ошибка. Следовательно, параметр  $P_3$  – это вероятность того, что в кадре присутствует ошибка и она обнаружена с помощью используемой схемы. Параметр  $P_2$  называют остаточным уровнем ошибок.  $P_2$  – это вероятность того, что ошибка не будет обнаружена, несмотря на использование схемы выявления ошибок.

Проверка четности. Наиболее простой метод обнаружения ошибок – добавление бита четности в конец каждого блока данных. Типичный пример: передача знаков, во время которой бит четности добавляется к каждому 7-битовому знаку. Значение этого контрольного бита выбирается так, чтобы общее число единиц в знаке было или нечетным (отрицательная четность), или четным (положительная четность).

Например, если передатчик передает 1110001, используя схему с отрицательной четностью, в конец последовательности будет добавлена двоичная единица. В итоге передана будет последовательность 11100011. После получения знака приемник проверяет количество двоичных единиц, и если оно нечетное, считается, что ошибок нет. Если же в процессе передачи один бит (или любое нечетное их количество) изменил значение – будет обнаружено наличие ошибки. Необходимо отметить, что если свое значение

изменяют два бита (или любое четное их количество), ошибка обнаружена не будет. Как правило, положительная четность используется при синхронной передаче данных, а отрицательная – при асинхронной.

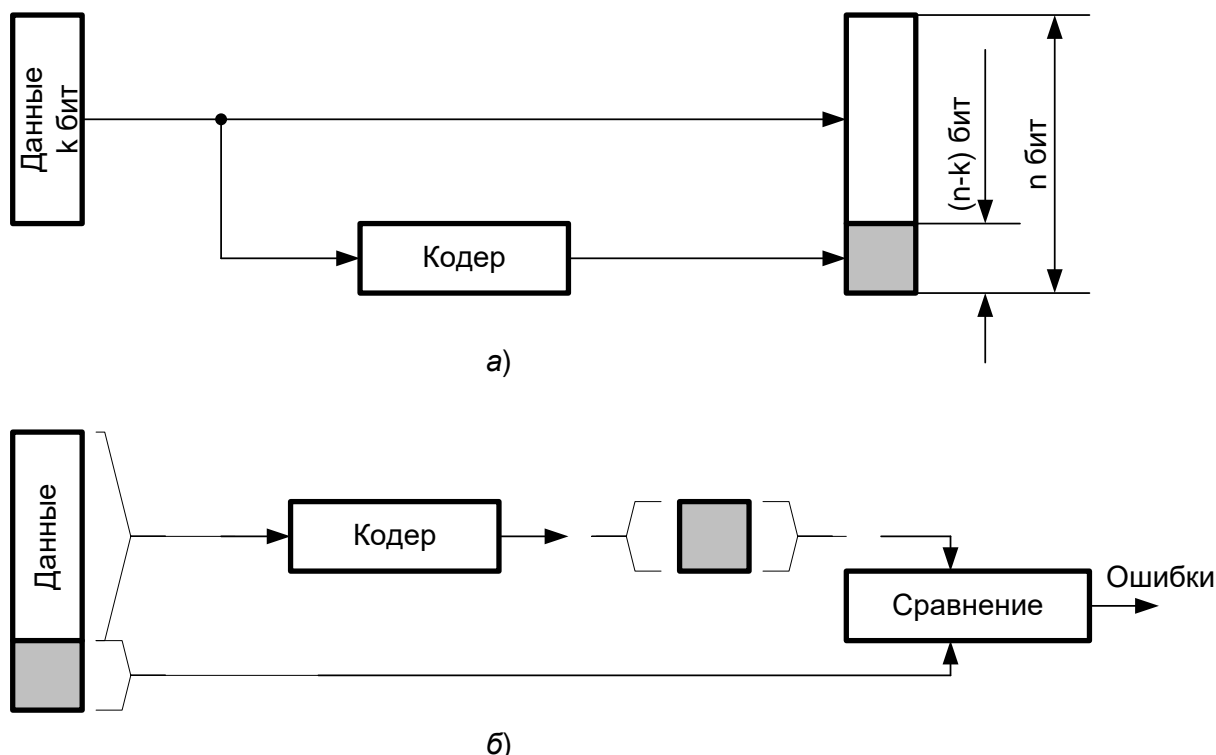


Рисунок 2.4 – Процесс обнаружения ошибок: а) формирование кода обнаружения ошибок; б) использование кода на приемной стороне

Проверка четности не является надежным средством обнаружения ошибок, поскольку шумовые импульсы обычно довольно продолжительны (в особенности при высоких скоростях передачи данных), чтобы изменить более одного бита.

#### 2.4.1 Циклическая проверка четности с избыточностью

Циклическая проверка четности с избыточностью (cyclic redundancy check – CRC) – это один из наиболее широко используемых и надежных методов обнаружения ошибок. Принцип работы данного метода сводится к следующему: для блока из  $k$  бит (сообщения) передатчик генерирует так называемую контрольную последовательность кадра (frame check sequence – FCS) из  $(n - k)$  бит. При этом результирующая последовательность (состоящая из данных и FCS) должна делиться без остатка на заданную константу. После передачи приемник делит полученную последовательность

на эту константу и, если деление не дало остаток считает, что ошибки в процессе передачи отсутствовали.

Для лучшего понимания этой процедуры мы опишем ее тремя способами: через арифметические действия по модулю 2, действия с полиномами и цифровую логику.

*Полиномы.* Процесс циклической проверки четности с избыточностью можно также рассмотреть, представив все значения в форме полиномов фиктивной переменной  $X$  с двоичными коэффициентами, которые соответствуют битам двоичного числа. Как и выше, арифметические операции выполняются по модулю 2. Процесс проверки можно описать следующим образом:

$$\frac{X^{n-k}D(X)}{P(X)} = Q(X) + \frac{R(x)}{P(X)} \quad (2.10)$$

$$T(X) = X^{n-k}D(X) + R(X),$$

Ошибка не обнаружится только в том случае, если соответствующий полином  $E(X)$  делится на  $P(X)$ . Можно показать, что при надлежащем выборе полинома  $P(X)$  выявляются такие ошибки:

- все 1-битовые ошибки, если  $P(X)$  имеет более одного ненулевого члена;
- все 2-битовые ошибки, если  $P(X)$  имеет делитель из трех членов;
- любое нечетное количество ошибок, если в разложении  $P(X)$  во множителе присутствует  $(X + 1)$ ;
- любой пакет ошибок, длина которого не превышает  $n - k$ , или, эквивалентно, не превышает длину контрольной последовательности кадра;
- часть пакета ошибок длиной  $n - k + 1$ ; часть равна  $1 - 2^{-(n-k-1)}$ ;
- часть пакета ошибок длиной более  $n-k+1$ ; часть равна  $1 - 2^{-(n-k)}$ .

Кроме того, можно показать, что если все последовательности ошибок считать равновероятными, то для пакета ошибок длиной  $r + 1$  вероятность появления необнаруженной ошибки (т.е. вероятность того, что  $E(X)$  делится на  $P(X)$ ) равна  $1/2^{r-1}$ . Для пакета ошибок большей длины вероятность появления необнаруженной ошибки равна  $1/2^r$ , где  $r$  – длина контрольной последовательности кадра.

Наиболее широко используются четыре полинома  $P(X)$ :

$$\text{CRC-12} \quad X^{12} + X^{11} + X^3 + X^2 + X + 1$$

$$\text{CRC-16} \quad X^{16} + X^{15} + X^2 + 1$$



$$\text{CRC-CCITT} \quad X^{16} + X^{12} + X^5 + 1$$

$$\text{CRC-32} \quad X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^3 + X^7 + X^5 + X^4 + X^2 + X + 1.$$

Система CRC-12 используется при передаче потоков 6-битовых знаков; здесь генерируются 12-битовые FCS. Системы CRC-16 и CRC-CCITT применяются при передаче 8-битовых знаков в США и Европе, соответственно, здесь создаются контрольные последовательности длиной 16 бит. В большинстве случаев этих полиномов достаточно, хотя иногда (например, в стандартах двухточечной синхронной передачи) как вариант указан полином CRC-32.

*Цифровая логика.* Процесс циклической проверки четности с избыточностью можно представить (и, как правило, он так и реализуется) как схему деления, состоящую из элемента исключающего ИЛИ и регистра сдвига. Регистр сдвига представляет собой строку 1-битовых ячеек памяти. Каждая ячейка имеет выходную шину, показывающую текущее хранимое значение, и входную шину. В дискретные моменты времени, которые называют тактами, значения ячеек памяти замещаются значениями, указанным во входной шине. Замена происходит синхронно во всем регистре, так что в результате значения ячеек регистра сдвигаются на один бит.

Реализация схемы выглядит следующим образом:

1. Регистр, содержащий  $(n - k)$  бит (по размеру контрольной последовательности кадра);
2. До  $(n - k)$  элементов исключающего ИЛИ;
3. Наличие или отсутствие логического элемента соответствует наличию или отсутствию члена в полиноме-делителе  $P(X)$ , исключая члены 1 и  $X^{n-k}$ .

Пример. Понять архитектуру схемы деления можно с помощью примера, приведенного на рисунке 2.5. В примере, как и ранее, используются такие величины:

$$\begin{array}{ll} \text{сообщение} & D = 1010001101; D(X) = X^9 + X^1 + X^3 + X^2 + 1; \\ \text{делитель} & P = 110101; P(X) = X^5 + X^4 + X^2 + 1. \end{array}$$

На рисунке 2.5, а представлена реализация регистра сдвига. Процесс начинается с очистки регистра (все ячейки обнуляются). После этого передаваемое сообщение (делимое) побитово вводится в регистр, начиная со старшего бита. На рисунке 2.5, б приведена таблица, которая иллюстрирует пошаговую работу схемы по мере введения отдельных битов. Строки таблицы содержат значения пяти ячеек регистра сдвига в соответствующие

моменты времени. Кроме того, в строках таблицы приводятся значения на выходе трех схем исключающего ИЛИ. Последнее число в каждой строке – значение следующего входного бита, который станет доступен для работы на следующем этапе.

Отметим, что операция исключающего ИЛИ влияет на значения ячеек  $C_4$ ,  $C_2$  и  $C_0$  при следующем сдвиге, что идентично рассмотренному ранее процессу двоичного деления. Процесс выполняется для всех битов передаваемого сообщения. Для обеспечения корректности выходного сигнала используются два ключа. При вводе битов данных оба ключа находятся в положении А. В результате за первых 10 шагов входные биты подаются в регистр сдвига и также используются в качестве выходных битов. По окончании обработки последнего бита данных регистр сдвига содержит остаток деления (FCS), выделенный серым цветом. При вводе последнего бита данных в регистр оба ключа устанавливаются в положение В. В этом случае: 1) все логические элементы больше не изменяют значения битов; 2) за следующие 5 шагов на выход подаются 5 бит CRC.

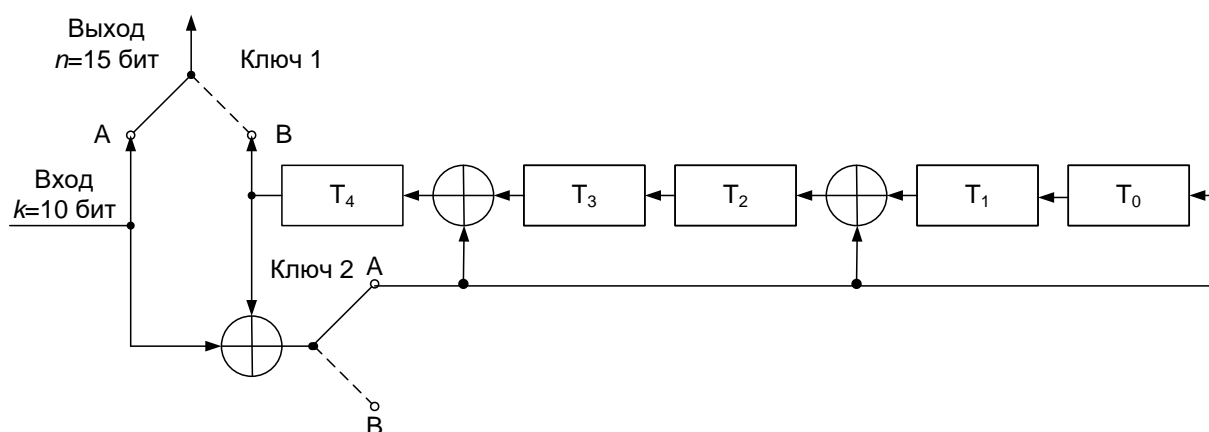


Рисунок 2.5 – Реализация устройства формирования кода CRC-5

Таблица 2.1 – Пошаговая работа устройства формирования кода CRC-5

Такты	$T_4$	$T_3$	$T_2$	$T_1$	$T_0$	$T_4 \oplus T_3 \oplus I$	$T_4 \oplus T_1 \oplus I$	$T_4 \oplus I$	$I$ (ввод данных )
0	0	0	0	0	0	1	1	1	1
1	1	0	1	0	1	1	1	1	0
2	1	1	1	1	1	1	1	0	1
3	1	1	1	1	0	0	0	1	0
4	0	1	0	0	1	1	0	0	0
5	1	0	0	1	0	1	0	1	0
6	1	0	0	0	1	0	0	0	1
7	0	0	0	1	0	1	0	1	1
8	1	0	0	0	1	1	1	1	0
9	1	0	1	1	1	0	1	0	1
10	0	1	1	1	0				

В приемнике используется аналогичная логика. Принятые биты последовательности  $K$  вводятся в регистр сдвига по мере поступления. Если ошибки отсутствуют, то после обработки  $K$  бит регистр сдвига будет содержать последовательность  $R$ . После этого начинает поступать переданная последовательность  $R$ . В результате на выход регистра будут подаваться двоичные нули и по завершении приема все ячейки будут иметь значение 0.

На рисунке 2.6 представлена общая архитектура реализации регистра сдвига CRC для полинома  $P(X) = \sum A_i X^i$ , где  $A_0 = A_{n-k} = 1$ , все остальные  $A_i$  равны 0 или 1.

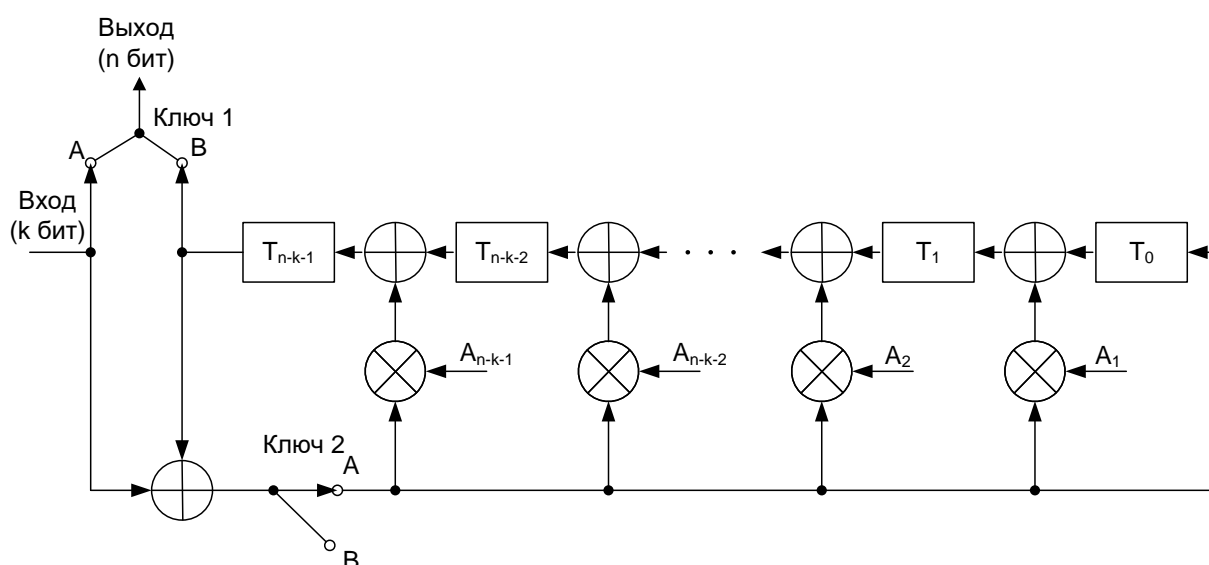


Рисунок 2.6 – Общая архитектура формирования кода CRC для реализации полинома-делителя  $P(X) = X^{n-k} + A_{n-k-1}X^{n-k-1} + \dots + A_2X^2 + A_1X + 1$

## 2.4.2 Блочные коды с коррекцией ошибок

Методы обнаружения ошибок широко применяются на практике в протоколах управления каналами передачи данных, таких, как HDLC, а также в транспортных протоколах, таких, как TCP. В то же время под использованием кодов обнаружения ошибок подразумевается повторная передача блоков данных согласно процедуре ARQ. Для беспроводных приложений такой подход неприемлем по двум причинам.

1. Уровень ошибок в беспроводном канале может быть довольно высок; в результате потребуется значительное число повторных передач.

2. В некоторых случаях, особенно в спутниковой связи, задержка распространения сигнала довольно велика по сравнению со временем передачи одного кадра. При повторной передаче, как правило, передается

кадр, содержащий ошибку, а также все последующие кадры. При большом расстоянии между приемником и передатчиком ошибка в одном кадре приводит к необходимости повторной передачи множества кадров.

Вместо повторной передачи было бы лучше, если бы приемник мог исправлять ошибки в полученном сигнале, используя информацию, содержащуюся в самом сигнале. На рисунке 2.7 представлена схема реализации этой идеи. С помощью кодера FEC (forward error correction – прямое исправление ошибок) передатчик преобразует каждый  $k$ -битовый блок данных в  $n$ -битовый блок ( $n > k$ ), именуемый кодовым словом, который затем передается (в беспроводной связи для передачи используется созданный модулятором аналоговый сигнал). При распространении сигнал может подвергаться воздействию шума, что может привести к появлению ошибочных битов. Приемник демодулирует полученный сигнал, преобразовывая его в строку битов, подобную переданной, но, возможно, с ошибками. Полученный блок данных обрабатывается декодером FEC, в результате возможны такие ситуации:

1. При отсутствии ошибочных битов вход декодера FEC идентичен исходному кодовому слову, так что на выход декодера поступает исходный блок данных;

2. Декодер может обнаружить и исправить определенные последовательности ошибок. Поэтому даже если вход декодера отличается от исходного кодового слова, из него можно получить исходные данные;

3. Некоторые последовательности ошибок могут быть обнаружены декодером, но не могут быть исправлены. В этом случае декодер сообщает о наличии неисправимой ошибки;

4. Наличие некоторых (обычно довольно редких) последовательностей ошибок не может быть обнаружено декодером. В результате декодер преобразовывает входной  $n$ -битовый блок в  $k$ -битовую последовательность, которая отличается от переданной, но которую кодер считает правильной.

Каким образом декодер исправляет ошибки? По сути, исправление ошибок производится с помощью добавления избыточных данных к передаваемому сообщению. Избыточность позволяет приемнику восстановить исходное сообщение даже при наличии определенного уровня ошибок. В данном разделе будет рассмотрен широко используемый класс кодов с коррекцией ошибок, известный как блочные коды. В начале будут рассмотрены общие принципы, после чего мы перейдем к анализу конкретных кодов.

Отметим, что довольно часто коды с коррекцией ошибок используются по схеме, изображенной на рисунке 2.4 для кодов обнаружения ошибок. Т.е. в алгоритме FEC к входному  $k$ -битовому блоку данных добавляется  $(n - k)$  контрольных битов, в результате размер передаваемого блока составляет  $n$  бит, все биты исходного  $k$ -битового блока содержатся в полученном  $n$ -

битовом блоке. Для некоторых схем прямого исправления ошибок (например, сверточных кодов) входная  $k$ -битовая последовательность так преобразовывается в  $n$ -битовое кодовое слово, что исходные  $k$  бит не фигурируют явно в кодовом слове.

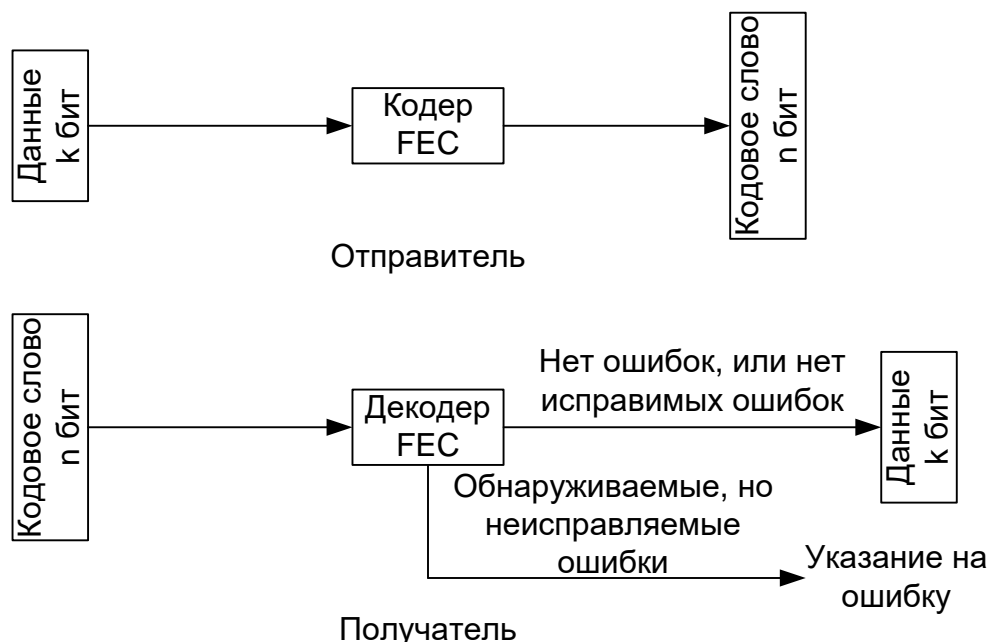


Рисунок 2.7 – Принцип прямого исправления ошибок

Принципы блочных кодов. Для начала определим термины, которыми мы будем оперировать. Расстоянием Хэмминга (Hamming distance)  $d(v_1, v_2)$  между двумя  $n$ -битовыми двоичными последовательностями  $v_1$ , и  $v_2$  называют число несовпадающих разрядов  $v_1$  и  $v_2$ . Например,

$$v_1 = 011011, v_2 = 110001, \\ d(v_1, v_2) = 3$$

Рассмотрим теперь метод блочного кодирования с целью коррекции ошибок. Пусть требуется передать определенное количество  $k$ -битовых блоков данных. Вместо передачи каждого блока как последовательности  $k$  бит, преобразуем каждую  $k$ -битовую последовательность в уникальное  $n$ -битовое кодовое слово.

Важные свойства блочных кодов с коррекцией ошибок. Блочный код  $(n, k)$  преобразует  $k$  бит данных в  $n$ -битовые кодовые слова. В большинстве случаев каждое корректное кодовое слово – это исходные  $k$  бит данных плюс  $(n-k)$  контрольных битов. Таким образом, структура блочного кода эквивалентна структуре функции вида  $v_c = f(v_d)$ , где  $v_d$  – вектор, состоящий из  $k$  бит данных;  $v_c$  – вектор, состоящий из  $n$  бит кодового слова.

В блочном коде  $(n, k)$  имеется  $2^k$  приемлемых кодовых слов из  $2^n$  возможных. Отношение числа избыточных битов к числу битов данных  $(n - k)/k$  принято называть избыточностью кода (code redundancy); отношение количества битов данных к полному числу битов  $(k/n)$  называют степенью кодирования (code rate). Степень кодирования – это мера того, какая дополнительная полоса потребуется после кодирования, если скорость передачи данных мы хотим сохранить такой же, какая была до кодирования. Например, при степени кодирования  $1/2$  для сохранения скорости передачи системе потребуется полоса, в два раза большая, чем та, которая необходима для передачи некодированного сигнала. Если степень кодирования равна  $2/5$ , следовательно, для сохранения скорости передачи потребуется увеличить ширину полосы в 2,5 раза. Т.е. если скорость передачи сигнала на входе кодера равна 1 Мбит/с, то для сохранения прежних параметров скорость выходного сигнала должна быть равна 2,5 Мбит/с.

Для кода, состоящего из кодовых слов  $w_1, w_2, \dots, w_s$ , где  $s = 2^n$ , минимальное расстояние кода  $d_{\min}$  определяется следующим образом

$$d_{\min} = \min[d(w_i, w_j)] \quad (2.11)$$

Можно показать, что если для кода выполняется неравенство  $d_{\min} > 2t + 1$  ( $t$  – некоторое положительное целое число), то с помощью данного кода можно исправить все символы, содержащие до  $t$  ошибочных битов, включительно. Если  $d_{\min} \geq 2t$ , то можно исправить все символы, содержащие до  $(t-1)$  ошибочных битов. Кроме того, будут обнаружены все символы с  $t$  ошибочными битами, исправить которые, в общем случае нельзя. Верно и обратное утверждение – каждый код, позволяющий исправлять до  $t$  ошибочных битов, должен удовлетворять условию  $d_{\min} \geq 2t + 1$ . Для каждого кода, позволяющего исправлять до  $(t-1)$  ошибочных битов и обнаруживать все символы с  $t$  ошибками, должно выполняться условие  $d_{\min} \geq 2t$ .

Связь между  $d_{\min}$  и  $t$  можно записать другим способом, выразив максимальное количество битовых ошибок в кодовом слове, гарантированно исправляемых кодом, в таком виде

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \quad (2.12)$$

где  $\lfloor x \rfloor$  – наибольшее целое число, не превышающее  $x$  (например,  $\lfloor 6,3 \rfloor = 6$ ).

Более того, если нас интересует только обнаружение ошибок, но не их исправление, количество обнаруживаемых ошибок  $t$  удовлетворяет следующему равенству

$$t = d_{\min} - 1 \quad (2.13)$$

Последнее выражение можно понять интуитивно, если вспомнить, что  $d_{\min}$  битовых ошибок может изменить одно корректное кодовое слово на другое. При любом меньшем количестве ошибок этого произойти не может.

Выбирая блочный код, нужно учитывать следующие соображения:

1. При данных  $n$  и  $k$  предпочтительным является максимально возможное  $d_{\min}$ ;
2. Код должен быть сравнительно простым для кодирования и декодирования требуя минимальной памяти и времени обработки;
3. Для уменьшения ширины требуемой полосы число избыточных битов ( $n - k$ ) должно быть небольшим;
4. Для снижения уровня ошибок число избыточных битов ( $n - k$ ) должно быть большим.

Очевидно, что два последних требования противоречат друг другу, так что нужно выбирать некоторый компромиссный вариант.

Перед тем как перейти к рассмотрению примеров кодов, обратимся к рисунку 2.8. Графики подобного типа довольно часто приводятся в литературе для демонстрации эффективности разных схем кодирования. Кодирование может использоваться для снижения необходимого значения  $E_b/N_0$ , что позволяет достичь заданного уровня битовых ошибок. Методы кодирования (модуляции) основываются на определении сигнальных посылок, представляющих биты.

Кодирование, рассматриваемое в данном разделе влияет на отношение  $E_b/N_0$ . На рисунке 2.8 кривая справа соответствует системе модуляции без кодирования; в затененной области параметры системы могут быть улучшены. В этой области, при заданном  $E_b/N_0$ , достигается меньшее значение BER. Верно и обратное утверждение – при заданном BER требуется меньшее значение  $E_b/N_0$ . Кривая, расположенная левее, представляет собой типичный результат обработки сигнала кодом со степенью кодирования  $1/2$  (в этом случае число битов данных равно числу контрольных битов). Для частоты возникновения ошибок  $10^{-5}$  использование кодирования позволяет снизить  $E_b/N_0$  на 2,77 дБ. Это улучшение называется эффективностью кодирования. Эффективность кодирования – это снижение необходимого уровня  $E_b/N_0$  для системы с кодированием по сравнению с системой без кодирования (подразумевается одна и та же модуляция) для достижения заданной частоты ошибок.

Необходимо отметить, что частота появления ошибок при коде со степенью  $1/2$  – это частота появления неисправленных ошибок, а  $E_b$  – энергия на один бит данных. Поскольку степень кодирования равна  $1/2$ , каждому биту данных соответствуют два канальных бита, следовательно удельная энергия кодированного бита в два раза меньше удельной энергии

бита данных, т.е. разница  $E_b/N_0$  составляет 3 дБ. Если рассмотреть удельную энергию кодированного бита такой системы, уровень битовых ошибок в канале составит  $2,4 \times 10^{-2}$ , или 0,024.

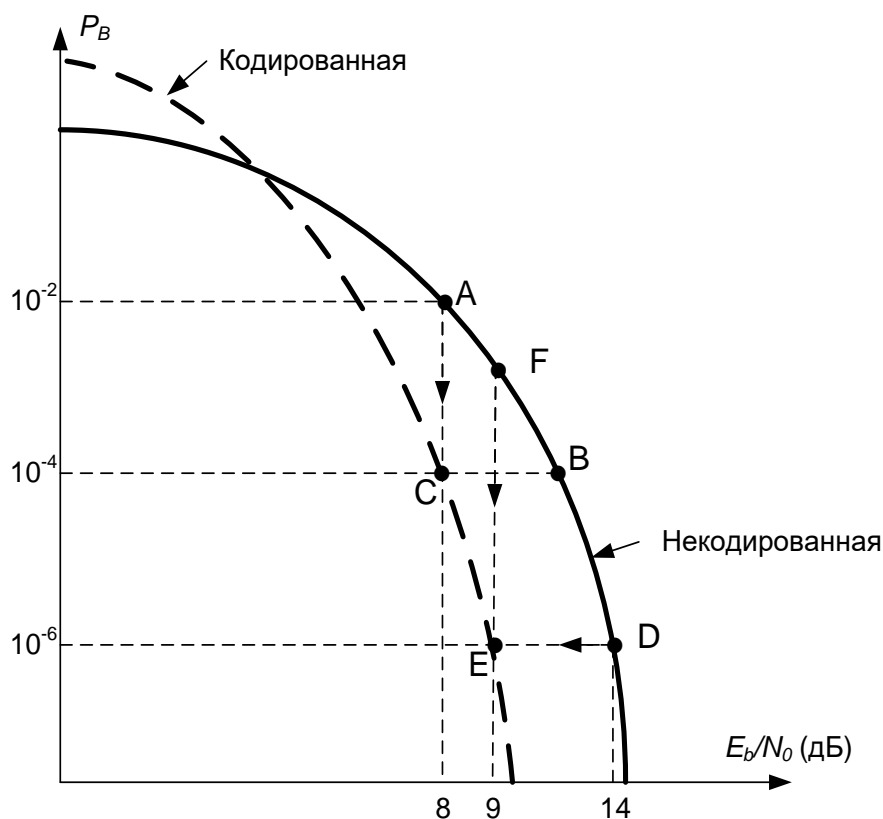


Рисунок 2.8 – Сравнение достоверности передачи при использовании схемы кодирования и без кодирования

В завершение необходимо отметить, что если значение  $E_b/N_0$  – ниже определенного порогового, методы кодирования отрицательно сказываются на производительности системы. В примере, приведенном на рисунке 2.8, это пороговое значение равно приблизительно 5,4 дБ. При более низком значении  $E_b/N_0$  использование дополнительных контрольных битов снижает удельную энергию битов данных, что ведет к увеличению числа ошибок. Если значение  $E_b/N_0$  выше порогового, способность кода исправлять ошибки (эффективность кодирования) позволяет компенсировать снижение  $E_b$  и улучшить работу системы.

Перейдем к рассмотрению конкретных примеров кодов с коррекцией ошибок.

#### *Коды Хэмминга.*

Коды Хэмминга – это семейство блочных кодов с коррекцией ошибок  $(n, k)$ , которые характеризуются следующими параметрами:



Длина блока	$n = 2^m - 1$
Количество битов данных	$k = 2^m - m - 1$
Количество контрольных битов	$n - k = m$
Минимальное расстояние	$d_{\min} = 3$

Здесь  $m \geq 3$ . Коды Хэмминга просты в использовании и легко поддаются анализу, однако редко используются на практике. Рассмотрим этот тип кодов, чтобы проиллюстрировать на их примере некоторые фундаментальные принципы работы блочных кодов.

Коды Хэмминга созданы для исправления 1-битовых ошибок. Для начала определим необходимую длину кода. Коды Хэмминга применяются так же, как методы определения ошибок (рисунок 2.4) – в процессе кодирования сохраняются  $k$  бит данных и добавляются  $(n - k)$  контрольных битов. При декодировании используются две последовательности из  $(n-k)$  бит, одна из которых является кодовым словом входящего сигнала, а другая рассчитывается на основе полученных битов данных. Последовательности побитово сравниваются с помощью логического исключающего ИЛИ. Результат сравнения называют синдромом. Биту синдрома присваивается значение 0, если биты двух последовательностей совпадают, и 1 – в противном случае.

Синдром – это  $(n - k)$ -битовое слово в диапазоне от 0 до  $2^{(n-k)} - 1$ . Значение 0 свидетельствует об отсутствии ошибки. Если же ошибка присутствует, ее местоположение определяется из синдрома. Поскольку ошибочным может быть любой из  $k$  бит данных или  $(n - k)$  проверочных битов, должно выполняться следующее соотношение:

$$2^{(n-k)} - 1 \geq k + (n - k) = n \quad (2.14)$$

Приведенное уравнение определяет количество битов, необходимое для исправления 1-битовой ошибки в слове, содержащем  $k$  бит данных. В таблице 2.2 приводится число контрольных битов, необходимое для последовательностей данных разной длины.

Для удобства генерируемый синдром должен обладать следующими свойствами:

- Если синдром состоит только из нулей – ошибки не обнаружены;
- Если один и только один бит синдрома равен 1 – ошибка присутствует в одном из контрольных битов; в этом случае исправлять ошибку не нужно;
- Если синдром содержит более одного бита со значением 1, он является указателем на положение ошибки в слове для исправления которой указанный бит инвертируется.

Для получения заданных характеристик контрольные биты и биты данных следующим образом комбинируются в  $n$ -битовый блок. Начиная с младшего (крайнего правого) разряда через промежутки, которые соответствуют степеням 2 (т.е. на позиции 1, 2, 4, ...,  $2^{(n-k)}$ ), вводятся контрольные биты Хэмминга. Оставшиеся позиции заполняются битами данных. Для расчета контрольных битов каждый бит данных со значением 1 представляется двоичным значением, соответствующим его положению в последовательности.

Таблица 2.2 – Требования к кодам Хэмминга

Количество битов данных	Исправление 1-битовых ошибок		Исправление 1-битовых ошибок, обнаружение 2-битовых ошибок	
	Количество контрольных битов	Увеличение блока в %	Количество контрольных битов	Увеличение блока в %
8	4	50	5	62,5
16	5	31,25	6	37,5
32	6	18,75	7	21,875
64	7	10,94	8	12,5
128	8	6,25	9	7,03
256	9	3,52	10	3,91

Таким образом, если 9-й бит равен 1, то соответствующее ему значение – 1001. Затем ко всем битам последовательности применяют операцию исключающего ИЛИ, в результате получают биты кода Хэмминга. В приемнике операция исключающего ИЛИ применяется ко всем элементам последовательности со значением 1. В этом случае операция применяется и к контрольным битам, и к битам данных. Поскольку положение контрольных битов определяется степенями 2, операцию исключающего ИЛИ можно применить ко всем информационным элементам последовательности, имеющим значение 1, и коду Хэмминга (состоящему из контрольных битов). Ошибки отсутствуют, если результатом операции будет 0. Если результат отличен от нуля – он представляет собой синдром, значение которого соответствует положению ошибочного бита.

Описанный код известен как код исправления однобитовых ошибок (single-error-correcting – SEC). Одной из разновидностей этого кода является код исправления 1-битовых и обнаружения 2-битовых ошибок (single-error-correcting, double-error-detecting – SEC-DED). Как показано в таблице 3.1, для таких кодов требуется на один бит больше, чем для кодов SEC. Этот дополнительный бит является битом четности для всего кодового блока.

### Циклические коды.

Большинство используемых на практике блочных кодов с коррекцией ошибок относятся к категории циклических. Для циклических кодов справедливо следующее: если  $n$ -битовая последовательность  $c = (c_0, c_1, \dots, c_{n-1})$  является кодовым словом, то последовательность  $(c_{n-1}, c_0, c_1, \dots, c_{n-2})$ , полученная с помощью циклического сдвига  $c$  на одну позицию вправо, также используется в качестве кодового слова. Данный класс кодов можно легко кодировать и декодировать с использованием линейных регистров сдвига с обратной связью (linear feedback shift register – LFSR). Примерами циклических кодов являются коды Боуза-Чоудхури-Хоквенгема (БЧХ) и коды Рида-Соломона.

Реализация циклического кодера как регистра LFSR подобна реализации приведенной на рисунке 2.6 для кодов обнаружения ошибок. Основное отличие состоит в том, что вход кода CRC имеет произвольную длину и в результате получается контрольный код CRC фиксированной длины, тогда как циклический код с коррекцией ошибок генерирует контрольный код  $(n - k)$  бит на основе входной последовательности фиксированной длины ( $k$  бит).

На рисунке 2.9 представлена реализация в виде LFSR декодера циклического блочного кода. Можно сравнить данный рисунок с логикой кодера, приведенной на рисунке 2.6. Отметим, что в кодере  $k$  бит данных используются как вход для получения в регистре сдвига кода длиной  $(n - k)$  бит. Для декодера входом являются полученные  $n$  бит, которые содержат  $k$  бит данных, за которыми следуют  $(n-k)$  контрольных битов. При отсутствии ошибок после первых  $k$  тактов регистр сдвига содержит последовательность контрольных битов, идентичную переданной. После оставшихся  $(n - k)$  тактов регистр сдвига будет содержать код-синдром.

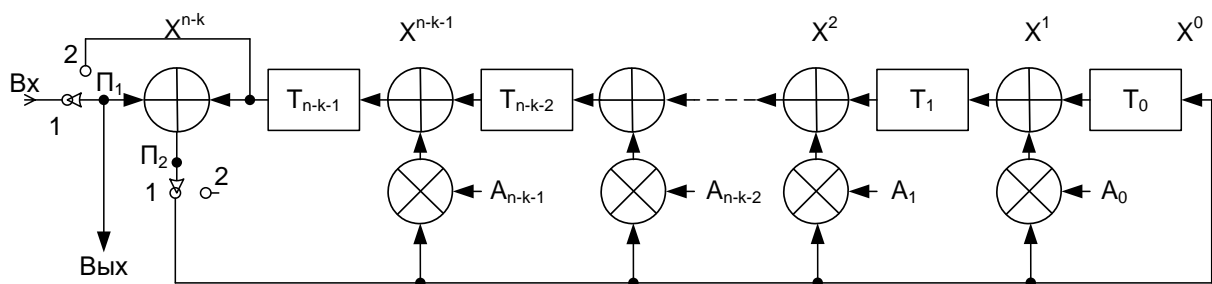


Рисунок 2.9 – Структурная схема декодера циклического кода

Для декодирования циклического кода используется следующая процедура:

1. С помощью полученных битов вычисляется код-синдром. Вычисления проводятся аналогично тому, как кодер обрабатывает биты данных для получения контрольного кода;
2. Если все биты синдрома равны нулю – ошибки отсутствуют;

3. Если синдром отличен от нуля, производится дополнительная обработка для исправления ошибки.

Значение синдромов можно понять, изучив блочный код с использованием полиномов. Как и при проверке четности с избыточностью, конкретный циклический код можно представить полиномиальным делителем, именуемым порождающим многочленом (или генератором). Для кода  $(n, k)$  порождающий многочлен можно записать в таком виде:

$$P(X) = 1 + \sum_{i=1}^{n-k-1} A_i X^i + X^{n-k}, \quad (2.15)$$

где каждый из коэффициентов  $A_i$  может принимать значения 0 или 1, что соответствует двоичному разряду в делителе. Например, для  $P = 11001$  многочлен  $P(X)$  имеет вид  $X^4 + X^3 + 1$ . Аналогично последовательность битов данных представляется полиномом  $D(X)$ , а контрольный код – полиномом  $R(X)$ . Контрольный код определяется следующим образом

$$\frac{X^{n-k} D(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}, \quad (2.16)$$

Т.е. блок данных  $D(X)$  сдвигается влево на  $(n-k)$  бит и делится на  $P(X)$ . В результате получим частное  $Q(X)$  и остаток  $R(X)$  длиной  $(n - k)$  бит. Передаваемый блок – это конкатенация  $D(X)$  и  $R(X)$ :

$$T(X) = X^{n-k} D(X) + R(X), \quad (2.17)$$

При отсутствии ошибок передачи  $T(X)$  должно делиться на  $P(X)$  без остатка, что легко продемонстрировать

$$\frac{T(X)}{P(X)} = \frac{X^{n-k} D(X)}{P(X)} + \frac{R(X)}{P(X)} = \left( Q(X) + \frac{R(X)}{P(X)} \right) + \frac{R(X)}{P(X)} = Q(X) \quad (2.18)$$

Последнее выражение справедливо в соответствии с правилами арифметики по модулю 2 ( $a \oplus a = 0$ ). Следовательно, если ошибки отсутствуют,  $T(X)$  делится на  $P(X)$  без остатка.

Если один или более битов являются ошибочными, полученный блок  $Z(X)$  будет иметь такой вид:

$$Z(X) = T(X) + E(X), \quad (2.19)$$

где  $E(X)$  –  $n$ -битовый полином ошибок, содержащий 1 в каждом двоичном разряде, где в  $Z(X)$  имеется ошибка. Если  $Z(X)$  передается через регистр LFSR, который показан на рисунке 2.9, фактически производится деление  $Z(X)/P(X)$ , что в результате дает синдром  $S(X)$  длиной  $(n - k)$  бит

$$\frac{Z(X)}{P(X)} = B(X) + \frac{S(X)}{P(X)}, \quad (2.20)$$

где  $B(X)$  – частное, а  $S(X)$  – остаток деления. Следовательно,  $S(X)$  является функцией  $Z(X)$ . Каким же образом полученный результат можно использовать для исправления ошибок? Для ответа на этот вопрос, запишем уравнение в развернутой форме

$$\begin{aligned} \frac{Z(X)}{P(X)} &= B(X) + \frac{S(X)}{P(X)}, \\ \frac{T(X) + E(X)}{P(X)} B(X) + \frac{S(X)}{P(X)}, \\ Q(X) + \frac{E(X)}{P(X)} &= B(X) + \frac{S(X)}{P(X)}, \\ \frac{E(X)}{P(X)} &= Q(X) + B(X) + \frac{S(X)}{P(X)}, \end{aligned} \quad (2.21)$$

Видим, что деление  $E(X)/P(X)$  дает тот же остаток, что и  $Z(X)/P(X)$ . Следовательно, значение синдрома  $S(X)$  зависит только от ошибочных битов и не зависит от начальной последовательности битов (переданного значения  $T(X)$ ). Если ошибочные биты  $E(X)$  можно получить из синдрома  $S(X)$ , то ошибки в  $Z(X)$  можно исправить посредством простого сложения

$$Z(X) + E(X) = T(X) + E(X) + E(X) = T(X), \quad (2.22)$$

Поскольку  $S(X)$  зависит только от  $E(X)$ , возможности блочного циклического кода определить очень легко. Синдром состоит из  $(n-k)$  бит, следовательно, он может принимать  $2^{n-k}$  возможных значений. Нулевой синдром указывает на отсутствие ошибок. Следовательно, всего можно исправить  $(2^{n-k} - 1)$  различных ошибочных комбинаций. Чтобы с помощью кода  $(n, k)$  можно было исправить все возможные однобитовые ошибки, должно выполняться неравенство  $n \leq (2^{n-k} - 1)$ . Исправление всех 1- и 2-битовых ошибок требует выполнения следующего неравенства

$$\left( n + \frac{n(n-1)}{2} \right) \leq (2^{n-k} - 1), \quad (2.23)$$

Способ получения  $E(X)$  из  $S(X)$  может зависеть от используемого кода. Наиболее простой подход – построить таблицу, которая ставила бы в соответствие значениям  $E(X)$  значения  $S(X)$ . После этого потребуется простой способ выполнения поиска в такой таблице.

Пример. Рассмотрим код  $(7, 4)$  с порождающим многочленом  $P(X) = X^3 + X^2 + 1$ . Имеем  $7 = 2^3 - 1$ , следовательно с помощью этого кода можно исправить все 1-битовые ошибки. В таблице 2.3,а приводятся все используемые кодовые слова. Отметим, что  $d_{\min} = 3$ , значит, данный код – это код с коррекцией 1-битовых ошибок. Например, для блока данных 1010 имеем  $D(X) = X^3 + X$  и  $X^{n-k}D(X) = X^6 + X^4$ . Деление выполняется согласно правилам деления полиномов

$$\begin{array}{rcl}
 2^6 D(X) \longrightarrow & \begin{array}{r} X^6 + \quad + X^4 \\ X^6 + X^5 + \quad X^3 \\ \hline X^5 + X^4 + X^3 \\ X^5 + X^4 + \quad X^2 \\ \hline X^3 + X^2 \\ X^3 + X^2 \quad + 1 \\ \hline 1 \end{array} & \begin{array}{l} | \quad X^3 + X^2 + 1 \longleftarrow P(X) \\ \hline X^3 + X^2 + 1 \longleftarrow Q(X) \end{array} \\
 & & 1 \longleftarrow R(X)
 \end{array}$$

Рисунок 2.10 – Получение остатка от деления на порождающий полином

Для коррекции ошибок требуется таблица синдромов, показанная в таблице 2.3,б. Например, для последовательности ошибок 1000000  $E(X)$  будет равно  $X^6$ .

$$\begin{array}{rcl}
 Z(X) \longrightarrow & \begin{array}{r} X^6 \\ X^6 + X^5 + \quad X^3 \\ \hline X^5 + \quad + X^3 \\ X^5 + X^4 + \quad X^2 \\ \hline X^4 + X^3 + X^2 \\ X^4 + X^3 + \quad X \\ \hline X^2 + X \end{array} & \begin{array}{l} | \quad X^3 + X^2 + 1 \longleftarrow P(X) \\ \hline X^3 + X^2 + 1 \longleftarrow Q(X) + B(X) \end{array} \\
 & & \longleftarrow S(X)
 \end{array}$$

Рисунок 2.11 – Получение синдрома ошибки

Таким образом,  $S = 110$ . Остальные ячейки табл. 2.3,б вычисляются аналогично. Теперь предположим, что был получен блок 1101101, или  $Z(X) = X^6 + X^5 + X^3 + X^2 + 1$ .

$$\begin{array}{rcl}
 Z(X) \longrightarrow & \begin{array}{r} X^6 + X^5 + \quad X^3 + X^2 + 1 \\ X^6 + X^5 + \quad X^3 \end{array} & \left| \begin{array}{l} X^3 + X^2 + 1 \longleftarrow P(X) \\ X^3 + X^2 + 1 \longleftarrow B(X) \end{array} \\
 \hline
 & X^2 + 1 & \longleftarrow S(X)
 \end{array}$$

Рисунок 2.12 – Получение синдрома из  $Z(X)$

Следовательно,  $S = 101$ . Используя табл. 3.2, б, получаем  $E = 0001000$ . Тогда

$$T = 1101101 \oplus 0001000 = 1100101$$

Итак, согласно табл. 2.3,а, был передан блок данных 1100.

Таблица 2.3 – Циклический код с коррекцией 1-битовых ошибок (7,4)

а) Таблица используемых кодовых слов

Блок данных	Кодовое слово
1	2
0000	0000000
0001	0001101
0010	0010111
0011	0001010
0100	0100011
0101	0101110
0110	0110100
0111	0111001
1000	1000110
1001	1001011
1010	1010001
1011	1011100
1100	1100101
1101	1101000
1110	1110010
1111	1111111

б) Таблица синдромов, соответствующих 1-битовым ошибкам

Ошибочная комбинация	Синдром
0000001	001
0000010	010
0000100	100
Продолжение таблицы 2.3	
0001000	101
0010000	111
0100000	011
1000000	110

*Коды БЧХ.* Коды БЧХ являются одним из наиболее мощных циклических блочных кодов и получили широкое применение в беспроводных приложениях. Для любой пары положительных целых чисел  $m$  и  $t$  существуют двоичные коды БЧХ  $(n, k)$  со следующими параметрами

Длина блока	$n = 2^m - 1$
Количество контрольных битов	$n - k \leq mt$
Минимальное расстояние	$d_{\min} \geq 2t + 1$

С помощью такого кода можно исправить все слова, содержащие  $t$  (или менее) ошибок. Порождающий многочлен кода БЧХ можно создать из множителей полинома  $(X^{2^m-1} + 1)$ . Использование кодов БЧХ предоставляет некоторую свободу выбора параметров (длина блока, степень кодирования).

Коды БЧХ очень важны, поскольку при блоках, длина которых равна порядка несколько сотен, коды БЧХ превосходят своими качествами другие блочные коды с той же длиной блока и степенью кодирования. В наиболее часто применяемых кодах БЧХ используется двоичный алфавит и блок кодового слова длиной  $n = 2^m - 1$ , где  $m = 3, 4, \dots$

В таблице 2.4 перечислены параметры БЧХ для кодов длиной до  $2^8 - 1$ . В таблице 2.5 приводятся некоторые порождающие многочлены БЧХ.

Таблица 2.4 – Параметры кодов БЧХ

$n$	$k$	$t$	$n$	$k$	$t$	$n$	$k$	$t$	$n$	$k$	$t$	$n$	$k$	$t$
7	4	1	63	30	6	127	64	10	255	207	6	255	99	23
15	11	1		24	7		57	11		199	7		91	25
	7	2		18	10		50	13		191	8		87	26
	5	3		16	11		43	14		187	9		79	27
31	26	1		10	13		36	15		179	10		71	29
	21	2		7	15		29	21		171	11		63	30
	16	3	127	120	1		22	23		163	12		55	31
	11	5		113	2		15	27		155	13		47	42



Продолжение таблицы 2.4														
63	6	7		106	3		8	31		147	14		45	43
	57	1		99	4	255	247	1		139	15		37	45
	51	2		92	5		239	2		131	18		29	47
	45	3		85	6		231	3		123	19		21	55
	39	4		78	7		223	4		115	21		13	59
	36	5		71	9		215	5		107	22		9	63

Таблица 2.5 – Порождающие полиномы кодов БЧХ

$n$	$k$	$t$	$P(X)$
7	4	1	$X^3 + X + 1$
15	11	1	$X^4 + X + 1$
15	7	2	$X^8 + X^7 + X^6 + X^4 + 1$
15	5	3	$X^{10} + X^8 + X^5 + X^4 + X^2 + X + 1$
31	26	1	$X^5 + X^2 + 1$
31	21	2	$X^{10} + X^9 + X^8 + X^6 + X^5 + X^3 + 1$

Для декодирования сигналов БЧХ было разработано большое количество методов, требующих меньше памяти, чем прямой поиск в таблице. Идея одной из наиболее простых схем стоит в вычислении полинома локализации ошибки с последующим нахождением его корней. Сложность данного алгоритма увеличивается как квадрат количества ошибок, которые нужно исправить.

*Коды Рида-Соломона.* Коды Рида-Соломона (Reed-Solomon codes – RS codes) – это широко используемый подкласс недвоичных кодов БХЧ. При использовании кодов Рида-Соломона данные обрабатываются порциями по  $m$  бит, именуемыми символами. Код  $(n, k)$  характеризуется следующими параметрами

Длина символа	$m$ бит
Длина блока	$n = (2^m - 1)$ символов = $m(2^m - 1)$ бит
Длина блока данных	$k$ символов
Размер контрольного кода	$n - k = 2t$ символов = $m(2t)$ бит
Минимальное расстояние	$d_{\min} = (2t + 1)$ символов

Таким образом, алгоритм кодирования расширяет блок  $k$  символов до размера  $n$ , добавляя  $(n-k)$  избыточных контрольных символов. Как правило,  $m$  является степенью 2; широко используется значение  $m = 8$ .

**Пример.** Пусть  $t = 1$ ,  $m = 2$ . Обозначая символы как 0, 1, 2, 3, их двоичные эквиваленты можно записать как  $0 = 00$ ;  $1 = 01$ ;  $2 = 10$ ;  $3 = 11$ . Код имеет следующие параметры:

$$n = 2^2 - 1 = 3 \text{ символа} = 6 \text{ бит},$$

$$n - k = 2 \text{ символа} = 4 \text{ бит}.$$

С помощью данного кода можно исправить любой пакет ошибок, который искажает 2-битовый символ.

Коды Рида-Соломона удобны для исправления пакетов ошибок. Данный тип кодов характеризуется высокоэффективным использованием избыточности, длина блоков и размеры символов могут легко приспособливаться под сообщения разных размеров. Кроме того, для таких кодов существуют эффективные методы декодирования [1. 2].

*Известные блочные коды.* На рисунке 2.13 приведены графики зависимости вероятности ошибки в декодированном бите от вероятности ошибки в канальном символе, по которым сравниваются разные блочные коды. Для кодов Хэмминга на графике взяты значения  $m = 3, 4$  и  $5$  или  $(n, k) = (7,4), (15,11), (31,26)$ . Для описания гауссового канала с использованием когерентной демодуляции сигналов BPSK, вероятность ошибки в канальном символе можно выразить через  $E_c/N_0$ :

$$p = Q\left(\sqrt{\frac{2E_c}{N_0}}\right) \quad (2.24)$$

Здесь  $E_c/N_0$  – отношение энергии кодового символа к. Чтобы связать  $E_c/N_0$  с отношением энергии бита информации к спектральной плотности мощности шума ( $E_b/N_0$ ), используем следующее выражение

$$\frac{E_c}{N_0} = \left(\frac{k}{n}\right) \frac{E_b}{N_0}, \quad (2.25)$$

Для кодов Хэмминга уравнение принимает следующий вид:

$$\frac{E_c}{N_0} = \left( \frac{2^m - 1 - m}{2^m - 1} \right) \frac{E_b}{N_0}, \quad (2.26)$$

Объединяя уравнения (2.25) и (2.26), вероятность ошибки  $P_b$  при когерентной демодуляции сигналов BPSK в гауссовом канале можно выразить как функцию  $E_b/N_0$ . Результаты для различных типов блочных кодов отображены на рисунке 2.14. Для кодов Хэмминга взяты следующие значения  $(n, k) = (7, 4), (15, 11), (31, 26)$ .

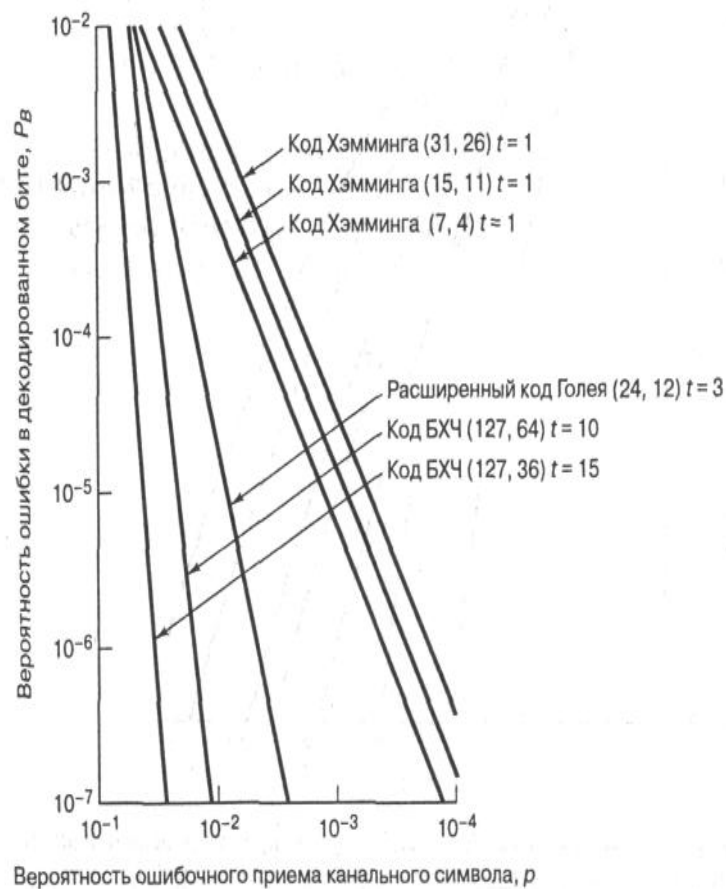


Рисунок 2.13 – Зависимость вероятности битовой ошибки от вероятности ошибки в канальном символе для нескольких блочных кодов

На рисунке 2.15 показаны расчетные характеристики кодов БЧХ для когерентно демодулированного сигнала BPSK с жестким и мягким декодированием. Мягкое декодирование для блочных кодов не применяется из-за своей сложности, хотя оно и дает увеличение эффективности кодирования порядка 2 дБ по сравнению с жестким декодированием. При данной степени кодирования вероятность ошибки при декодировании уменьшается с ростом длины блока  $n$ . Таким образом, при данной степени

кодирования интересно рассмотреть необходимую длину блока для сравнения характеристик жесткого и мягкого декодирования. Из рисунка видно, что при фиксированной степени кодирования и жестком декодировании кода БЧХ длиной  $8n$  или более наблюдаются лучшие характеристики, чем при мягком декодировании кода БЧХ длиной  $n$ . Существует специальный подкласс кодов БЧХ (которые были разработаны раньше кодов БЧХ), который является *недвоичным* набором, это коды Рида-Соломона (Reed-Solomon code).

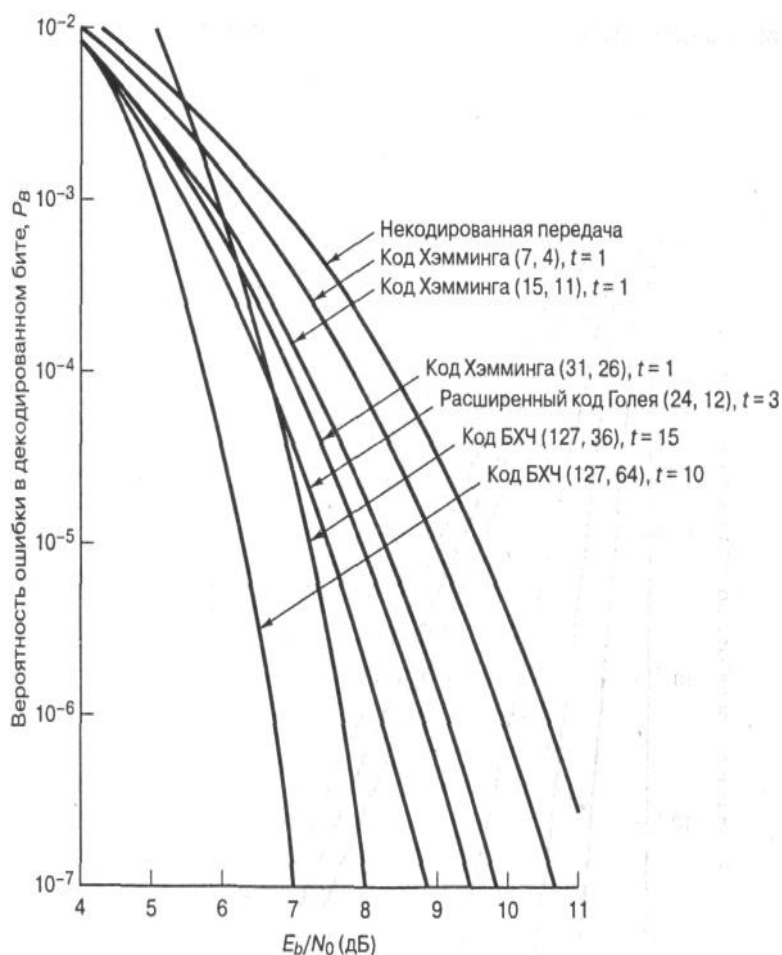


Рисунок 2.14 – Зависимость  $P_B$  от  $E_b/N_0$  при когерентной демодуляции сигналов BPSK в гауссовом канале для нескольких блочных кодов

На рисунке 2.16 показана зависимость  $P_B$  от вероятности появления ошибки в канальном символе  $p$ , полученная из уравнений для различных ортогональных 32-ричных кодов Рида-Соломона с возможностью коррекции  $t$  ошибочных бит в символе и  $n = 31$  (тридцать один 5-битовый символ в кодовом блоке). На рисунке 2.17 показана зависимость  $P_B$  от  $E_b/N_0$  для таких систем кодирования при использовании модуляции MFSK и некогерентной демодуляции в канале с AWGN. Для кодов Рида-Соломона вероятность

появления ошибок является убывающей степенной функцией длины блока,  $n$ , а сложность декодирования пропорциональна небольшой степени длины блока. Иногда коды Рида-Соломона применяются в каскадных схемах. В таких системах внутренний сверточный декодер сначала осуществляет некоторую защиту от ошибок за счет мягкой схемы решений на выходе демодулятора; затем сверточный декодер передает данные, оформленные согласно жесткой схеме, на внешний декодер Рида-Соломона, что снижает вероятность появления ошибок [3].

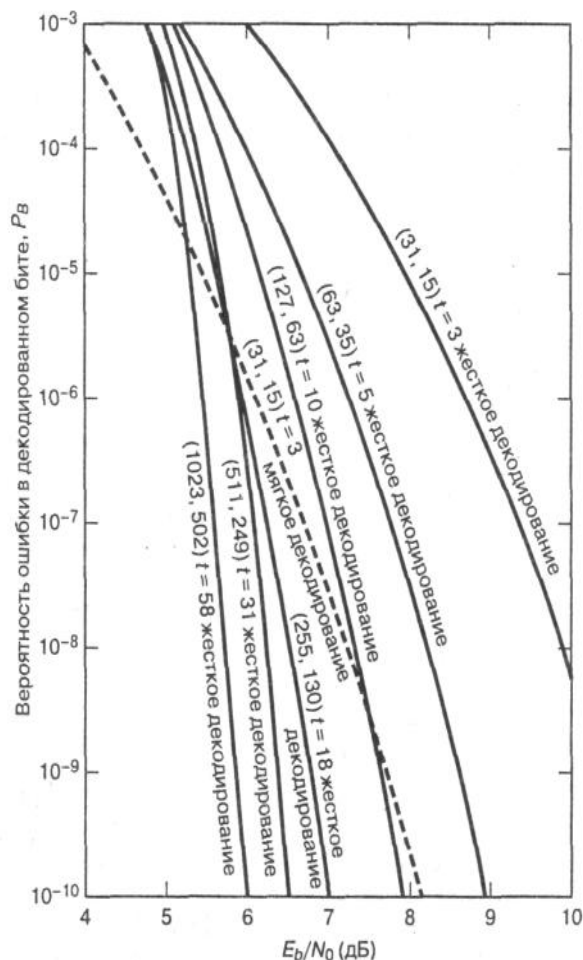


Рисунок 2.15 – Зависимость  $P_B$  от  $E_b/N_0$  для когерентно демодулируемого сигнала BPSK в гауссовом канале с использованием кодов БХЧ

### 2.4.3 Сверточные коды.

Блочные коды являются одной из двух категорий кодов с коррекцией ошибок, широко используемых при беспроводной передаче. Вторая

категория – это сверточные коды. Блочный код  $(n, k)$  обрабатывает данные блоками по  $k$  бит, генерируя на выходе блок из  $n$  бит ( $n > k$ ) для каждого  $n$ -битового блока на входе.

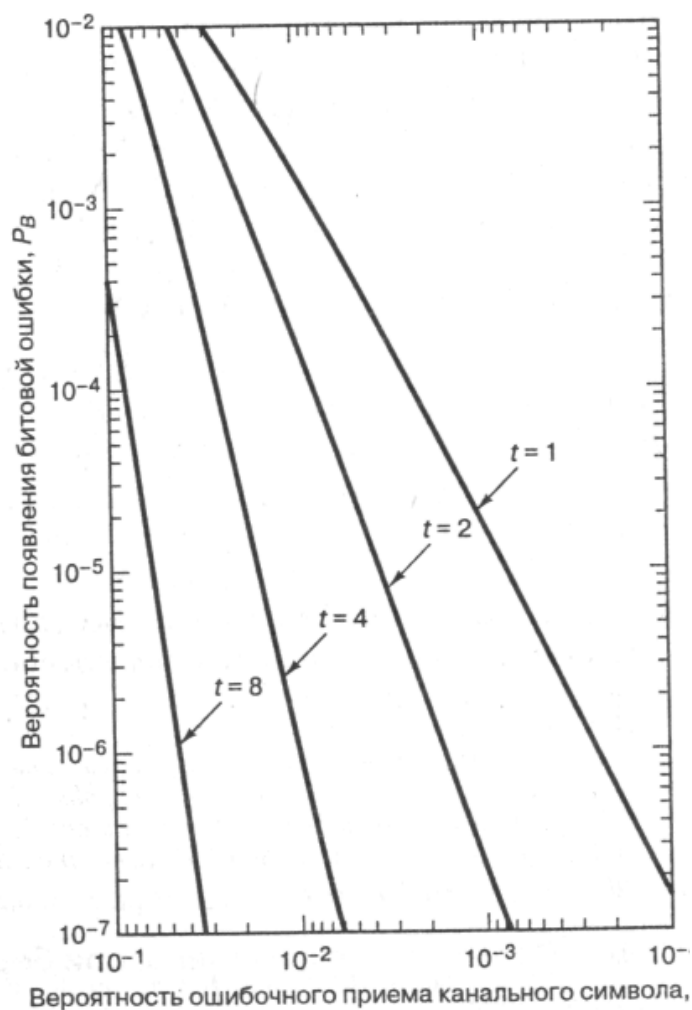


Рисунок 2.16 – Зависимость  $P_B$  от  $p$  для различных ортогональных 32-ричных кодов Рида-Соломона с возможностью коррекции  $t$  бит в символе и  $n = 31$

Если прием и передача данных происходят относительно непрерывным потоком, то блочный код (в частности, с большим значением  $n$ ) может быть не так удобен, как код, который генерирует избыточные биты непрерывно. В последнем случае обнаружение и исправление ошибок выполняется непрерывно, и именно в этом состоит преимущество сверточных кодов.

Сверточный код задается тремя параметрами:  $n$ ,  $k$  и  $K$ . Код  $(n, k, K)$  обрабатывает входящие данные порциями по  $k$  бит и генерирует выходную последовательность, состоящую из  $n$  бит для каждого  $k$  бит входа. До этого момента принципы работы сверточных и блочных кодов не отличаются. Для сверточных кодов  $n$  и  $k$ , как правило, являются очень малыми числами. Разница между двумя типами кодов состоит в том, что сверточные коды

используют память, которая характеризуется длиной кодового ограничения  $K$ . По сути, текущая  $n$ -битовая выходная последовательность кода  $(n, k, K)$  зависит не только от значений текущего входного блока, состоящего из  $k$  бит, но также и от предыдущих  $(K - 1)$   $k$ -битовых блоков. Следовательно, текущая выходная  $n$ -битовая последовательность является функцией последних  $(K \times k)$  входных битов.

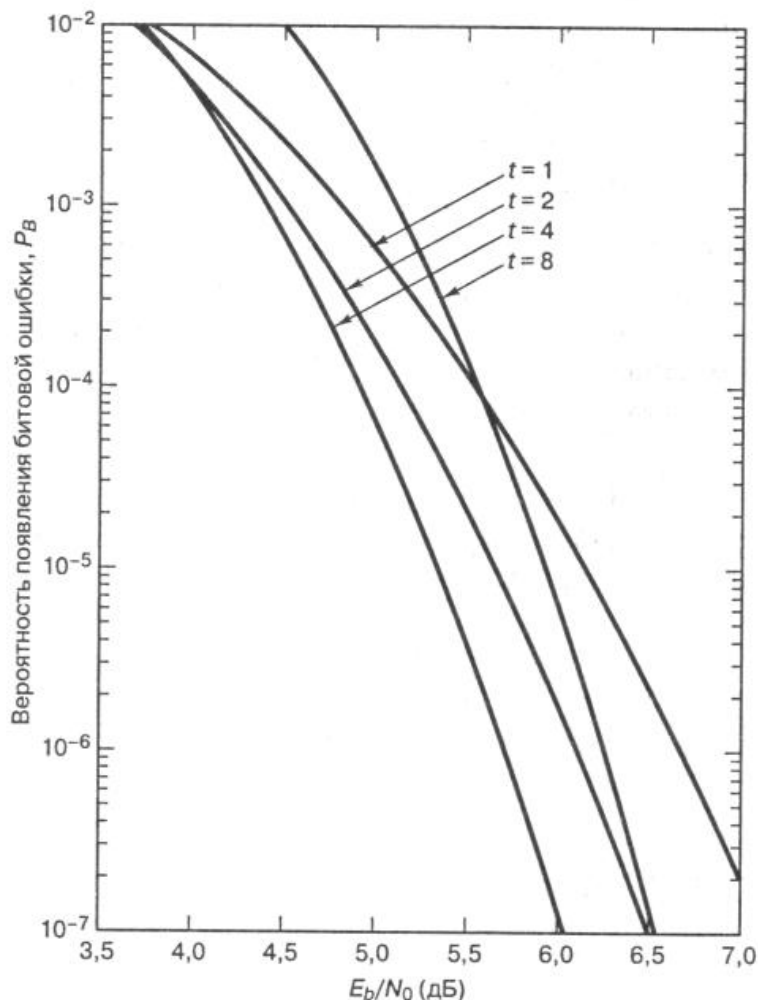


Рисунок 2.17 – Зависимость  $P_b$  от  $E_b/N_0$  для различных ортогональных кодов Рида-Соломона с возможностью коррекции  $t$  бит в символе и  $n = 31$ , при 32-ричной модуляции MFSK в канале AWGN

Принципы работы сверточных кодов удобно рассмотреть на конкретном примере, представленном на рисунке 2.18. Здесь приведены два альтернативных представления кода. Рисунок 2.18, *а* – это регистр сдвига, который наиболее удобен для описания и реализации процесса кодирования. Рисунок 2.18, *б* – это эквивалентное представление, удобное для изучения процесса декодирования.

Для кода  $(n, k, K)$  регистр сдвига содержит последние  $(K \cdot k)$  входных битов; в исходном состоянии все ячейки регистра содержат нули. Кодер генерирует  $n$  выходных битов, после чего наиболее "старые"  $k$  бит регистра стираются и вводится новая  $k$ -битовая последовательность. Хотя выходные  $n$  бит зависят от  $(K \times k)$  входных битов, степень кодирования равна отношению  $k$  входных битов к  $n$  выходных битов. Следовательно, как и для блочного кода, степень кодирования равна  $k/n$ . Наиболее широко используемые двоичные кодеры имеют  $k = 1$ ; соответственно, длина регистра такого кодера равна  $K$ . В рассматриваемом примере (рисунок 2.18, а) используется код  $(2, 1, 3)$ . Здесь кодер преобразует входной бит  $u_n$  в два выходных бита –  $v_{n1}$  и  $v_{n2}$ , используя три последних полученных бита. Первый сгенерированный бит поступает из верхнего логического контура ( $v_{n1} = u_n \oplus u_{n-1} \oplus u_{n-2}$ ) а второй – из нижнего ( $v_{n2} = u_n \oplus u_{n-2}$ ).

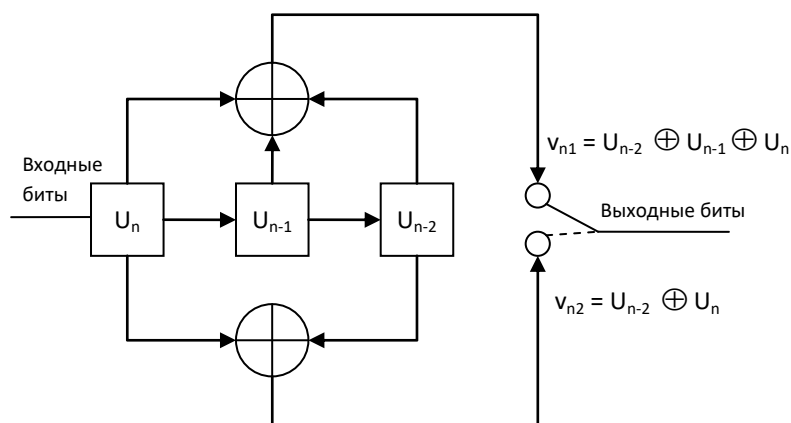
Для любого данного  $k$ -битового входа существует  $2^{k(K-1)}$  различных функций, отображающих  $k$  входных битов в  $n$  выходных. Решение о том, какая из этих функций будет использована, зависит от истории последних  $(K - 1)$   $k$ -битовых входных блоков. Следовательно, сверточный код можно представить как конечный автомат. Автомат имеет  $2^{k(K-1)}$  различных состояний, переход между которыми определяется последними  $k$  входными битами и дает  $n$  выходных битов. Начальное состояние автомата – все нули. Для примера приведенного на рисунок 2.18, б существует 4 состояния, по одному для каждой возможной комбинации последних двух битов. Следующий входной бит инициирует переход и дает два выходных бита. Например, если последние два бита — 10 ( $u_{n-1} = 1$ ;  $u_{n-2} = 0$ ), а следующий бит равен 1 ( $u_n = 1$ ), то текущим состоянием будет  $b(10)$ , а следующим состоянием –  $d(11)$ . Выход имеет такой вид

$$\begin{aligned}v_{n1} &= u_{n-2} \oplus u_{n-1} \oplus u_n = 0 \oplus 1 \oplus 1 = 0 \\v_{n2} &= 0 \oplus 1 = 1\end{aligned}$$

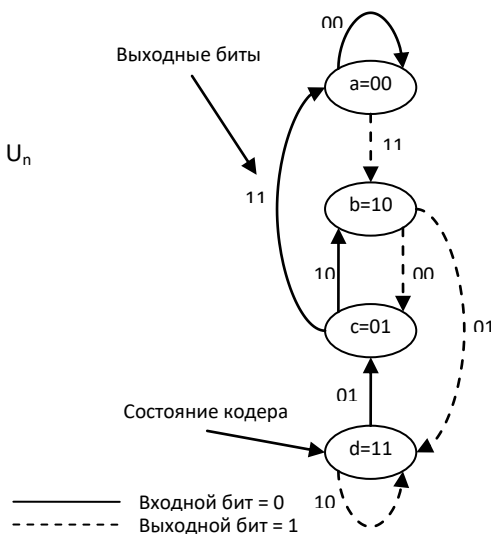
#### *Декодирование свёрточных кодов.*

Чтобы описать процесс декодирования, можно расширить диаграмму состояний, показав в кодере хронологическую последовательность битов. Если диаграмма состояний расположена вертикально, как на рисунке 2.18, б, то расширенная диаграмма называемая решетчатой, строится путем воспроизведения состояний и представления переходов между состояниями в горизонтальном направлении слева направо, в соответствии с течением времени или посредством ввода данных (рисунок 2.19).





а) Регистр сдвига кодера



б) Диаграмма состояний кодера

Рисунок 2.18 – Сверточный кодер (2, 1, 3)

Если длина кодового ограничения  $K$  велика, то решетчатая диаграмма будет слишком громоздкой. В таком случае для отображения переходов можно использовать  $2^{K-2}$  упрощенных решетчатых фрагментов. На рисунке 2.20 приводится подобное отображение для кода (2, 1, 7). Здесь показано каждое состояние кодера и даны определения всех ветвей диаграммы.

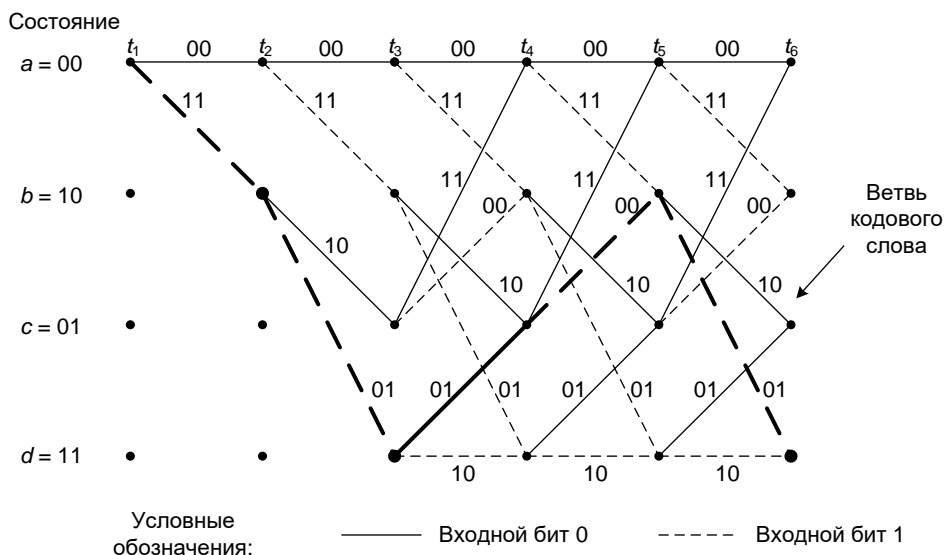


Рисунок 2.19 – Решетчатая диаграмма для кодера, изображенного на рисунке 2.18

Любая корректная выходная последовательность определяется маршрутом в решетчатой диаграмме. Для рассматриваемого примера маршрут a-b-c-b-d-c генерируется входной последовательностью 1011000 и дает на выходе последовательность 11 10 00 01 01 11 00. При некорректном переходе, например a-c, декодер пытается исправить ошибку, определив, какому входу с наибольшей вероятностью соответствует некорректный выход.

Для сверточных кодов разработано большое количество алгоритмов исправления ошибок. Пожалуй, важнейшим из них является алгоритм Витерби (Viterbi code). По сути, метод Витерби – это сравнение полученной последовательности со всеми возможными переданными последовательностями. Алгоритм подбирает маршрут через решетчатую диаграмму, кодированная версия которого отличается от полученной наименьшим числом элементов. Как только выбран правильный маршрут, декодер может восстановить входные биты данных из выходных битов кода.

В зависимости от метрики, используемой для определения отличия полученной и истинной последовательностей, существует несколько разновидностей алгоритма Витерби. Для представления идеи алгоритма используем широко распространенную метрику – расстояние Хэмминга.

Представим полученную закодированную последовательность в виде слова  $w = w_0w_1w_2, \dots$  и попытаемся найти наиболее вероятный маршрут через решетчатую диаграмму. Для каждого момента времени  $i$  и для каждого состояния перечислим активный маршрут (или маршруты) через диаграмму к заданному состоянию. Под активным маршрутом подразумевают корректный маршрут через решетчатую диаграмму, для которого расстояние Хэмминга до полученного слова минимально до момента времени  $i$ . В момент времени  $i$  пометим каждое состояние расстоянием от его активного маршрута до полученного слова.

Используется следующее соотношение

$$\text{расстояние до полученного слова} = \text{расстояние на предыдущем шаге} + \text{расстояние для последнего перехода}.$$

Алгоритм выполняется в  $(b + 1)$  шаг, где  $b$  – предопределенный размер окна. Для кода  $(n, k, K)$  декодирование первого выходного  $n$ -битового блока  $(w_0w_1w_2\dots w_{n-1})$  проходит следующим образом.

- Шаг 0. Начальное состояние решетки в момент времени 0 помечается нулем, поскольку на этот момент маршруты не отличаются.

Шаг  $i+1$ . Для каждого состояния  $S$  в момент времени  $(i + 1)$  с помощью уравнения находятся все ведущие к нему активные маршруты. Состояние  $S$  помечается длиной данного маршрута (маршрутов).



- Шаг  $b$ . Алгоритм прекращается в момент времени  $b$ . Если в этот момент времени все активные маршруты начинаются с одного отрезка, и метка этого отрезка –  $x_0x_1x_2\dots x_{n-1}$  первый кодовый блок  $w_0w_1w_2\dots w_{n-1}$  исправляется на  $x_0x_1x_2\dots x_{n-1}$ . Если же существуют два активных начальных отрезка, ошибка является неисправимой.

После принятия и исправления, при необходимости, первого кодового блока окно декодирования сдвигается на  $n$  бит вправо, после чего выполняется декодирование следующего блока.

Сверточные коды дают возможность достичь хорошей производительности в каналах с шумом, где искаженной является большая доля переданных битов. Следовательно, данные коды все шире используются в беспроводных приложениях [1. 2].

Как правило, декодирование по алгоритму Витерби используется в двоичном канале с жестким или мягким 2-битовым квантованным выходом. Длина кодового ограничения варьируется от 3 до 9, причем степень кодирования кода редко оказывается меньше  $1/3$ , и память путей составляет несколько длин кодового ограничения. Памятью путей называется глубина входных битов, которая сохраняется в декодере. На рисунке 2.21 показаны характерные результаты моделирования достоверности передачи при декодировании по алгоритму Витерби с жесткой схемой квантования [3]. Каждое увеличение длины кодового ограничения приводит к улучшению требуемого значения  $E_b/N_0$  на величину, равную 0,5 дБ, при  $P_B = 10^{-5}$ .

На рисунке 2.22 показаны типичные кривые, отображающие зависимость  $P_B$  от  $E_b/N_0$  для двух распространенных схем – декодирования по алгоритму Витерби и последовательного декодирования. Здесь сравниваются их характеристики при использовании когерентной демодуляции BPSK в канале с AWGN. Сравниваются кривые для декодирования по алгоритму Витерби (степень кодирования  $1/2$  и  $1/3$ ,  $K = 7$ , жесткое декодирование), декодирования по алгоритму Витерби (степень кодирования  $1/2$  и  $1/3$ ,  $K = 7$ , мягкое декодирование) и последовательного декодирования (степень кодирования  $1/2$  и  $1/3$ ,  $K = 41$ , жесткое декодирование). Из рисунка 2.22 можно видеть, что при последовательном декодировании можно достичь эффективности кодирования порядка 8 дБ при  $P_B = 10^{-6}$ . Поскольку в работе Шеннона (Shannon) предсказывается потенциальная эффективность кодирования около 11 дБ, по сравнению с некодированной передачей с модуляцией BPSK, похоже, что, в основном, теоретически достижимые возможности уже получены.

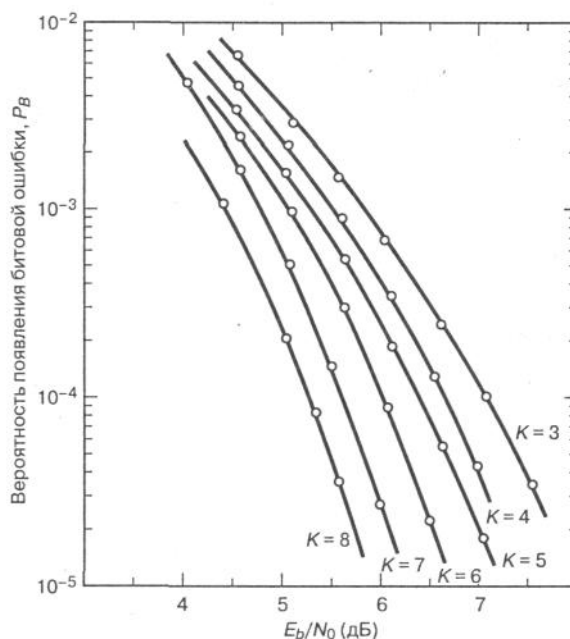


Рисунок 2.21 – Зависимость вероятности появления битовой ошибки от  $E_b/N_0$  при степени кодирования кодов 1/2; используется когерентная модуляция BPSK в канале BSC, декодирование согласно алгоритму Витерби и 32-битовая память путей

#### 2.4.4 Чередование битов

В предыдущих разделах мы подразумевали, что у канала отсутствует память, поскольку рассматривались коды, которые должны были противостоять случайным независимым ошибкам. Канал с памятью – это такой канал, в котором проявляется взаимная зависимость ухудшений передачи сигнала.

Канал, в котором проявляется замирание вследствие многолучевого распространения, когда сигнал поступает на приемник по двум или более путям различной длины, будет примером канала с памятью. Следствием является различная фаза сигналов, и в итоге, суммарный сигнал оказывается искаженным. Таким эффектом обладают каналы мобильной беспроводной связи, так же как ионосферные и тропосферные каналы. В некоторых каналах также имеются коммутационные и другие импульсные помехи (например, телефонные каналы или каналы с создаваемыми импульсными помехами). Все эти ухудшения коррелируют во времени и, в результате, дают статистическую взаимную зависимость успешно переданных символов. Иными словами, искажения вызывают ошибки, имеющие вид пакетов, а не отдельных изолированных ошибок.

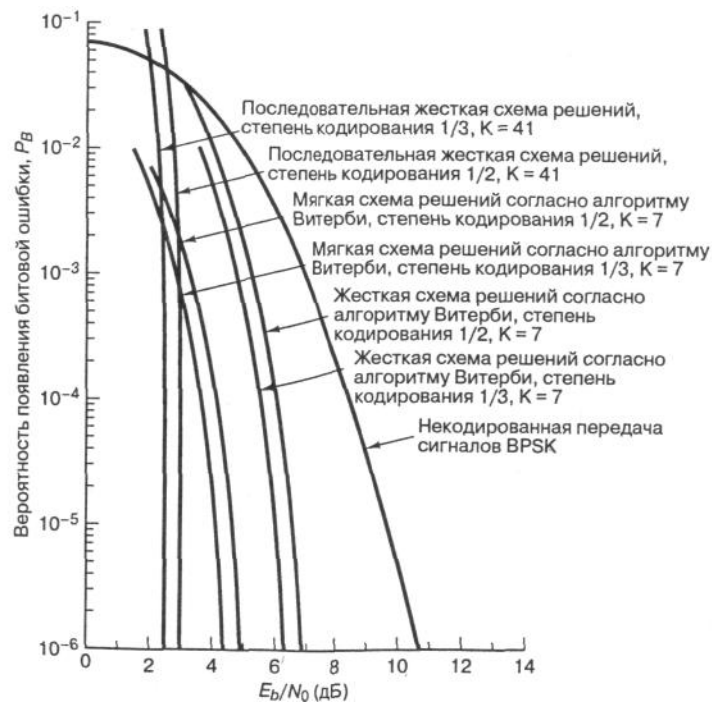


Рисунок 2.22 – Вероятности появления битовых ошибок для различных схем декодирования по алгоритму Витерби и последовательного декодирования при когерентной модуляции BPSK в канале с AWGN

Если канал имеет память, то ошибки не являются независимыми, одиночными случайно распределенными. Большинство блочных и сверточных кодов разрабатывается для борьбы с независимыми случайными ошибками. Влияние канала с памятью, на кодированный таким образом сигнал приведет к ухудшению достоверности передачи. Существуют схемы кодирования для каналов с памятью, но наибольшую проблему в этом кодировании представляет расчет точных моделей сильно нестационарных статистик таких каналов. Подход, при котором требуется знать только объем памяти канала, а не его точное статистическое описание, использует временное разнесение или чередование битов.

Чередование битов кодированного сообщения перед передачей и обратная операция после приема приводят к рассеиванию пакета ошибок во времени: таким образом, они становятся для декодера случайно распределенными. Поскольку в реальной ситуации память канала уменьшается с временным разделением, идея, лежащая в основе метода чередования битов, 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 канале с 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 битового перераспределения должны быть известны приемнику, чтобы иметь возможность выполнить восстановление порядка битов перед декодированием.

Идея чередования битов используется во всех блочных и сверточных кодах. Обычно применяются два типа устройств чередования – блочные и сверточные.

#### *Блочное чередование.*

Блочное устройство чередования принимает кодированные символы блоками кодера, переставляет их, а затем передает измененные символы на модулятор. Как правило, перестановка блоков завершается заполнением столбцов матрицы  $M$  строками и  $N$  столбцами ( $M \times N$ ) кодированной последовательности. После того как матрица полностью заполнена, символы подаются на модулятор (по одной строке за раз), а затем передаются по каналу. В приемнике устройство восстановления выполняет обратные операции; оно принимает символы из демодулятора восстанавливает исходный порядок битов и передает их на декодер. Символы поступают в массив устройства восстановления по строкам и считываются по столбцам. Ниже перечисляются наиболее важные характеристики блочного устройства:

1. Пакет ошибок, который содержит меньше  $N$  последовательных канальных символов, дает на выходе устройства восстановления исходного порядка символов ошибки, разнесенные между собой, по крайней мере, на  $M$  символов;

2. Пакет из  $bN$  ошибок, где  $b > 1$ , дает на выходе устройства восстановления пакет, который содержит не меньше  $\lceil b \rceil$  символьных ошибок. Каждый из пакетов ошибок отделен от другого не меньше, чем на  $M - \lfloor b \rfloor$  символов. Запись  $\lceil x \rceil$  означает наименьшее целое число, не меньшее  $x$ , а запись  $\lfloor x \rfloor$  – наибольшее целое число, не превышающее  $x$ ;

3. Периодическая последовательность одиночных ошибок, разделенных  $N$  символами, дает на выходе устройства восстановления одиночные пакеты ошибок длиной  $M$ ;

4. Прямая задержка между устройствами чередования и восстановления равна приблизительно длительности  $2MN$  символов. Если быть точным, перед тем как начать передачу, нужно заполнить лишь  $M(N - 1) + 1$  ячеек памяти (как только будет внесен первый символ последнего столбца массива  $M \times N$ ). Соответствующее время нужно приемнику, чтобы начать декодирование. Значит, минимальная прямая задержка будет составлять длительность  $(2MN - 2M + 2)$  символов, не учитывая задержку на передачу по каналу;

5. Необходимая память составляет  $MN$  символов для каждого объекта (устройств чередования и восстановления исходного порядка). Однако массив  $M \times N$  нужно заполнить (по большей части) до того, как он будет

считан. Для каждого объекта нужно предусмотреть память для  $2MN$  символов, чтобы опорожнить массив  $M \times N$ , пока другой будет наполняться, и наоборот.

#### *Сверточное чередование.*

Сверточные устройства чередования были предложены Рамси (Ramsey) и Форни (Forney). Схема, предложенная Форни, показана на рисунке 2.23. Кодовые символы последовательно подаются в блок из  $N$  регистров; каждый последующий регистр может хранить на  $J$  символов больше, чем предыдущий. Нулевой регистр не предназначен для хранения (символ сразу же передается). С каждым новым кодовым символом коммутатор переключается на новый регистр, и кодовый символ подается на него до тех пор, пока наиболее старый кодовый символ в регистре не будет передан на модулятор/передатчик. После  $(N - 1)$ -го регистра коммутатор возвращается к нулевому регистру и повторяет все снова. После приема операции повторяются в обратном порядке. И вход, и выход устройств чередования и восстановления должны быть синхронизированы.

На рисунке 2.23 показан пример простого сверточного четырехрегистрового ( $J = 1$ ) устройства чередования, загруженного последовательностью кодовых символов. Одновременно представлено синхронизированное устройство восстановления, которое передает обработанные символы на декодер. Рабочие характеристики сверточного устройства чередования сходны с параметрами блочного устройства.

Важным преимуществом сверточного устройства перед блочным является то, что при сверточном чередовании прямая задержка составляет  $M(N - 1)$  символов при  $M = NJ$ , а требуемые объемы памяти -  $M(N - 1)/2$  на обоих концах канала. Очевидно, что требования к памяти и время задержки снижаются вдвое, по сравнению с блочным чередованием [1].

### **2.4.5 Скремблирование**

Скремблирование наряду с избыточным кодированием является способом логического кодирования информации. Одна из задач скремблирования – уменьшить длину последовательности нулей или единиц входного сигнала, поскольку длинная последовательность может вызвать проблемы синхронизации на приемном конце.

Методы скремблирования заключаются в побитном вычислении результирующего кода на основании битов исходного кода и полученных в предыдущих тактах битов результирующего кода. Например, скремблер может реализовывать следующее соотношение

$$B_i = A_i B_{i-3} B_{i-5} \quad (2.27)$$



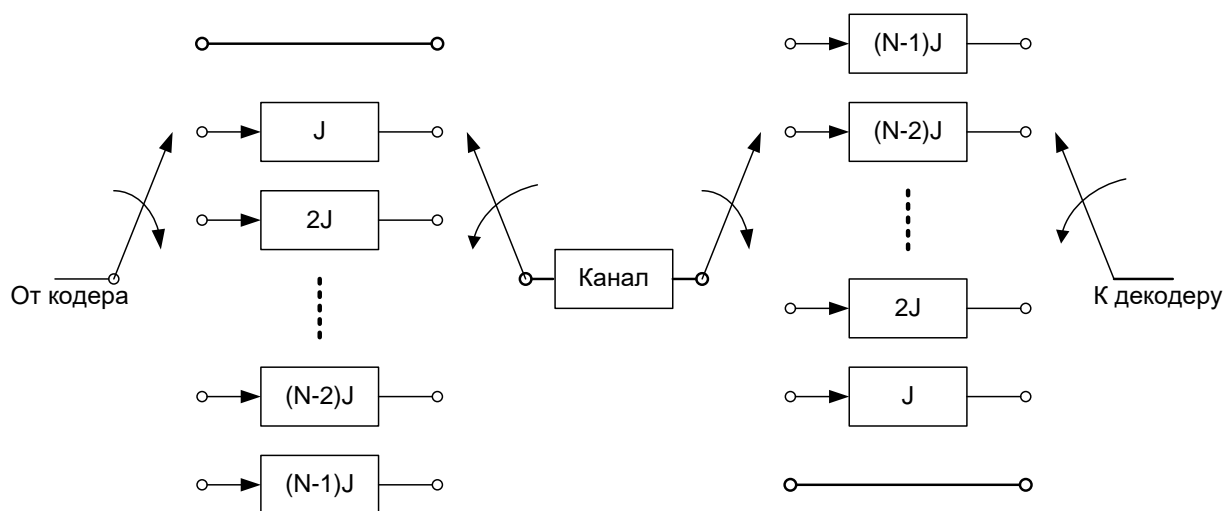


Рисунок 2.23 – Реализация регистра сдвига для сверточного устройства чередования/восстановления

Здесь  $B_i$  – двоичная цифра результирующего кода, полученная на  $i$ -м такте работы скремблера,  $B_{i-3}$  и  $B_{i-5}$  – двоичные цифры результирующего кода, полученные на предыдущих тактах работы скремблера, соответственно на 3 и на 5 тактов ранее текущего такта, и объединенные операцией исключающего ИЛИ.

После получения результирующей последовательности приемник передает ее дескремблеру, который восстанавливает исходную последовательность на основании обратного соотношения

$$C_i = B_i B_{i-3} B_{i-5} = (A_i B_{i-3} B_{i-5}) B_{i-3} B_{i-5} = A_i \quad (2.28)$$

Различные алгоритмы скремблирования отличаются количеством слагаемых, дающих цифру результирующего кода, и сдвигом между слагаемыми. Так, в сетях ISDN при передаче данных от сети к абоненту используется преобразование со сдвигами в 5 и 23 позиции, а при передаче данных от абонента в сеть – со сдвигами 18 и 23 позиции.

## Контрольные вопросы

1. Назовите виды дискретных каналов связи
2. Почему проверка четности не является надежным средством обнаружения ошибок?
3. В каком случае, при использовании циклической проверки четности с избыточностью ошибка может не обнаружиться?
4. Почему процедура ARQ не подходит для использования в беспроводных системах передачи?
5. Почему, если значение  $E_b/N_0$  – ниже определенного порогового, методы кодирования отрицательно сказываются на производительности системы?
6. Назовите основные свойства генерируемого синдрома в коде Хэмминга
7. Пояснить процедуру декодирования циклического кода
8. Почему эффективность кодирования Рида-Соломона возрастает с увеличением длины блока?
9. Назовите сходства и различия сверточного и блочного кода
10. Поясните процедуру декодирования сверточного кода
11. Какова необходимая память устройства при блочном и сверточном чередовании?
12. Какова основная задача скремблирования?

## **3 Принципы компьютерного моделирования в среде MATLAB**

### **3.1 История возникновения системы MATLAB**

В наши дни компьютерная математика получила должную известность и интенсивно развивается как передовое научное направление на стыке математики и информатики. Программируемые микрокалькуляторы и персональные компьютеры уже давно применяются для математических расчетов. Для подготовки программ использовались различные универсальные языки программирования. В начале 90-х гг. на смену им пришли специализированные системы компьютерной математики (СКМ).

Среди них наибольшую известность получили системы Eureka, Mercury, Mathcad, Derive, Mathematica, Maple V и Maple и др. Каждая из этих систем имеет свои достоинства и недостатки и заслуживает отдельного рассмотрения. Повышенный интерес пользователей к подобным системам подтверждают результаты выпуска в последние годы целого ряда книг на русском языке, посвященных указанной теме. За рубежом по каждой серьезной СКМ на web-сайтах их разработчиков можно найти перечни, включающие сотни наименований книг.

Система MATLAB предлагается разработчиками (фирма Math Works, Inc.) как лидирующий на рынке, в первую очередь в системе военно-промышленного комплекса, в аэрокосмической отрасли и автомобилестроении, язык программирования высокого уровня для технических вычислений с большим числом стандартных пакетов прикладных программ [10]. Система MATLAB вобрала в себя не только передовой опыт развития и компьютерной реализации численных методов, накопленный за последние три десятилетия, но и весь опыт становления математики за всю историю человечества. Около миллиона легально зарегистрированных пользователей уже применяют эту систему. Ее охотно используют в своих научных проектах ведущие университеты и научные центры мира. Популярности системы способствует ее мощное расширение Simulink, предоставляющее удобные и простые средства, в том числе визуальное объектно-ориентированное программирование, для моделирования линейных и нелинейных динамических систем, а также множество других пакетов расширения системы [5, 6, 9].

MATLAB – одна из старейших, тщательно проработанных и проверенных временем систем автоматизации математических расчетов, построенная на расширенном представлении и применении матричных операций. Это нашло отражение в названии системы – MATrix LABoratory – матричная лаборатория. Однако синтаксис языка программирования системы продуман настолько тщательно, что эта ориентация почти не ощущается теми пользователями, которых не интересуют непосредственно матричные вычисления.

Матрицы широко применяются в сложных математических расчетах, например при решении задач линейной алгебры и математического моделирования статических и динамических систем и объектов. Они являются основой автоматического составления и решения уравнений состояния динамических объектов и систем. Это существенно повышает интерес к системе MATLAB, вобравшей в себя лучшие достижения в области быстрого решения матричных задач. Однако в настоящее время MATLAB далеко вышла за пределы специализированной матричной системы и стала одной из наиболее мощных универсальных интегрированных СКМ. Слово «интегрированная» указывает на то, что в этой системе объединены удобная оболочка, редактор выражений и текстовых комментариев, вычислитель и графический программный процессор. В новой версии используются такие мощные типы данных, как многомерные массивы, массивы ячеек, массивы структур, массивы Java и разреженные матрицы, что открывает возможности применения системы при создании и отладке новых алгоритмов матричных и основанных на них параллельных вычислений и крупных баз данных.

В целом MATLAB – это уникальная коллекция реализаций современных численных методов компьютерной математики, созданных за последние три десятка лет. Она вобрала в себя и опыт, правила и методы математических вычислений, накопленные за тысячи лет развития математики. Это сочетается с мощными средствами графической визуализации и даже анимационной графики. Систему с прилагаемой к ней обширной документацией вполне можно рассматривать как фундаментальный многотомный электронный справочник по математическому обеспечению ЭВМ – от массовых персональных компьютеров до супер-ЭВМ.

MATLAB пока представлен полностью лишь на английском и частично на японском языках.

Возможности MATLAB весьма обширны, а по скорости выполнения задач система нередко превосходит своих конкурентов. Она применима для расчетов практически в любой области науки и техники. Например, очень широко используется при математическом моделировании механических устройств и систем, в частности в динамике, гидродинамике, аэродинамике, акустике, энергетике и т. д. Этому способствует не только расширенный набор матричных и иных операций и функций, но и наличие пакета расширения SIMULINK, специально предназначенного для решения задач блочного моделирования динамических систем и устройств, а также десятков других пакетов расширений.

В обширном и постоянно пополняемом комплексе команд, функций и прикладных программ (пакетов расширения, пакетов инструментов, - toolbox) [Пакет инструментов, пакет расширения, прикладная программа – почти синонимы при переводе термина toolbox, но пакет инструментов собственно MATLAB рассматривается как один из toolbox всей системы,

включающей MATLAB, SIMULINK и другие пакеты. В дальнейшем и под прикладной программой, и под пакетом расширения имеется в виду toolbox в терминах MATLAB. системы MATLAB содержатся специальные средства для электротехнических и радиотехнических расчетов (операции с комплексными числами, матрицами, векторами и полиномами, обработка данных, анализ сигналов и цифровая фильтрация), обработки изображений, реализации нейронных сетей, а также средства, относящиеся к другим новым направлениям науки и техники. Они иллюстрируются множеством практически полезных примеров. К разработкам расширений для системы MATLAB привлечены многие научные школы мира и руководящие ими крупные ученые и педагоги университетов.

Важными достоинствами системы являются ее открытость и расширяемость. Большинство команд и функций системы реализованы в виде текстовых m-файлов (с расширением m) и файлов на языке Си, причем все файлы доступны для модификации. Пользователю дана возможность создавать не только отдельные файлы, но и библиотеки файлов для реализации специфических задач.

Поразительная легкость модификации системы и возможность ее адаптации к решению специфических задач науки и техники привели к созданию десятков пакетов прикладных программ (toolbox), намного расширивших сферы применения системы. Некоторые из них, например Notebook (интеграция с текстовым процессором Word и подготовка «живых» электронных книг), Symbolic Math и Extended Symbolic Math (символьные вычисления с применением ядра системы Maple V R5) и SIMULINK (моделирование динамических систем и устройств, заданных в виде системы блоков), настолько органично интегрировались с системой MATLAB, что стали ее составными частями.

В данной работе использовалась прикладная программа-расширение SIMULINK, которая в силу представляемого ей особого интереса, будет рассмотрена несколько подробнее [5, 6, 8, 11].

### **3.2 SIMULINK – общие сведения**

Программа SIMULINK является приложением к пакету MATLAB. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от классических способов моделирования, пользователю не нужно досконально изучать язык программирования и численные методы математики, а достаточно общих знаний требующихся при работе на компьютере и, естественно, знаний той предметной области в которой он работает.

SIMULINK является достаточно самостоятельным инструментом MATLAB и при работе с ним совсем не требуется знать сам MATLAB и остальные его приложения. С другой стороны доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в Simulink. Часть входящих в состав пакетов имеют инструменты, встраиваемые в Simulink (например, LTI-Viewer приложения Control System Toolbox – пакета для разработки систем управления). Имеются также дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset – набор блоков для разработки цифровых устройств и т.д.).

При работе с SIMULINK пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков.

При моделировании пользователь может выбирать метод решения дифференциальных уравнений, а также способ изменения модельного времени (с фиксированным или переменным шагом). В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки SIMULINK. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Преимущество Simulink заключается также в том, что он позволяет пополнять библиотеки блоков с помощью подпрограмм написанных как на языке MATLAB, так и на языках C++, Fortran и Ada.

Таким образом, SIMULINK представляет собой мощную, универсальную и, что немаловажно, наглядную простую среду для моделирования всевозможных систем с достаточно простым пользовательским интерфейсом и широкими возможностями [6].

### **3.3 Описание исследуемой модели**

В данном исследовании использовалась реализованная в Simulink модель системы цифровой радиосвязи, которая приведена на рисунке 3.1. Ниже приведена последовательность этапов, которые проходит сигнал в процессе симуляции [5, 6, 7, 11, 13].

С помощью генератора вырабатывается псевдослучайная последовательность битов, моделирующих цифровой сигнал. Этот сигнал поступает в кодер канала. Кодированный сигнал поступает на вход многопозиционного модулятора. С выхода модулятора сигнал поступает в фильтр, а затем в радиоканал, где имитируются шумовые характеристики реального канала связи – белый гауссовский шум, замирания Релея-Райса. Далее выполняется обратная последовательность преобразований.

С целью обеспечения корректной работы модели выбираются соответствующие параметры блоков. Наблюдения проводятся по осциллограммам, спектрограммам и вектограммам в различных точках модели. На приемном конце проводится подсчет ошибок и вычисление вероятности их появления.

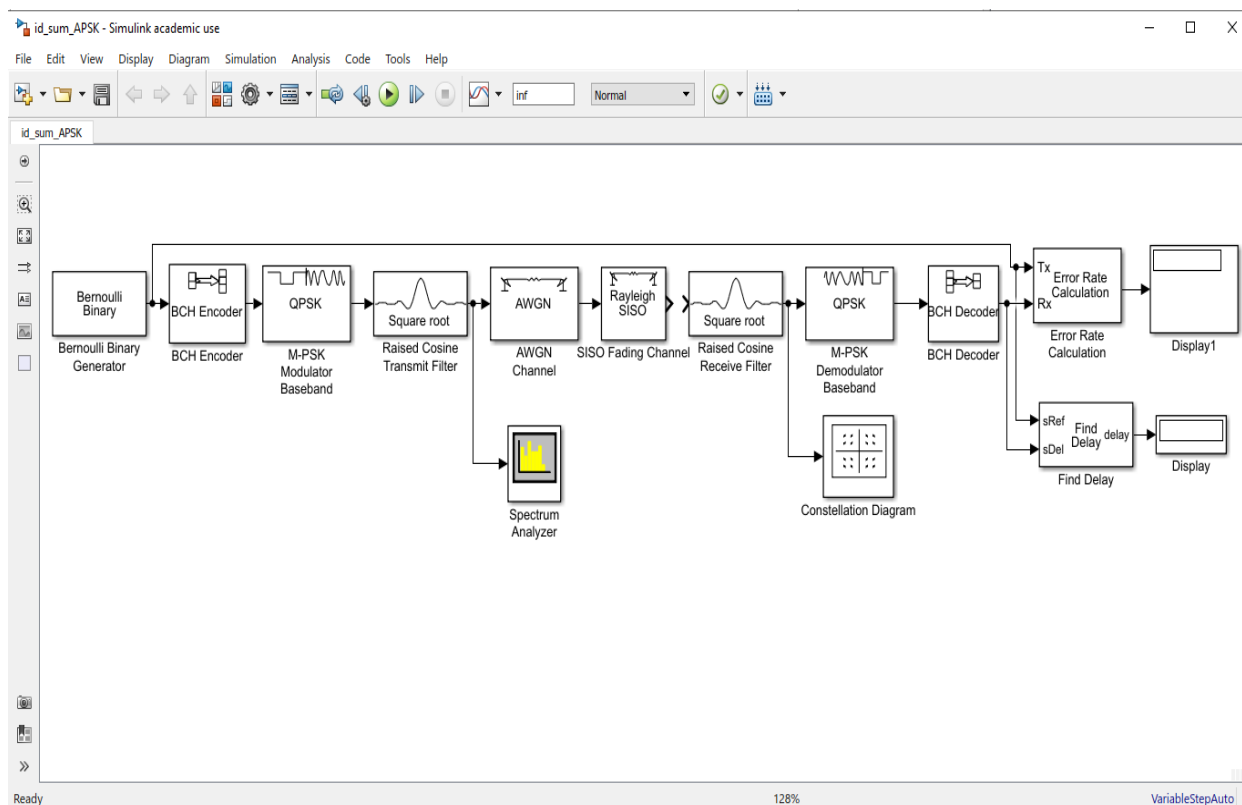


Рисунок 3.1 – Модель цифровой системы радиосвязи, используемая в исследовании

Модель состоит из нескольких основных частей:

- Генератор псевдослучайной последовательности импульсов (Bernoulli Binary Generator);
- Кодер/декодер канала (Channel coder/decoder);
- Модулятор/демодулятор (Modulator/demodulator);
- Цифровые фильтры (Raised Cosine Transmit/Receive Filter);
- Имитаторы каналов и характеристик приемо-передающего тракта (Memoryless nonlinearity, AWGN Channel, Fading Channel);
- Устройства визуализации (Spectrum Scope, Scatter Plot Scope, Scope);
- Счетчик ошибок;

В соответствии с этим, все блоки, используемые в модели можно разбить на несколько категорий:

- Генераторы;

- Канальное кодирование;
- Модуляторы и демодуляторы;
- Имитаторы каналов;
- Цифровые фильтры;
- Устройства графического отображения;
- Устройства регистрации и подсчета ошибок.

### 3.4 Особенности моделирования в SIMULINK

В силу своей специфики, компьютерное моделирование систем связи имеет ряд характерных особенностей, о которых необходимо упомянуть, прежде чем переходить к более подробному рассмотрению компонентов модели.

В первую очередь следует рассмотреть метод моделирования для ВЧ сигналов. Любой непрерывный сигнал за какой-либо отрезок времени принимает бесконечное число значений. Поскольку описать такой сигнал как массив значений амплитуды не представляется возможным (получится бесконечный массив), в MATLAB любой сигнал представляется как последовательность отсчетов, взятых с периодом, величина которого в настройках обозначается как «время отсчета» (Sample Time). Очевидно, что чем меньше этот период, тем точнее отображен сигнал. Для иллюстрации служит рисунок 3.2, на котором приведен синусоидальный сигнал частотой 1 Гц, дискретизированный с периодом дискретизации 0,01 секунды, то есть за 1 секунду ЭВМ обрабатывает 100 значений, каждое из которых является амплитудой синусоиды в определенный момент времени. Безусловно, для современных компьютеров эта задача не представляет никакой сложности. Но исследование сложных систем, с многократными преобразованиями сигналов СВЧ оказывается достаточно ресурсоемким процессом, и время симуляции, даже на мощных компьютерах, иногда значительно превышает время реального процесса.

Для снижения нагрузки на центральный процессор и ускорения процесса симуляции в системе MATLAB для моделирования цифровых сигналов используется так называемый эквивалентный метод низких частот (lowpass equivalent method) или узкополосная модуляция (baseband modulation). Суть метода заключается в том, что вместо множества отсчетов амплитуды за время равное длине модуляционного символа передается комплексное число, модуль которого равен амплитуде, фаза – фазе модулированного сигнала. Допустим, модулированный сигнал описывается выражением

$$S(t) = Y_1(t) \cos(2\pi f_c t + \theta) - Y_2(t) \sin(2\pi f_c t + \theta) \quad (3.1)$$



где:  $Y_1$  и  $Y_2$  – амплитудные составляющие;  
 $f_c$  – несущая частота;  
 $t$  – время;  
 $\theta$  – начальная фаза.

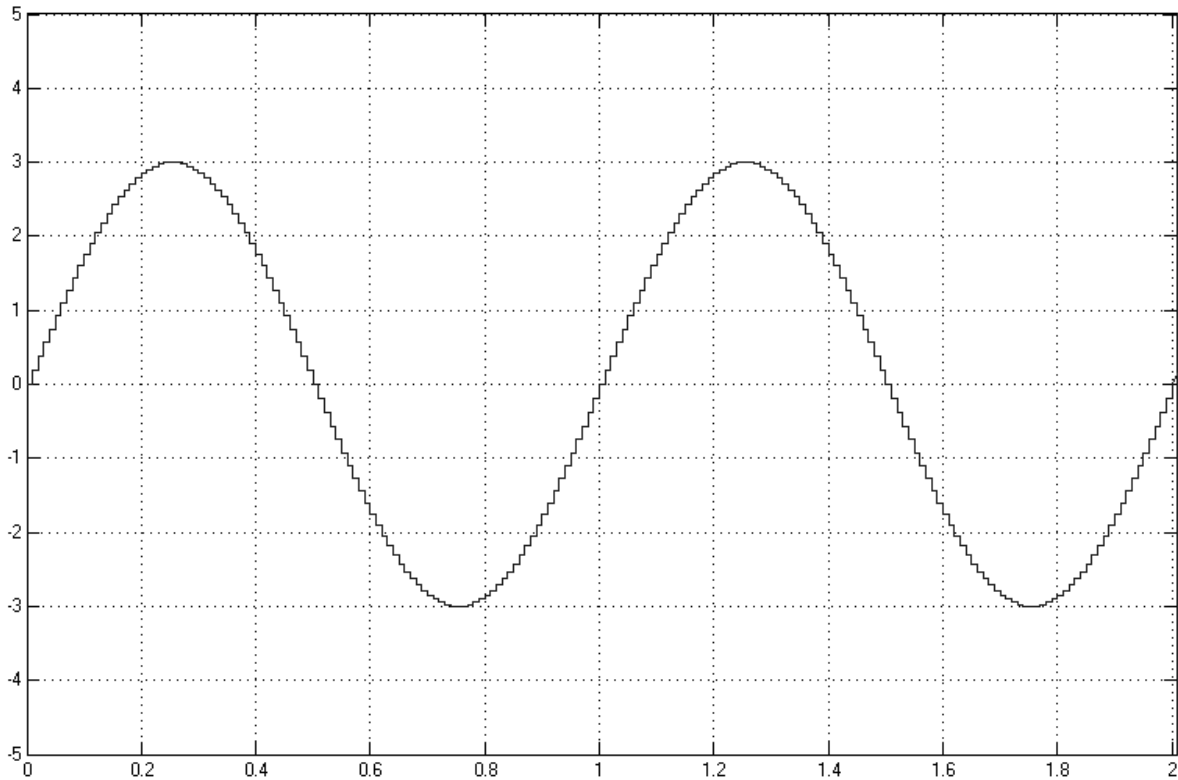


Рисунок 3.2 – Синусоидальный сигнал, дискретизированный с частотой 100 Гц

При узкополосной модуляции учитывается, что это выражение эквивалентно действительной части выражения

$$[(Y_1(t) + jY_2(t))e^{j\theta}]e^{j2\pi f_c t}, \quad (3.1.1)$$

и для моделирования используется только выражение внутри квадратных скобок

$$S(t) = (Y_1(t) + jY_2(t))e^{j\theta} \quad (3.2)$$

Вероятность ошибки при использовании моделей с полосовой (Passband) и узкополосной (Baseband) модуляцией отличается менее чем на 0,1%, однако при этом модели с использованием полосовой модуляции требуются несравнимо большее время, для обработки такого же объема

данных. Разница в скорости симуляции особенно заметна, при больших значениях несущей частоты.

Еще одним способом применяемым в MATLAB с целью уменьшить время симуляции процесса является использование многоканальных сигналов или кадров (Frames). Кадр – это последовательность отсчетов, выстроенная в единый вектор (матрицу столбец). Использование моделирования на основе кадров (Frame-Based processing) можно сравнить с передачей двоичных сигналов в последовательном и параллельном кодах. При передаче сигналов в «последовательном коде» каждый отчет передается строго за предыдущим, при использовании «параллельного кода» по нескольким каналам одновременно передаются  $n$  символов, вместе составляющих кодовое слово. Время затраченное на передачу одного и того же количества значений во втором случае уменьшается кратно размеру используемого кадра (если говорить о параллельном коде, то это число параллельных каналов). Еще одним плюсом использования кадров является то, что при прохождении через различные блок все значения кадра обрабатываются одновременно.

Итак, использование узкополосной модуляции на основе кадров очень значительно увеличивает скорость обработки информации, и по сравнению с полосовой модуляцией, позволяет достичь того же результата за гораздо меньшее время. Кроме того, многие блоки библиотеки Communications Blockset требуют в качестве входного сигнала исключительно Baseband сигнал с определенным размером кадра (например, блок Rician Fading Channel – Канал с Райсовскими замираниями), что делает использование узкополосной модуляции необходимым условием для моделирования систем коммуникации.

Теперь мы можем рассмотреть каждую категорию используемых в модели блоков в отдельности и более подробно.

### **3.5 Генератор псевдослучайной двоичной последовательности**

Для генерации псевдослучайной последовательности двоичных импульсов, имитирующей реальный цифровой сигнал, был использован генератор двоичной последовательности Бернулли Bernoulli Binary Generator (рисунок 3.3) из раздела Comm Sources библиотеки Communications Toolbox.

Блок генерирует псевдослучайные двоичные символы согласно закону распределения Бернулли. Генератор выдает на выходе ноль с вероятностью  $p$  и единицу с вероятностью  $1-p$ . Параметр «Вероятность нуля» определяет  $p$  и задается как действительное число между нулем и единицей.

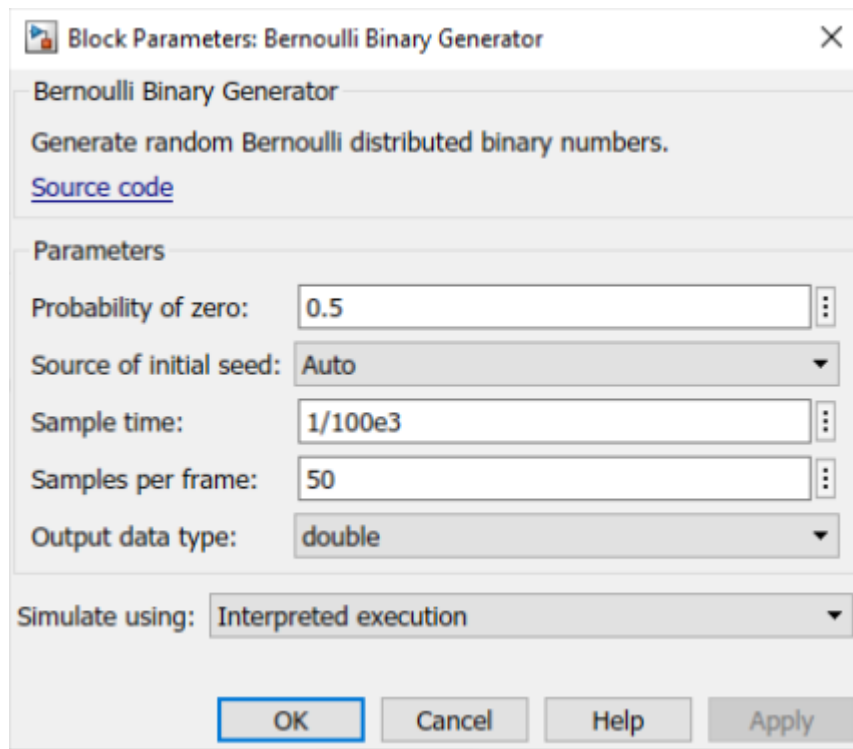


Рисунок 3.3 – Параметры блока Bernoulli Binary Generator

Настраиваемые параметры:

Probability of a zero – вероятность нуля на выходе генератора;

Source of Initial seed – источник псевдослучайной последовательности;

Sample time – обратная величина частоты дискретизации период двоичных символов, векторов или матриц на выходе, численно равный единица делённая на скорость передачи;

Samples per frame – размер фрейма;

### 3.6 Модуляторы

Библиотека Communications Toolbox, раздел Modulation, подраздел Digital Baseband Modulation. В исследовании рассматривались следующие виды модуляций: M-PSK, M-DPSK и M-QAM. Поскольку модулятор и демодулятор выполняют дополняющие друг друга функции, настройки у них соответствующих блоков симметричны, поэтому имеет смысл рассмотреть все блоки этого раздела на примере модуляторов.

#### 3.6.1 Узкополосный модулятор M-PSK (M-PSK Modulator Baseband)

Формирует из M бит поступающих на вход модуляционный символ, используя M-арную фазовую модуляцию M-PSK (рисунки 3.4 и 3.5). В

качестве входного сигнала блок может воспринимать как двоичную последовательность, так и дискретный во времени сигнал с целочисленной амплитудой, лежащей в промежутке между нулем и  $(M-1)$ , и постоянным периодом токовых посылок.

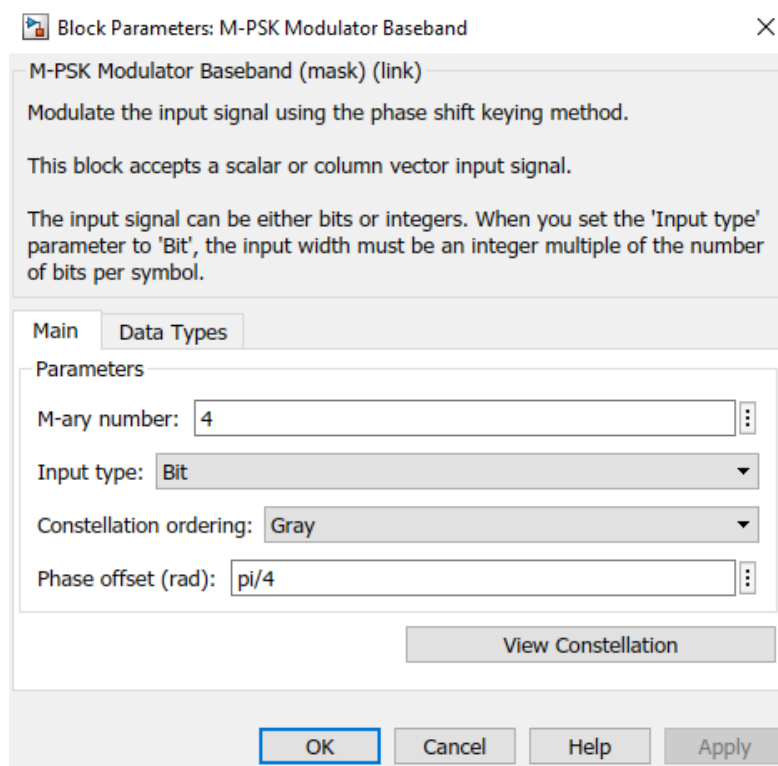


Рисунок 3.4 – Параметры блока M-PSK Modulator Baseband

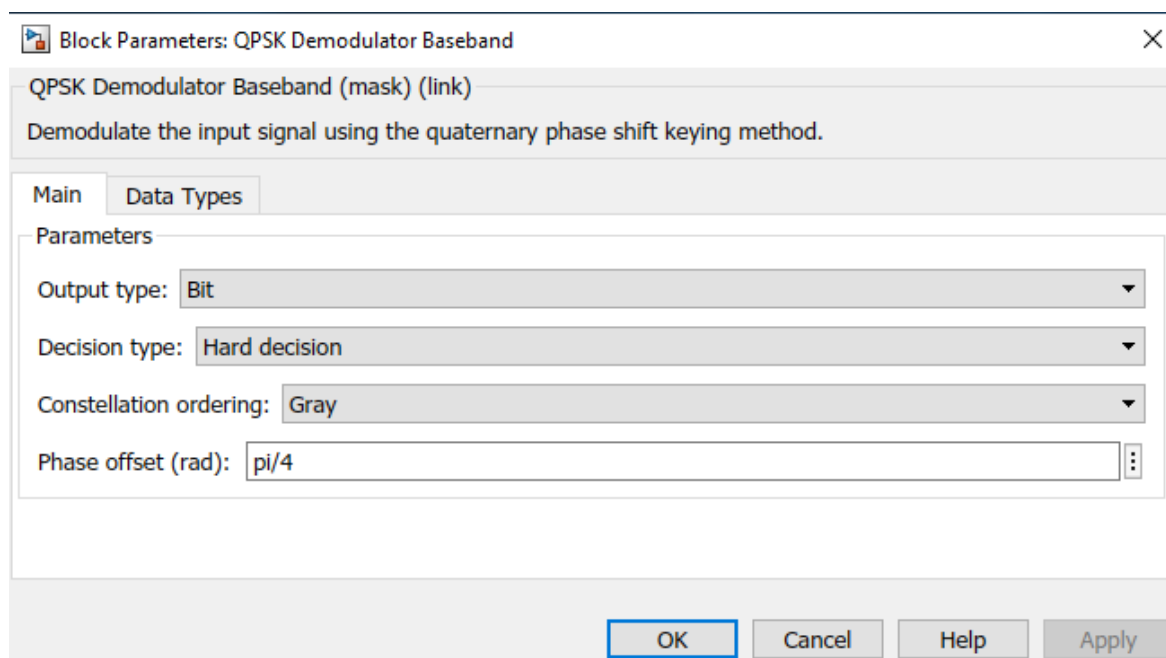


Рисунок 3.5 – Параметры блока M-PSK Demodulator Baseband

Настраиваемые параметры блока:

M-ary number – число точек, в созвездии сигнала (позиционность модуляции);

Input type – показывает тип входного сигнала – целые числа, или группы бит. Если этот параметр установлен в позицию Bit, то параметр M-ary number должен быть равен  $2^k$ , при любом целом положительном k;

Constellation ordering – определяет то, как блок ставит в соответствие группе бит сигнал определенной амплитуды и фазы. Возможно обычное соответствие, либо представление в коде Грея (рисунок 3.6);

Phase offset (rad) – фаза нулевой сигнальной точки созвездия сигнала;

View Constellation – можно посмотреть созвездие (расположение сигнальных точек на фазово-амплитудной плоскости).

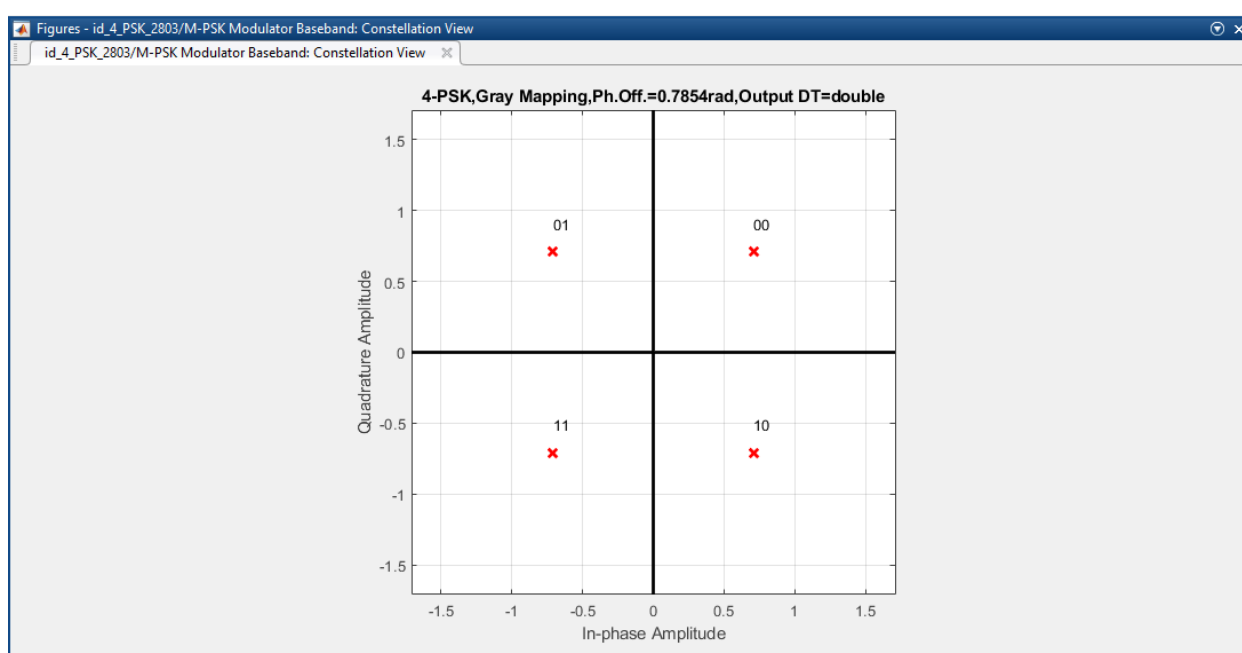


Рисунок 3.6 – Созвездие для модуляции 4-PSK в коде Грея

### 3.6.2 Узкополосный модулятор M-DPSK (M-DPSK Modulator Baseband)

Действует аналогично модулятору M-PSK, и формирует из M бит поступающих на вход модуляционный символ, используя M-арную дифференциальную фазовую модуляцию (M-DPSK). В случае, если тип входного сигнала определен как целые числа, допустимым сигналом на входе являются целые значения, лежащие в промежутке между нулем и (M-1). В этом случае, входной сигнал может быть как скалярной величиной, так и последовательностью двумерных векторов (передаваться в параллельном

коде). Если первое входное значение равно  $k$ , то модуляционный символ записывается как

$$\exp(j\theta + j2\pi \cdot \frac{k_1}{M}) \quad (3.3)$$

где  $\theta$  – фаза нулевой точки сигнала. Если обозначить последующее входное значение как  $k$ , то следующий символ равен

$$\exp(j\theta + j2\pi \cdot \frac{k}{M}) - (\text{предыдущий символ}).$$

Настраиваемые параметры блока:

Аналогично модулятору M-PSK.

### 3.6.3 Узкополосный модулятор M-QAM (Rectangular QAM Modulator Baseband)

Блок модулирует входной сигнал, используя квадратурную амплитудную модуляцию. Созвездие сигнала состоит из  $M$  точек, где  $M$  должно удовлетворять условию

$$M = 2^k \quad (3.4)$$

для некоторого натурального  $k$ .

Вид созвездия модулированного сигнала определяется установленным методом нормализации (Normalization method parameter). Таблица 3.1 иллюстрирует возможные конфигурации.

Таблица 3.1 – Методы нормализации

Метод нормализации	Способ масштабирования
Минимальное расстояние между символами (Min. distance between symbols)	Ближайшая пара точек созвездия разделены величиной «минимальное расстояние между символами»
Средняя мощность, Ватт (Average Power)	Средняя мощность символов созвездия (в Ваттах) определяется параметром «средняя мощность»
Пиковая мощность (Peak Power)	Максимальная мощность символов созвездия (в Ваттах) определяется параметром «пиковая мощность»

Аналогично модуляторам, рассмотренным выше, модулятор QAM работает как с двоичными импульсами, так и с квантованными значениями целых чисел на входе.

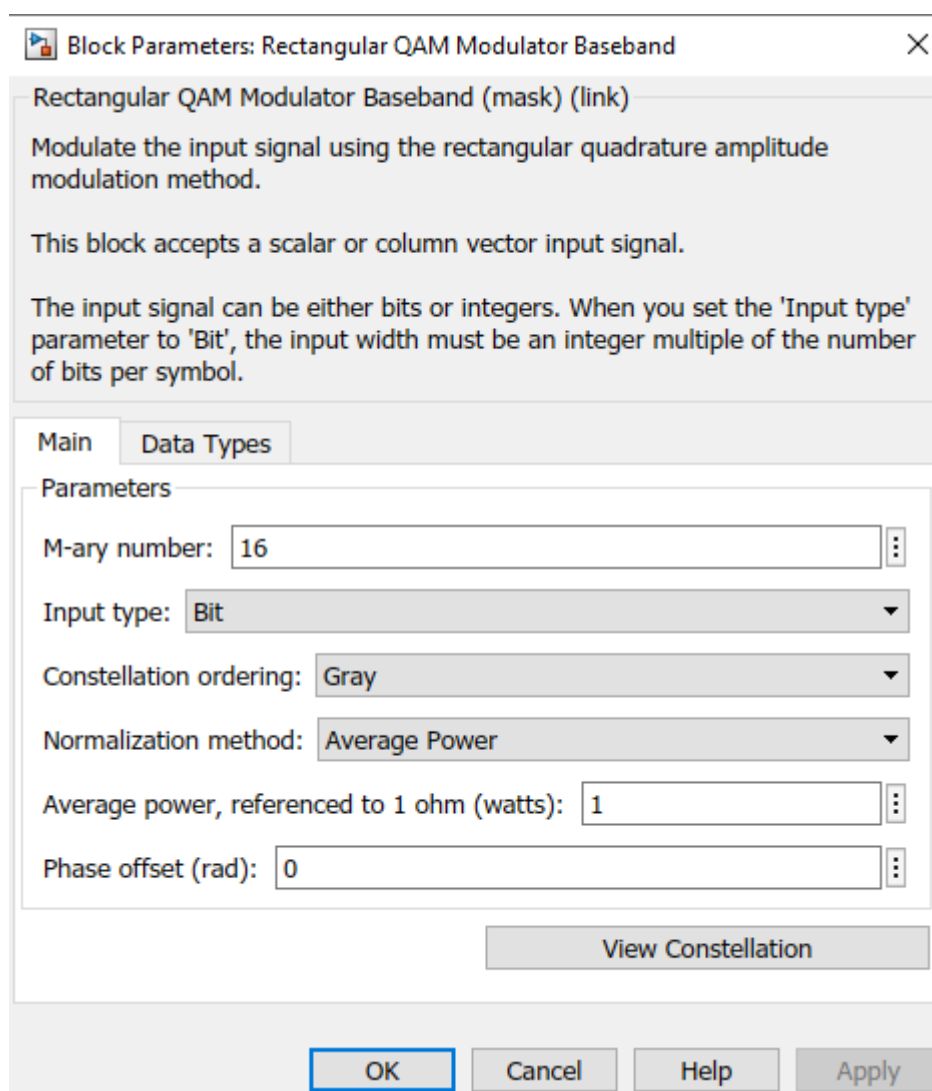


Рисунок 3.7 – Параметры блока QAM Modulator Baseband

Настраиваемые параметры блока:

M-ary number – число точек, в созвездии сигнала. Число должно удовлетворять условию  $M = 2^k$  для некоторого натурального  $k$ ;

Input type – показывает тип входного сигнала – целые числа, или группы бит;

Constellation ordering – определяет то, как блок ставит в соответствие группе бит сигнал определенной амплитуды и фазы. Возможно обычное

соответствие, либо представление в коде Грея. Поле активно только в случае двоичного входного сигнала;

Normalization method – задает способ масштабирования созвездия. Предполагает варианты: Min. distance between symbols, Average Power, и Peak Power;

Phase offset (rad) – фаза нулевой точки созвездия сигнала;

Samples per symbol – дискретизация выходного сигнала, ставящегося в соответствие каждому входному двоичному кодовому слову.

На рисунке 3.8 приведена осциллограмма, иллюстрирующая изменения, происходящие с сигналом QAM-16 после его прохождения через фильтр. Для наглядности на сигнал непосредственно перед осциллографом наложена несущая частота (в канал передается лишь узкополосное представление сигнала).

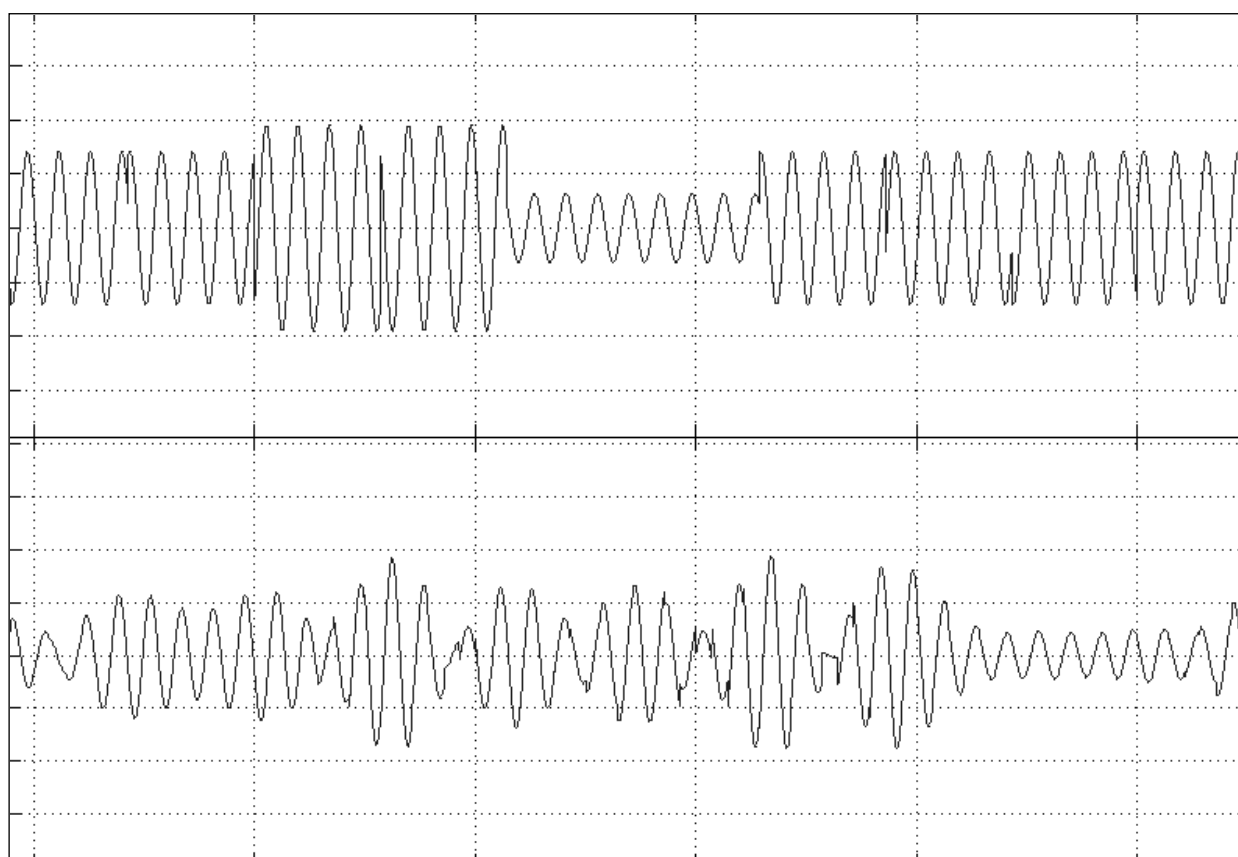


Рисунок 3.8 – Пример использования фильтра Raised Cosine Transmit Filter, для сигнала модулированного 16-QAM

### 3.7 Цифровые фильтры

Блоки Фильтр типа «Приподнятый косинус» для предающего и приемного концов (Raised Cosine Transmit Filter и Raised Cosine Receive Filter



соответственно) предназначены для ограничения спектра модулированного сигнала. На выходе модулятора фильтр установлен для уменьшения передаваемой в канал мощности и ограничения бесконечного спектра исходного сигнала, а на входе модулятора – затем, чтобы уменьшить мощность принимаемых шумов. Строго говоря, любой фильтр типа «приподнятый косинус» является НЧ фильтром, а для заявленных выше операций необходим полосовой фильтр. Выбор упомянутых блоков объясняется следующим образом. Блоки Raised Cosine Transmit Filter и Raised Cosine Receive Filter являются узкополосными моделями соответствующих фильтров. Как отмечалось выше, при узкополосном представлении модулированного сигнала несущая частота не используется, и спектр сигнала можно сравнить со спектром последовательности прямоугольных импульсов, а для ограничения такой полосы частот нужен именно НЧ фильтр.

Импульсная характеристика фильтра у которого Rolloff Factor равен  $\alpha$ , для периода входного символа  $T$  имеет вид:

$$h(t) = \frac{\sin(\pi t / T)}{(\pi t / T)} \cdot \frac{\cos(\pi \alpha t / T)}{(1 - 4\alpha^2 t^2 / T^2)} \quad (3.5)$$

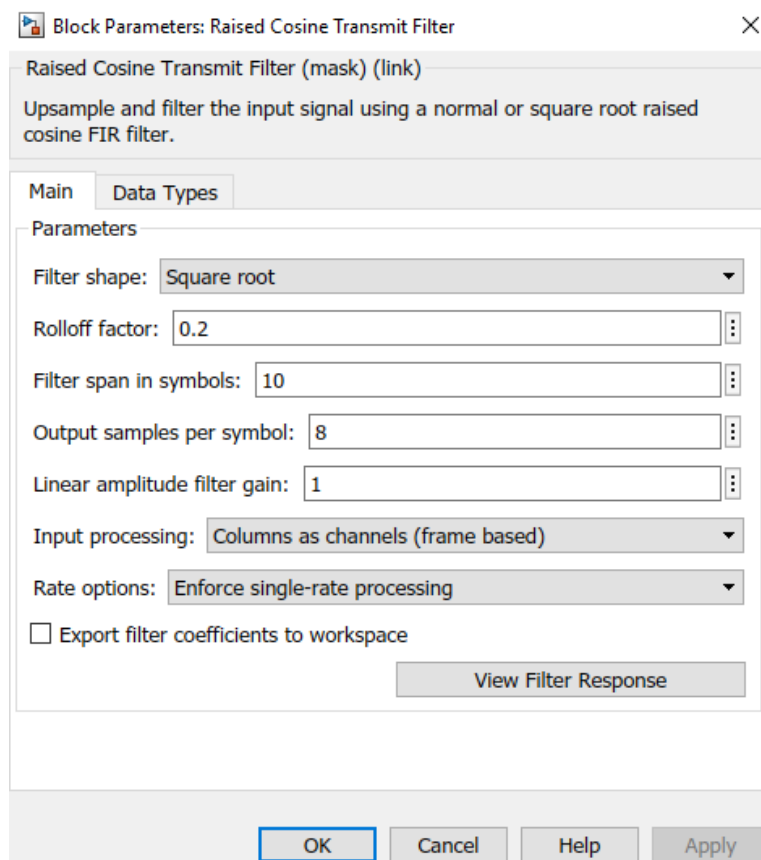


Рисунок 3.9 – Параметры блока Raised Cosine Transmit Filter

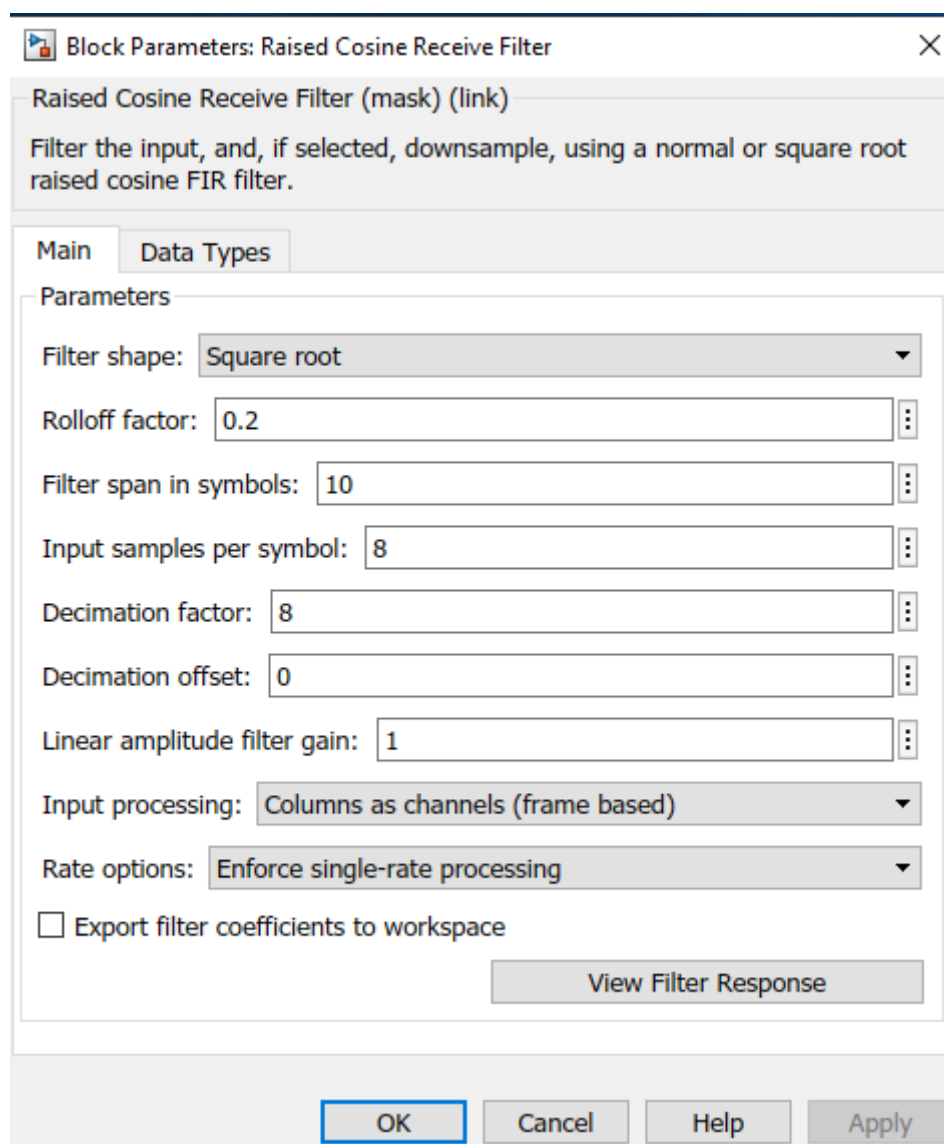


Рисунок 3.10 – Параметры блока Raised Cosine Receive Filter

Настраиваемые параметры блока:

Filter type – тип фильтра: квадратный корень или нормальный;

Rolloff factor (ROF) – коэффициент сглаживания характеристики фильтра, задается в пределах от нуля до единицы;

Filter span in symbols – длина фильтра;

Output samples per symbol – коэффициент интерполяции;

Linear amplitude filter gain – усиление фильтра;

View Filter Response – посмотреть отклик фильтра (рисунок 3.11)

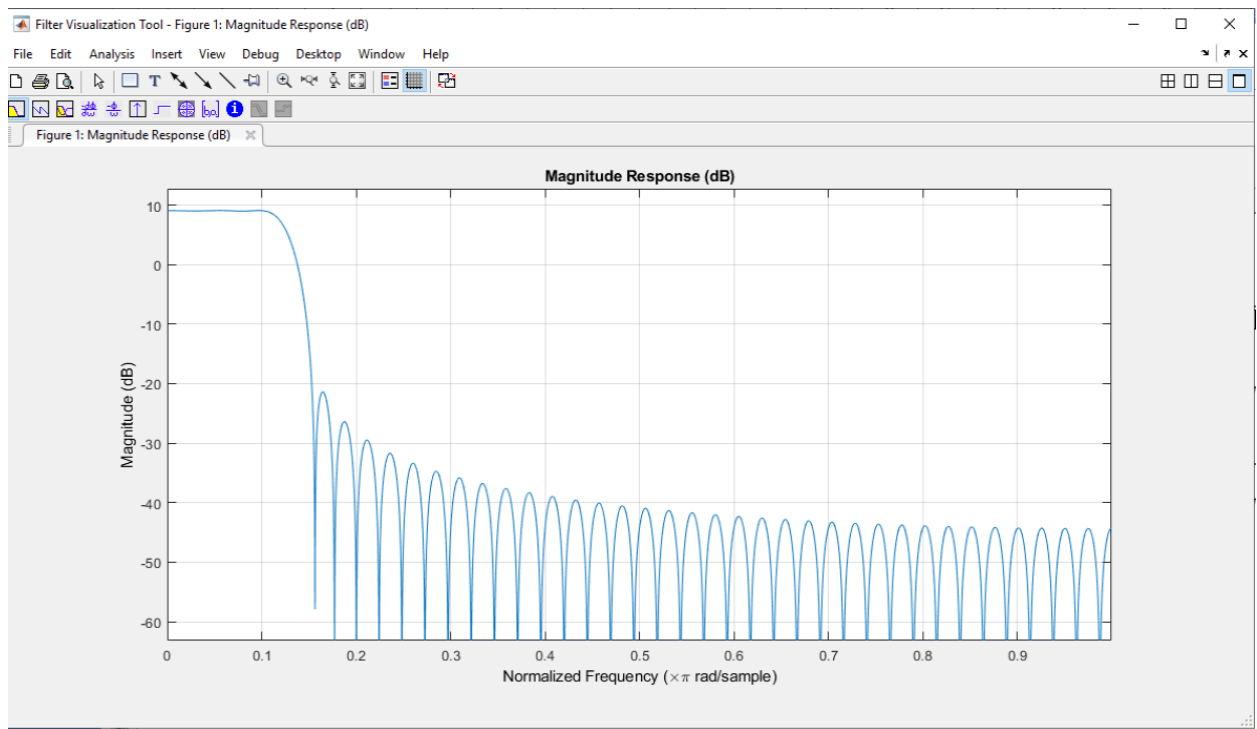


Рисунок 3.11 – Импульсная характеристика Raised Cosine Transmit Filter

### 3.8 Имитаторы каналов

#### 3.8.1 Канал с аддитивным белым Гауссовским шумом (AWGN Channel)

Блок AWGN (рисунок 3.12) смешивает входной сигнал с белым Гауссовским шумом. Его плотность распределения может задаваться прямо или косвенно с помощью следующих режимов:

Signal to noise ratio ( $E_b/N_0$ ) – отношение битовой энергии сигнала к спектральной плотности мощности шума в децибелах;

Signal to noise ratio ( $E_s/N_0$ ) – отношение символьной энергии сигнала к спектральной плотности мощности шума в децибелах;

Signal to noise ratio (SNR) – отношение уровней мощностей входного сигнала и шума в децибелах;

Variance from mask – в диалоговом окне задается, непосредственно, плотность вероятности шума;

Variance from port – функция, описывающая дисперсию, подается на отдельный вход блока.

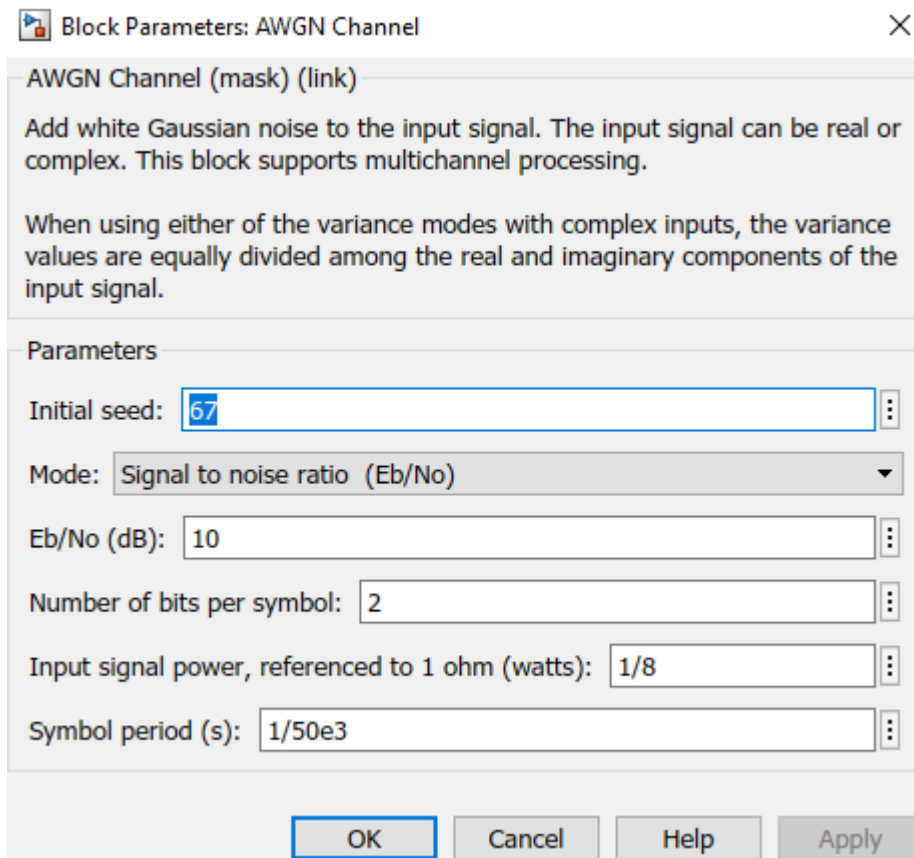


Рисунок 3.12 – Параметры блока AWGN Channel

Настраиваемые параметры блока:

Initial seed – номер псевдослучайной последовательности;

Mode – режим, определяющий дисперсию шума ( $E_b/N_0$ ,  $E_s/N_0$  и т.д.);

Number of bits per symbol – количество бит, приходящееся на каждый входной символ,  $k = \log_2 M$ . Поле появляется только в режиме  $E_b/N_0$  ;

Input signal power (watts) – среднеквадратичная мощность входных символов равна  $1/8$ , так как коэффициент интерполяции фильтра равен 8. Поле появляется в режимах  $E_b/N_0$ ,  $E_s/N_0$  или SNR;

Symbol period (s) – период символов входного сигнала в секундах численно равный единица делённая на скорость передачи символов. Поле появляется только в режимах  $E_b/N_0$  или  $E_s/N_0$  ;

Variance – Дисперсия белого Гауссовского шума. Поле активно только в режиме Variance from mask.

### 3.8.2 Канал с замираниями Райса (Rician Fading Channel)

Блок реализует среду распространения с замираниями Райса для узкополосным систем, то есть, применим только для моделей использующих узкополосные (baseband) модуляторы. Этот блок полезен для моделирования

мобильных беспроводных систем, в которых передаваемый сигнал может достигнуть приемника как в отраженном виде, так и по пути прямой видимости.

Настраиваемые параметры блока:

K-factor – отношение мощности сигнала, проходящего по прямому пути к мощности рассеянного (отраженного) сигнала. Отношение задается линейно (не в децибелах);

Maximum Doppler shift (Hz) – положительная величина, задающая максимальный Доплеровский сдвиг частоты, который характеризует относительное движение между передатчиком и приемником;

Sample time – период элементов во входном сигнале;

Delay (s) – величина, задающая задержку распространения в секундах;

Gain (dB) – величина, задающая уровень сигнала в децибелах;

Initial seed – псевдослучайная последовательность для Гауссовского генератора шума.

### **3.8.3 Канал с замираниями Релея (Multipath Rayleigh Fading Channel)**

Блок осуществляет узкополосное моделирование среды распространения с замираниями Релея. Так же как и канал с замираниями Райса, блок применяется для моделирования мобильных беспроводных систем. В многолучевом канале распространения сигнал проходит от передатчика до приемника несколькими разными путями, возможно имеющими разную протяженность и вносящими разную временную задержку.

Настраиваемые параметры блока:

Maximum Doppler shift (Hz) – положительная величина, задающая максимальный Доплеровский сдвиг частоты, который характеризует относительное движение между передатчиком и приемником;

Sample time – период элементов во входном сигнале;

Delay vector (s) – вектор, определяющий величину задержки распространения для каждого пути в секундах;

Gain vector (dB) – вектор, определяющий уровень сигнала в децибелах для каждого пути распространения;

Normalize gain vector to 0 dB overall gain – распределяет уровень сигнала в канале таким образом, что уровень для всех путей распространения составляет ноль децибел;

Initial seed – псевдослучайная последовательность для Гауссовского генератора шума.

### 3.9 Канальное кодирование

Кодер канала осуществляет преобразование сигнала по трем направлениям:

- Обнаружение и исправление ошибок – Error Detection and Correction;
- Перемежение – Interleaving;
- Скремблирование – Scrambling.

Каждое из этих направлений представлено в программе Simulink соответствующими библиотеками.

#### 3.9.1 Обнаружение и исправление ошибок (Error Detection and Correction)

Библиотека «Error Detection and Correction» состоит из трех подбиблиотек:

- Блочное кодирование – Block;
- Сверточное кодирование – Convolutional;
- Вычисление циклических сумм – CRC.

Блочное кодирование – Block Coding

Пакет поддерживает работу с произвольными линейными блочными кодами. Имеется также дополнительная поддержка нескольких более узких классов кодов – кодов Хэмминга, кодов Боуза – Чоудхури – Хоквингема (БЧХ) и кодов Рида – Соломона.

*Кодер Хэмминга – Hamming Coder*

Коды Хэмминга являются одним из подклассов циклических блочных кодов. Порождающий полином для кодов Хэмминга неприводим и примитивен, а длина кодированного блока равна  $2^m - 1$ . Порождающая и проверочная матрицы для кодов Хэмминга генерируются функцией `hamngen`.

Блок производит код Хэмминга из  $K$  информационных битов с длиной кодового слова  $N$ . Длина кодового слова  $N$  должна иметь форму  $2^M - 1$ . Где  $M$  – целое положительное число большее или равное 3. Тогда  $K = N - M$ .

Входной сигнал должен содержать ровно  $K$  элементов. Выходная последовательность – вектор длиной  $N$ .

Настраиваемые параметры блока:

Codeword length  $N$  – Длина кодового слова;

Message length  $K$ , or  $M$ -degree polynomial – Длина информационного слова, или вектор двоичных чисел, представляющих собой простейший полином для полей Галуа.

*Линейный кодер – Binary Linear Coder*

Линейный блочный код в общем случае описывается порождающей матрицей (generator matrix). Кодирование блока (вектора) производится путем его умножения на порождающую матрицу. Помимо порождающей, существует проверочная матрица кода (parity-check matrix). Она может использоваться для обнаружения ошибок – при отсутствии ошибок умножение кодированного блока на проверочную матрицу должно давать нулевой вектор. Преобразование порождающей матрицы в проверочную и обратно осуществляется функцией `gen2par`.

Если умножение кодированного блока на проверочную матрицу не дает нулевого вектора, то полученный результат (его называют синдромом – syndrome) позволяет определить, какие именно символы были искажены в процессе передачи. Если код является двоичным (то есть символы могут принимать только значения 0 и 1), это позволяет исправить ошибки. Декодирование линейного блочного кода, таким образом, можно осуществить с помощью таблицы, в которой для каждого значения синдрома указан соответствующий вектор ошибок. Создать такую таблицу на основании проверочной матрицы кода позволяет функция `syndtable`.

Функция `gfweight` позволяет определить кодовое расстояние для линейного блочного кода по его порождающей или проверочной матрице.

Блок Линейного кодера генерирует линейный код, используя заданную генерирующую матрицу.  $K$  – количество информационных битов. Генерирующая матрица должна иметь  $K$  строк. Если  $N$  – длина кодового слова, то генерирующая матрица должна иметь  $N$  столбцов.

Входной сигнал должен содержать ровно  $K$  элементов. Выходная последовательность – вектор длиной  $N$ .

Настраиваемые параметры блока:

Generator matrix –  $K \times N$  матрица, где  $K$  – количество информационных битов,  $N$  – длина кодового слова.

### *Циклический кодер – Binary Cyclic Coder*

Циклические коды – это подкласс линейных кодов, обладающие тем свойством, что циклическая перестановка символов в кодированном блоке дает другой возможный кодированный блок того же кода.

Для работы с циклическими кодами в пакете Communications имеются две функции. Задав число символов в кодируемом и закодированном блоках, с помощью функции `cyclpoly` можно получить порождающий полином циклического кода. Далее, используя этот полином в качестве одного из параметров функции `cyclgen`, можно получить порождающую и проверочную матрицы для данного кода.

Блок циклического кодера производит систематический циклический код из  $K$  информационных битов с длиной кодового слова  $N$ . Длина кодового слова  $N$  должна иметь форму  $2^M - 1$ . Где  $M$  – целое положительное число большее или равное 3.

Входной сигнал должен содержать ровно  $K$  элементов. Выходная последовательность – вектор длиной  $N$ .

Настраиваемые параметры блока:

Codeword length  $N$  – Длина кодового слова;

Message length  $K$ , or  $M$ -degree polynomial – Длина информационного слова, или вектор двоичных чисел, представляющих собой простейший образующий полином.

#### *Кодер БЧХ – BCH Coder*

Коды БЧХ являются одним из подклассов циклических блочных кодов. Для работы с ними функции высокого уровня вызывают специализированные функции `bchenco` (кодирование) и `bchdeco` (декодирование). Кроме того, функция `bchpoly` позволяет рассчитывать параметры или порождающий полином для двоичных кодов БЧХ.

Блок БЧХ кодера производит код ВСН из  $K$  информационных битов с длиной кодового слова  $N$ .  $b$  и  $K$  задаются в качестве параметров согласно таблицам  $[N-K-t]$ .

Входной сигнал должен содержать ровно  $K$  элементов. Выходная последовательность – вектор длиной  $N$ .

Настраиваемые параметры блока:

$N$  – Длина кодового слова;  $b$  Длина информационного слова.

#### *Кодер Рида-Соломона – Binary RS Coder*

Коды Рида – Соломона являются одним из подклассов циклических блочных кодов. Это единственные поддерживаемые пакетом Communications коды, которые работают не с однобитовыми, а с многобитовыми символами.

Для работы с кодами Рида – Соломона функции высокого уровня вызывают специализированные функции `rsenco` (кодирование) и `rsdeco` (декодирование). Кроме того, функции `rsencode` и `rsdecode` позволяют использовать при кодировании и декодировании экспоненциальный формат данных, а функции `rsencof` и `rsdecof` осуществляют кодирование и декодирование текстового файла. Наконец, функция `rspoly` генерирует порождающие полиномы для кодов Рида-Соломона.

Блок кодера производит код Рида-Соломона из  $K$  информационных битов с длиной кодового слова  $N$ . Символы кода это двоичные последовательности длиной  $M$ , соответствующие элементам полей Галуа  $GF(2^M)$ , где первый бит в каждой последовательности – наиболее значимый бит.

Входной сигнал – двоичный вектор, длиной  $M \times K$  элементов. Выходная последовательность – вектор длиной  $M \times N$ .

$M$  должно быть больше или равным  $\log_2(N+1)$ . Есть возможность изменить  $b$ , задавая специальный образующий полином. Если  $N$  меньше  $2^M - 1$ , то блок использует упрощенный код Рида-Соломона.



(N, K) код может исправить до  $(N-K)/2$  символьных (не битовых) ошибок в каждом кодовом слове.

Настраиваемые параметры блока:

Codeword length N – Длина кодового слова  $M \times N$ ;

Message length K – Длина информационного слова  $M \times N$ ;

Specify primitive polynomial – Выбрав данный параметр, можно задать простейший полином двоичным вектором;

Primitive polynomial – Двоичный вектор, представляющий собой простейший полином со степенями в порядке убывания;

Specify generator polynomial – Выбрав данный параметр, можно задать образующий полином десятичным вектором;

Generator polynomial – Десятичный вектор, представляющий собой простейший полином со степенями в порядке убывания.

Сверточное кодирование – Convolutional Coding

Кодер для сверточного кода представляет собой устройство с памятью. Поступающие на вход кодера символы по определенному закону меняют внутреннее состояние кодера. Выходные символы (символы кодированного сообщения) зависят от входных символов и от внутреннего состояния кодера. Этот процесс реализуется с помощью функции convenc. На рисунке 3.13 представлена схема сверточного кодера со скоростью кода 2/3.

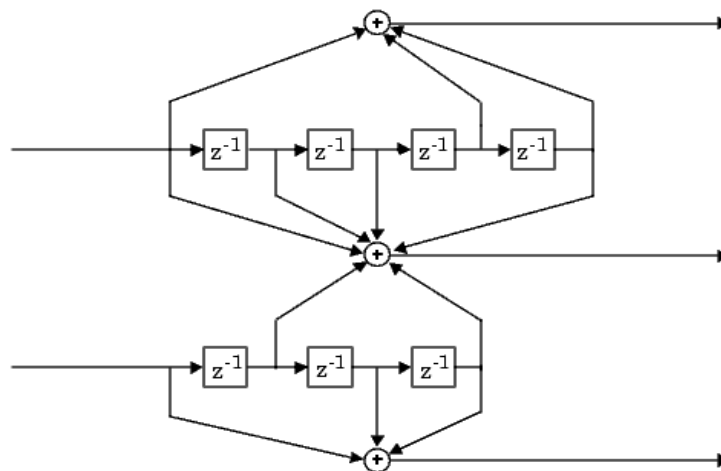


Рисунок 3.13 – Схема сверточного кодера (2/3)

В версии 2.0 пакета Communications по сравнению с предыдущими версиями сильно изменен формат представления сверточных кодов – теперь они описываются наглядной таблицей переходов (функциями poly2trellis и istrellis). Также существенно изменена реализация мягкого декодирования сверточных кодов.

*Сверточный кодер – Convolutional Coder*

Сверточный кодер кодирует последовательность входных двоичных векторов в последовательность выходных двоичных векторов.

При поступлении на вход  $k$  двоичных потоков (возможно  $2^k$  потоков) длина входного вектора равняется  $L \times k$ , при некотором целом положительном  $L$ . Подобно этому кодер производит  $n$  выходных двоичных потоков (возможно  $2^n$  потоков) при длине выходного вектора  $L \times n$ .  $L$  – определяет количество входных бит, обрабатываемых за один такт.

Настраиваемые параметры блока:

Trellis structure – MATLAB-структура, которая содержит Треллис-описание сверточного кодера;

Reset – Определяет условия, при которых состояние кодера будет сброшено на «0»:

- None – никогда;
- On each frame – после каждого кадра;
- On nonzero Rst input – при поступлении сигнала на порт «Rst».

*Декодер Витерби – Viterbi Decoder*

Декодирование сверточного кода производится по алгоритму Витерби функцией vitdec. Поддерживаются мягкое и жесткое декодирование.

Если сверточный код использует алфавит из  $2^n$  возможных символов, то длина входного вектора равна  $L \times n$  для некоторого положительного  $L$ .  $L$  – определяет количество входных бит, обрабатываемых за один такт.

Настраиваемые параметры блока:

Trellis structure – MATLAB-структура, которая содержит Треллис-описание сверточного кодера. Должна использоваться такая же структура, как и в кодере;

Decision type – Тип принимаемого решения:

- Quantized – квантованное решение [+1, -1]
- Hard Decision – «жесткое» решение [0, 1]
- Soft Decision – «мягкое» решение

Number of soft decision bits – Количество бит для принятия «мягкого» решения для каждого входа;

Traceback depth – Число Треллис-ветвей для построения пути принятия решения;

Operation mode – Режим сброса состояния декодера;

Reset input – Специальный порт для сброса состояния декодера [7].

Вычисление циклических сумм – Cyclic Redundancy Check

Подбиблиотека состоит из различных устройств для добавления и проверки циклических сумм.

*Генератор циклических сумм – CRC-N Generator.*

Блок генерирует CRC-код для каждого входного двоичного вектора и добавляет код к нему.

Данный блок предоставляет возможность выбора образующего полинома для CRC алгоритма из списка наиболее широко используемых

полиномов. N – степень образующего полинома. Таблица 3.2 показывает варианты образующих полиномов.

Настраиваемые параметры блока:

CRC-N method – Образующий полином для CRC-алгоритма;

Initial states – Двоичный вектор длиной, равной степени образующего полинома, определяющий начальные состояния внутренних регистров сдвига;

Таблица 3.2 – Образующие полиномы

CRC метод	Образующий полином	Количество бит
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$	32
CRC-24	$x^{24}+x^{23}+x^{14}+x^{12}+x^8+1$	24
CRC-16	$x^{16}+x^{15}+x^2+1$	16
Reversed CRC-16	$x^{16}+x^{14}+x+1$	16
CRC-8	$x^8+x^7+x^6+x^4+x^2+1$	8
CRC-4	$x^4+x^3+x^2+x+1$	4

Checksums per frame – Число циклических сумм, вычисляемых генератором на каждый кадр.

#### *Детектор циклических сумм – CRC-N Syndrome Detector*

Блок подсчитывает контрольные суммы для всего входного кадра. С первого выхода блока поступают информационные слова с удаленными контрольными суммами. Со второго выхода блока поступает вектор, длина которого равна количеству контрольных сумм, а каждый элемент – результат проверки.

Настраиваемые параметры блока:

CRC-N method – Образующий полином для CRC-алгоритма;

Initial states – Двоичный вектор длиной, равной степени образующего полинома, определяющий начальные состояния внутренних регистров сдвига;

Checksums per frame – Число циклических сумм, вычисляемых генератором на каждый кадр.

### 3.9.2 Перемежение (Interleaving)

Библиотека «Interleaving» состоит из двух подбиблиотек:

- Блочное перемежение – Block;
- Сверточное перемежение – Convolutional.

### Блочное перемежение – Block Interleaving

Пакет поддерживает работу с произвольными блочными перемежителями. Имеется также дополнительная поддержка нескольких более узких классов перемежителей – матричное перемежение, алгебраическое перемежение и перемежение по псевдослучайному закону.

#### *Алгебраический перемежитель – Algebraic Interleaver*

Алгебраический перемежитель реорганизует входной двоичный вектор используя алгебраически заданный закон перемежения.

Для  $N$  элементов входного вектора задается тип таблицы перемежения из списка с соответствующими параметрами.

Настраиваемые параметры блока:

Type – Тип таблицы перемежения:

- Welch-Costas;
- Takeshita-Costello;

Number of elements - Количество элементов  $N$  во входном векторе;

Multiplicative factor – Коэффициент, используемый при расчете выходного вектора. Используется только при типе таблицы Takeshita-Costello;

Cyclic shift – Число на которое блок сдвигает индексы при вычислении таблицы коммутации. Используется только при типе таблицы Takeshita-Costello;

Primitive element – Простейший элемент  $N$ -го порядка в конечных полях Галуа  $GF(N+1)$ . Используется только при типе таблицы Welch-Costas.

#### *Общий блочный перемежитель – General Block Interleaver*

Общий блочный перемежитель реорганизует входной двоичный вектор без повторения или пропуска какого-либо элемента. Входной вектор может быть вещественным или комплексным.

Для  $N$  элементов входного вектора задается, в качестве параметра блока, вектор, элементы которого являются порядковыми индексами элементов входного вектора.

Настраиваемые параметры блока:

Elements – Вектор, длиной  $N$ , элементы которого являются порядковыми индексами элементов входного вектора.

*Пример:*

Пусть входной вектор – [40, 56, 7, 88]. Параметр “Elements” = [4, 2, 1, 3]. Тогда выходная последовательность равна: [88, 56, 40, 7].

#### *Матричный перемежитель – Matrix Interleaver*

Матричный перемежитель производит перемежение путем заполнения матрицы входным сигналом строка за строкой, и затем отправляя содержимое матрицы в выходной порт столбец за столбцом.

Настраиваемые параметры блока:

Number of rows – Число строк матрицы;

Number of columns – Число столбцов матрицы.

*Матричный диагональный перемежитель – Matrix Helical Interleaver*

Матричный диагональный перемежитель производит перемежение путем заполнения матрицы входным сигналом строка за строкой, и затем отправляя содержимое матрицы в выходной порт по диагонали.

Специальный параметр Array Step Size регулирует наклон диагонали.

Настраиваемые параметры блока:

Number of rows – Число строк матрицы;

Number of columns – Число столбцов матрицы;

Array step size – Наклон диагонали.

*Псевдослучайный перемежитель – Random Interleaver*

Псевдослучайный перемежитель реорганизует входной двоичный вектор используя псевдослучайное перемежение.

Специальный параметр задает номер псевдослучайной последовательности.

Настраиваемые параметры блока:

Number of elements – Число элементов во входном векторе;

Initial seed – Номер псевдослучайной последовательности.

Используется для генератора случайных чисел.

*Сверточное перемежение – Convolutional Interleaving*

Пакет поддерживает работу с различными сверточными перемежителями. Имеется также дополнительная поддержка нескольких более узких классов перемежителей – общее сверточное перемежение и диагональное сверточное перемежение. Общая схема сверточного перемежителя представлена на рисунке 3.14.

*Общий сверточный перемежитель – General Multiplexed Interleaver*

Общий сверточный перемежитель реорганизует входную последовательность. Он состоит из набора регистров сдвига, каждый со своей задержкой.

Входной сигнал может быть как вещественным так и комплексным.

Параметр Interleaver Delay описывает вектор длина которого равна количеству регистров сдвига, а элементы его означают задержку для каждого регистра.

Настраиваемые параметры блока:

Interleaver delay – Вектор, содержащий задержки для каждого регистра сдвига;

Initial conditions – Начальное состояние регистров сдвига.

*Сверточный перемежитель – Convolutional Interleaver*

Сверточный перемежитель реорганизует входную последовательность. Он состоит из набора регистров сдвига, каждый со своей задержкой.

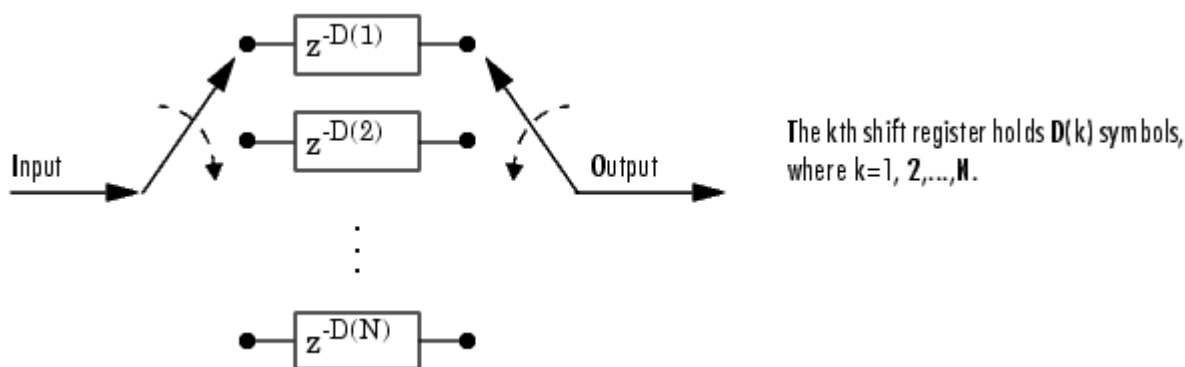


Рисунок 3.14 – Общая схема сверточного перемежителя

Входной сигнал может быть как вещественным так и комплексным.

Параметр Register length step определяет увеличение элементов памяти каждого последующего регистра сдвига. Параметр Rows of shift registers определяет количество регистров сдвига.

Настраиваемые параметры блока:

Rows of shift registers – Количество регистров сдвига;

Register step length – Количество дополнительных элементов памяти в каждом последующем регистре;

Initial conditions – Начальное состояние регистров сдвига.

*Сверточный диагональный перемежитель – Convolutional Helical Interleaver.*

Сверточный диагональный перемежитель производит перемежение путем заполнения матрицы входным сигналом по диагонали, и затем отправляя содержимое матрицы в выходной порт строка за строкой.

Специальный параметр Helical Array Step Size регулирует наклон диагонали.

Настраиваемые параметры блока:

Number of columns in helical array – Число столбцов в диагональном массиве [C];

Group size – Размер группы входных символов [N]. Длина входного вектора равна C×N;

Helical array step size – Сдвиг диагонали. Служит для регулировки наклона диагонали;

Initial conditions – Начальное состояние регистров сдвига.

### 3.9.3 Скремблирование (Scrambling)

Скремблирование реализуется с помощью специального блока «Scrambler» в библиотеке «Sequence Operations».

Блок «Scrambler» скремблирует входной сигнал различных видов исчисления. Если база исчисления равна  $N$ , то входной сигнал должен иметь значения от 0 до  $N-1$ .

Одна из задач скремблирования – уменьшить длину последовательности нулей или единиц входного сигнала, поскольку длинная последовательность может вызвать проблемы синхронизации на приемном конце.

На рисунке 3.15 представлена схема скремблера.

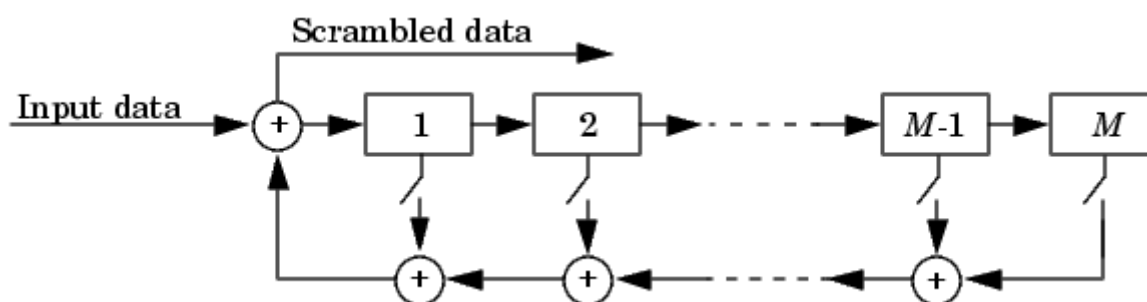


Рисунок 3.15 – Схема скремблера

Настраиваемые параметры блока:

Calculation base – база исчисления  $[N]$ . Входной сигнал должен быть представлен целыми числами из набора  $[0; N-1]$ ;

Scramble polynomial – полином, определяющий связи в схеме скремблера;

Initial states – начальные состояния регистров [8].

### 3.10 Устройства графического отображения

#### 3.10.1 Осциллограф (Scope)

Осциллограф (рисунок 3.16) отображает график зависимости величины входного сигнала от времени. Возможен просмотр нескольких сигналов одновременно, в реальном времени. Отображение производится по следующему принципу: из входного отсчета блок берет отсчеты с заданной частотой дискретизации, значение каждого отсчета сохраняется в оперативной памяти компьютера, а при отображении продлевается на весь период дискретизации входного сигнала.

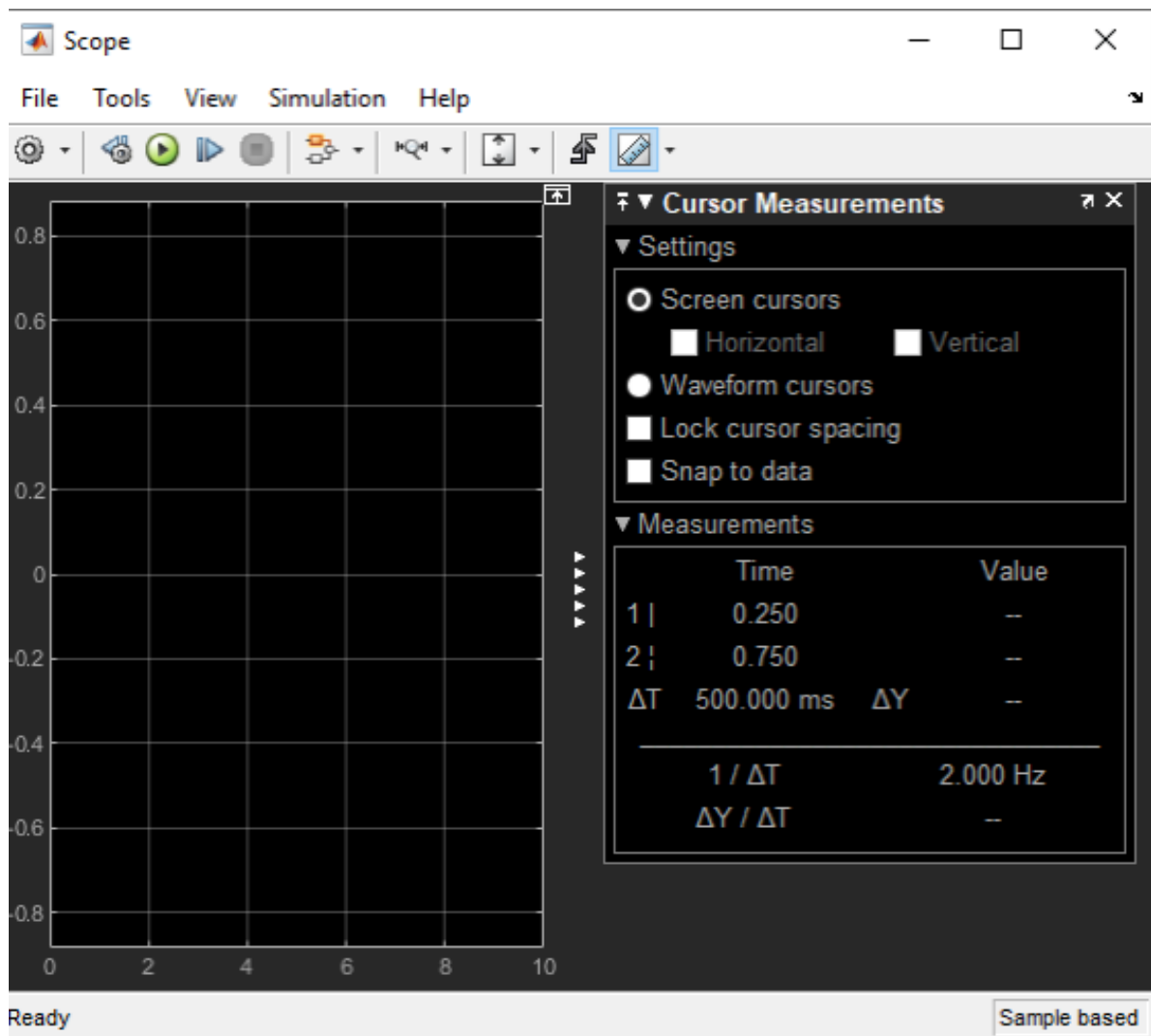


Рисунок 3.16 – Осциллограф (Scope)

### 3.10.2 Анализатор спектра (Spectrum Analyzer)

Анализатор спектра (рисунки 3.17, 3.18) буферизирует входной дискретный сигнал, после чего производит над накопленным массивом процедуру быстрого преобразования Фурье. Полученная спектральная характеристика сигнала отображается на интервале

$$[-f_d; f_d] \text{ либо } [-f_d/2; f_d/2]$$

где  $f_d$  – частота дискретизации.



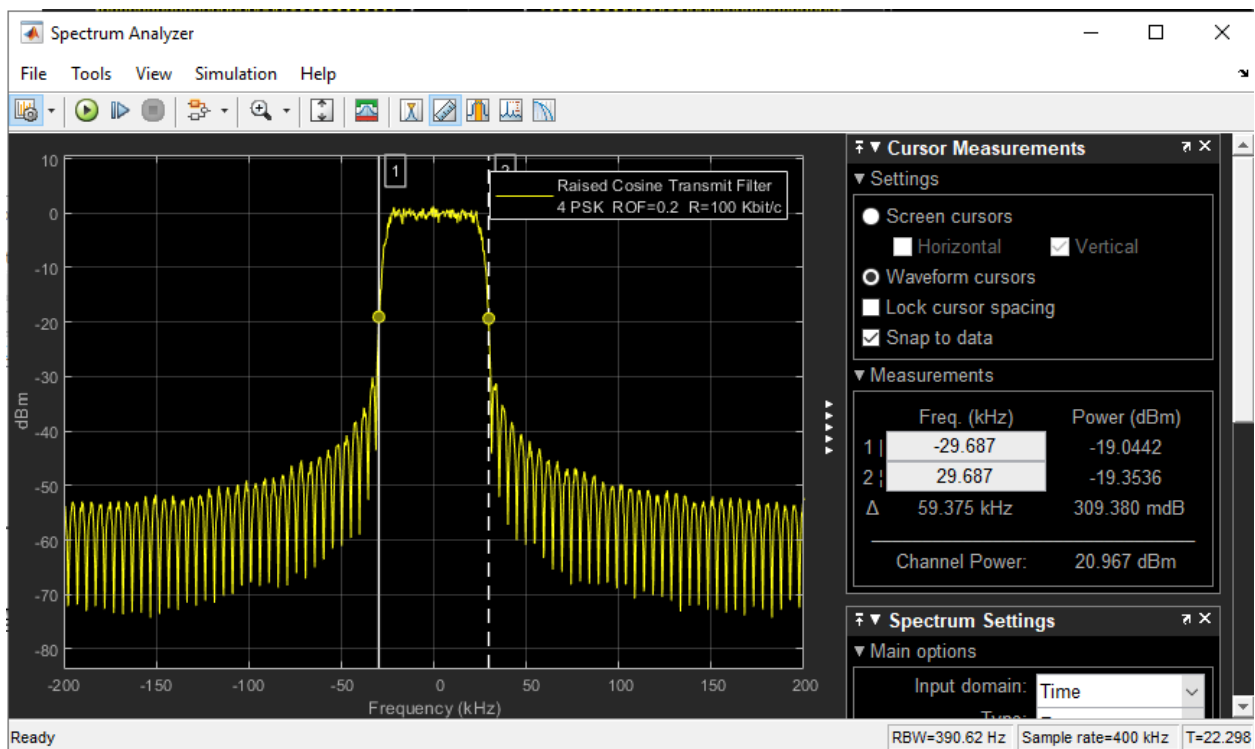


Рисунок 3.17 – Анализатор спектра с настройками Cursor Measurements

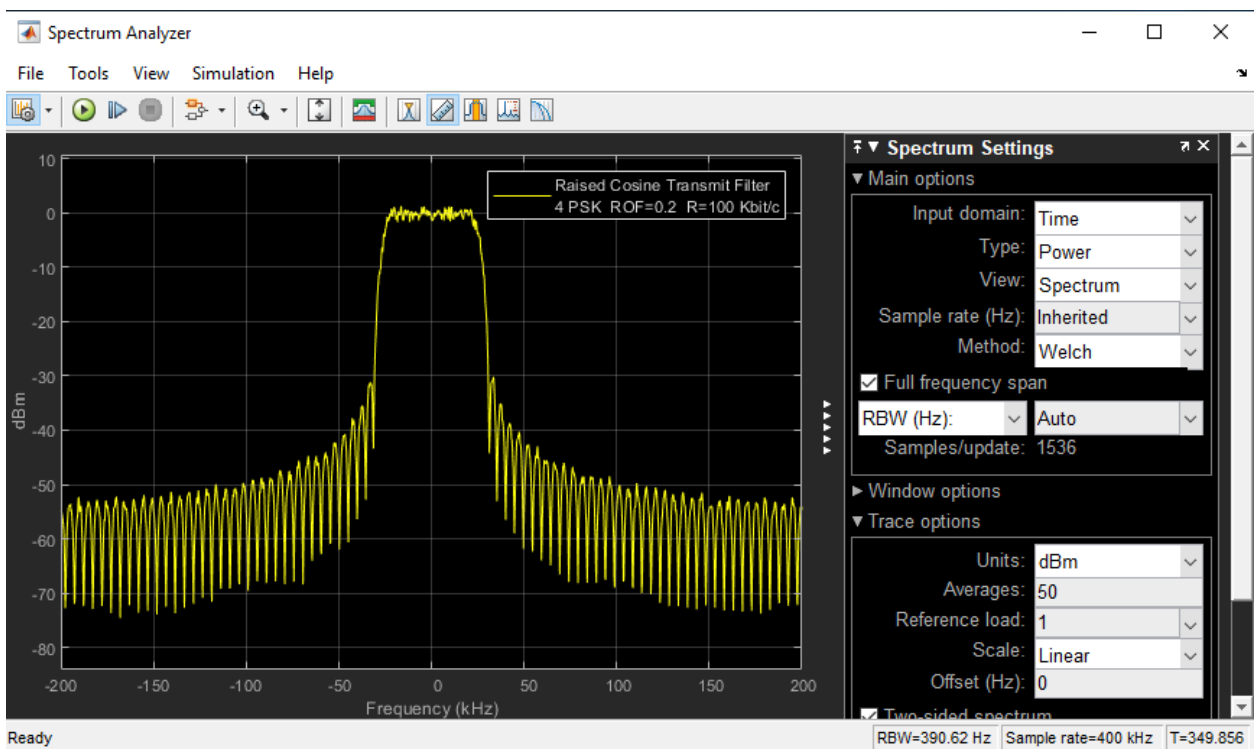


Рисунок 3.18 – Анализатор спектра с настройками Spectrum Settings

### 3.10.3 Вектограф (Constellation diagram)

Блок отображает созвездие (Constellation) сигнальных точек модулированного сигнала на фазово-амплитудной плоскости в синфазной и квадратурной координатах. На рисунке 3.19 приведен пример созвездия на выходе модулятора QPSK.

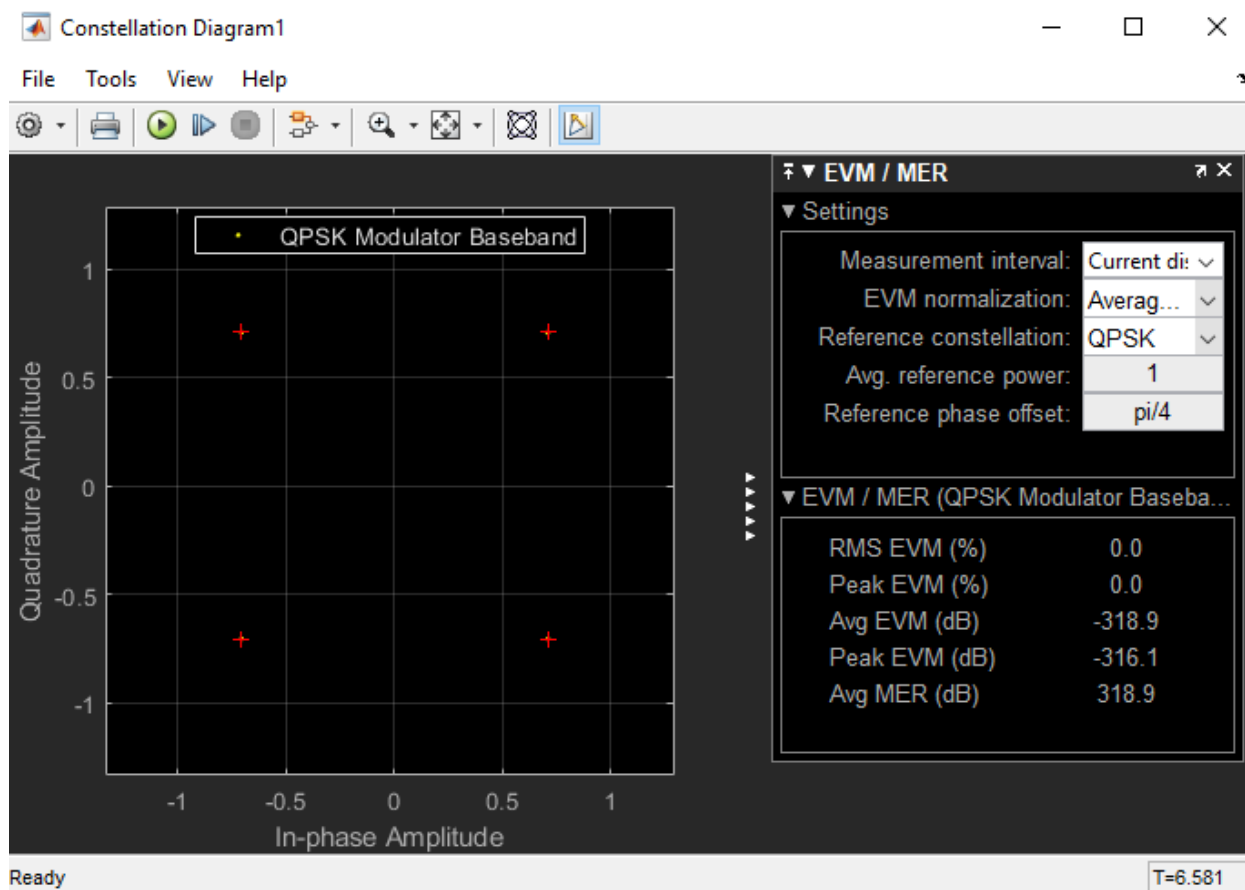


Рисунок 3.19 – Созвездие на выходе модулятора QPSK и настройки блока Constellation diagram

Величина вектора ошибки EVM (Error Vector Magnitude) (рисунок 3.20), иногда также называемая относительной ошибкой созвездия - это мера, используемая для количественной оценки качества работы цифрового радио-передатчика или приемника. Сигнал, переданный идеальным передатчиком и принятый приемником, будет иметь все точки созвездия точно в идеальных местах. Шум, искажения, паразитные сигналы и фазовый шум приводят к тому, что фактические точки созвездия отклоняются от идеальных местоположений. В этом случае EVM является мерой того, насколько далеко находятся точки созвездия от идеальных мест.

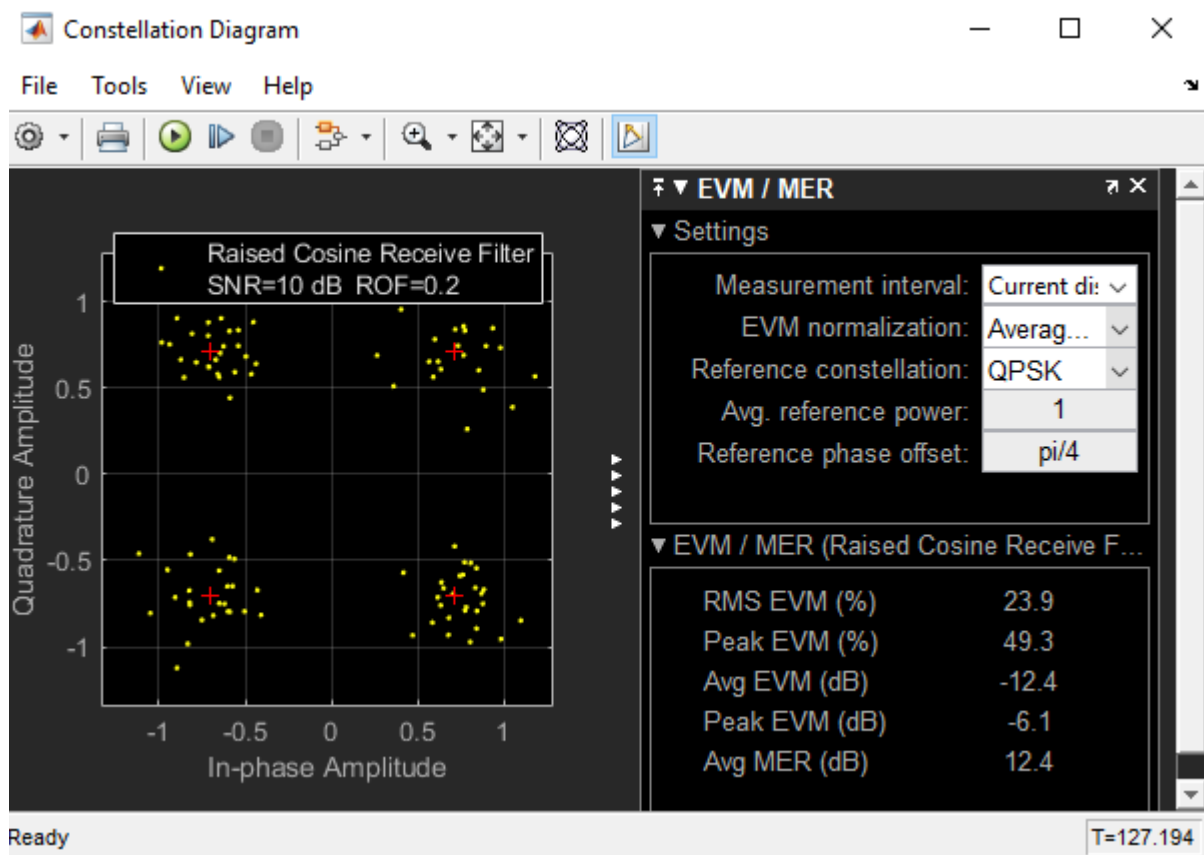


Рисунок 3.20 – Созвездие сигнала QPSK на выходе Raised Cosine Receive Filter

Для вычисления величины (амплитуды) вектора ошибки EVM для каждого символа сигнала необходимо найти значения вектора ошибки как разность между идеальным опорным положением сигнальной точки  $P_{ref}$  и положением сигнальной точки реального измеряемого сигнала  $P_{err}$  (рисунок 3.21). Как правило, величина вектора EVM нормализуется относительно среднеквадратичного значения (root-mean-square – RMS) мощности символа:

$$EVM(dB) = 10 \log_{10} \left( \frac{P_{err}}{P_{ref}} \right). \quad (3.6)$$

Другим параметром оценки отклонения реальных сигнальных точек от их идеального положения является MER (Modulation Error Ratio). Параметр MER определяется выражением

$$EVM(dB) = 10 \log_{10} \left( \frac{P_{ref}}{P_{err}} \right). \quad (3.7)$$

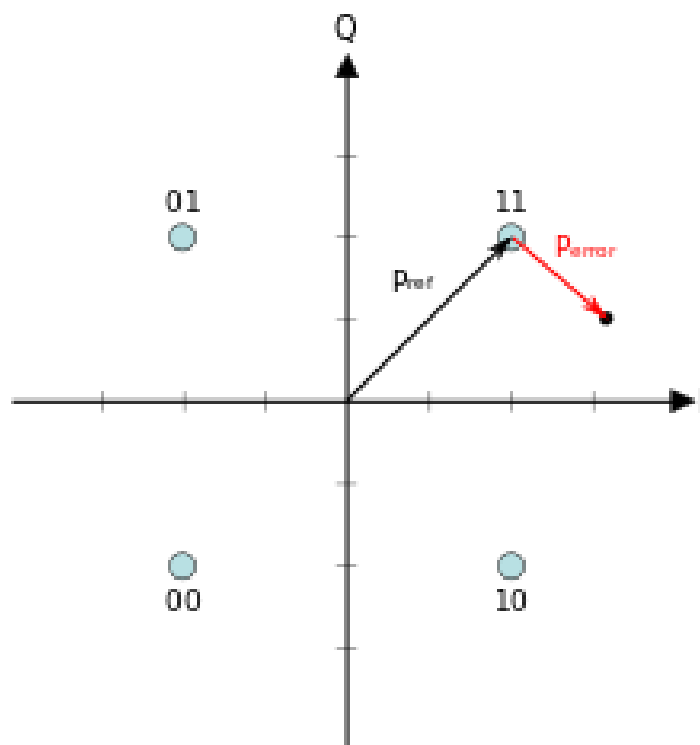


Рисунок 3.21 – К определению векторов опорного сигнала  $P_{ref}$  и сигнала ошибки  $P_{err}$

Величина MER в дБ численно равна величине EVM только с положительным знаком.

### 3.11 Устройства регистрации и подсчета ошибок (Error Rate Calculation)

Блок (рисунок 3.22) вычисления частоты ошибок сравнивает входные данные передатчика Tx с выходными данными приемника Rx. Он вычисляет частоту ошибок как текущую статистику, путём деления общего число неравных пар элементов данных (количество ошибочных бит) на общее число элементов входных данных из одного источника.

Этот блок используется для вычисления частоты символьных или битовых ошибок, поскольку он не учитывает величину разницы между элементами входных данных. Если входы являются битами, то блок вычисляет частоту битовых ошибок. Если входные данные являются символами, то он вычисляет частоту ошибок символа.

Для счетчика задается задержка вычисления, используемая в случае, если принимаемый сигнал отстает от переданного по времени. Также предусмотрена возможность остановки симуляции при достижении заданного числа ошибок или принятых символов.

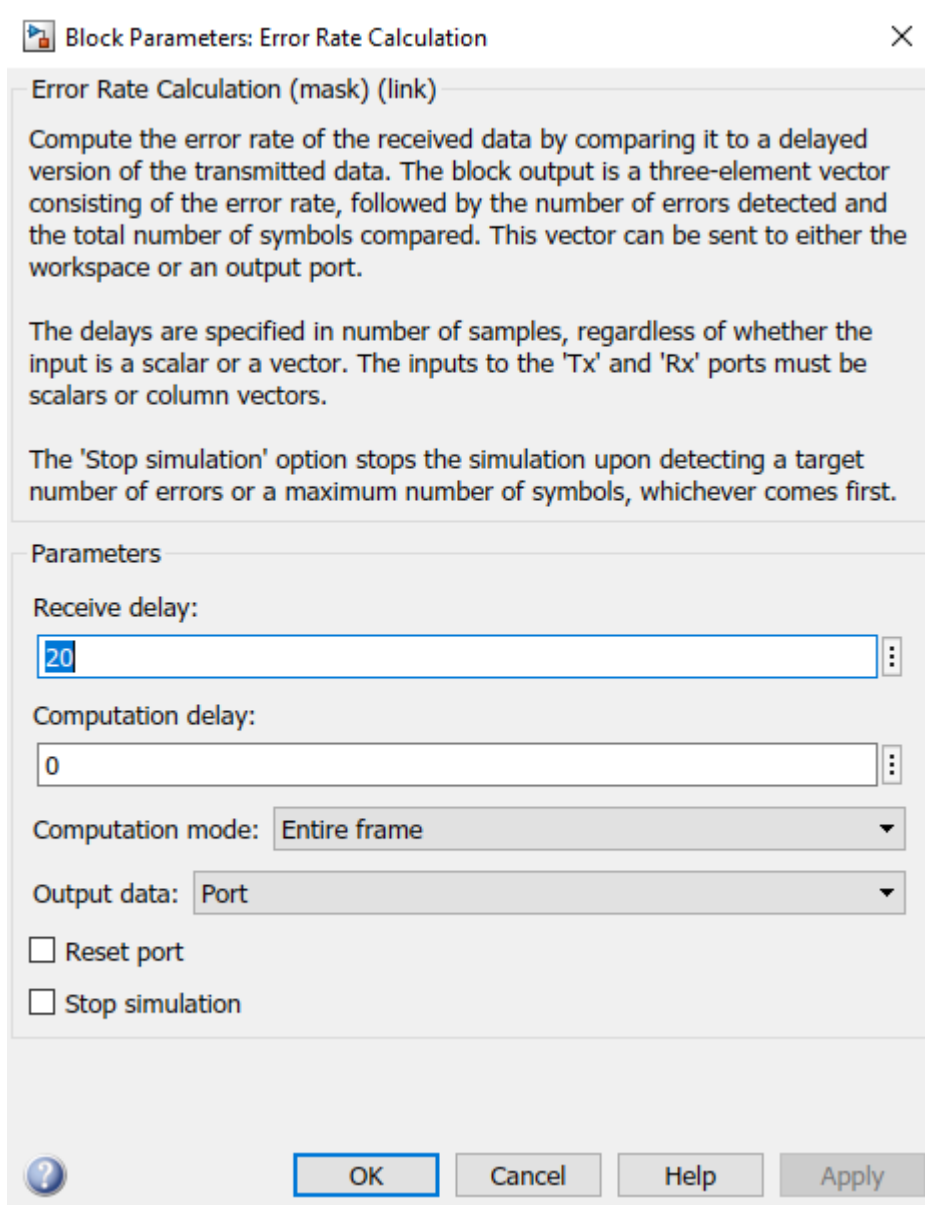


Рисунок 3.22 – Настройки блока Error Rate Calculation

#### *Блок определения задержки Find Delay*

Блок поиска задержки (рисунок 3.23) находит величину задержки между цифровым сигналом на входе передатчика sRef и цифровым сигналом на выходе приёмника sDel. Задержка вызвана обработкой сигнала в фильтрах на передаче и на приёме. Порт вывода с надписью delay выводит задержку в единицах выборки (samples). Параметр длины окна корреляции блока (correlation window length) указывает, сколько выборок сигналов блок использует для вычисления перекрестной корреляции. Полученное в блоке Find Delay значение задержки заносится в поле Receive delay блока Error Rate Calculation.

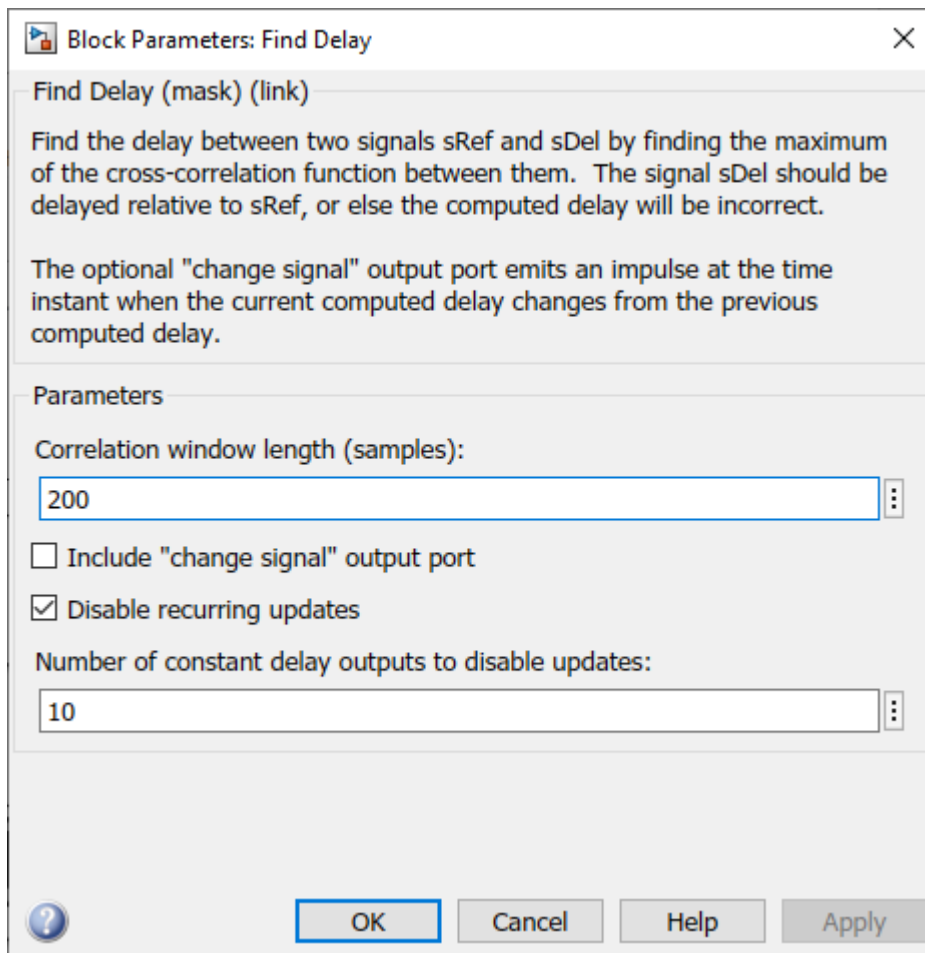


Рисунок 3.23 – Настройки блока Find Delay

### 3.12 Анализатор результатов моделирования (Bit Error Rate Analysis)

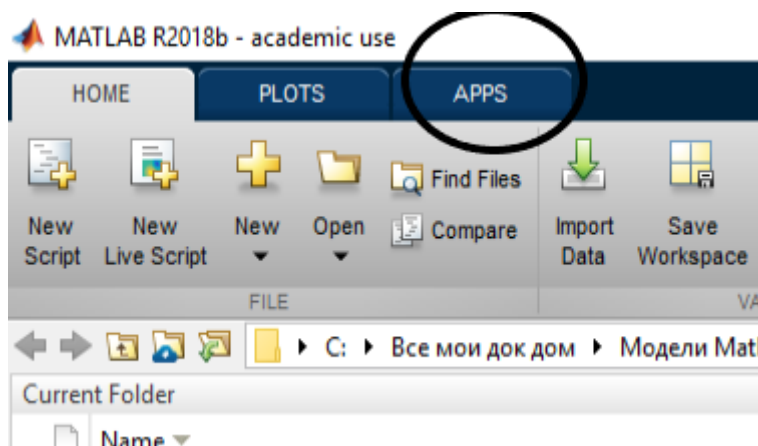
Для анализа частоты битовых ошибок системы используется специализированный графический пользовательский интерфейс – Bit Error Rate Analysis.

Для открытия блока необходимо в основном окне MATLAB R2018b выбрать кнопку APPS (рисунок 3.24,а), а затем в разделе SIGNAL PROCESSING AND COMMUNICATION открыть блок Bit Error Rate Analysis (рисунок 3.24,б).

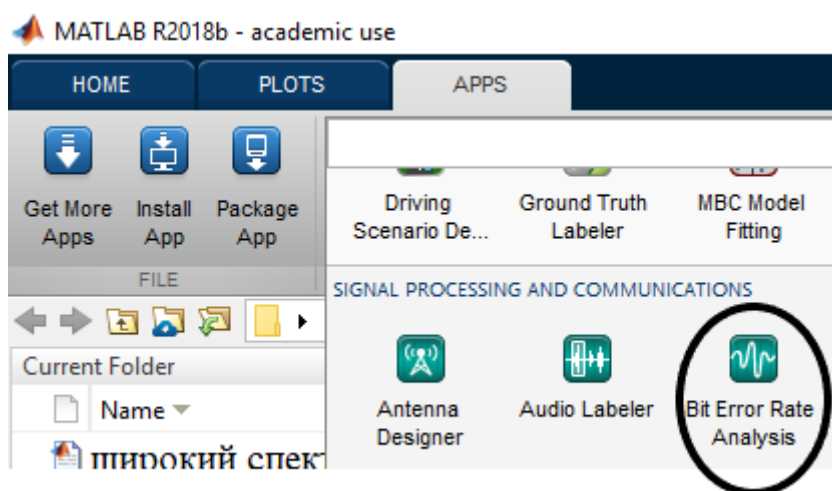
Блок Bit Error Rate Analysis (рисунок 3.25) позволяет систематизировать информацию о частоте битовых ошибок для системы связи, реализованной в MATLAB или SIMULINK и построить график зависимости коэффициента ошибок от отношения  $E_b/N_0$ .

Кроме того, в программе уже заложены формулы для вычисления коэффициента BER для некоторых (режим Theoretical) типов каналов (например, AWGN), видов модуляции (например, M-PSK, M-QAM); что

позволяет пользователю сравнить результаты анализа собственной системы (режимы Semianalytic, Monte Carlo) с теоретическими.



a)



b)

Рисунок 3.24 – Открытие блока Bit Error Rate Analysis

Bit Error Rate Analysis наблюдает симуляцию указанной системы в течении заданного промежутка времени либо до достижения требуемого значения  $E_b/N_0$  или BER, усредняет результаты и строит график зависимости коэффициента ошибок от отношения сигнал/шум.

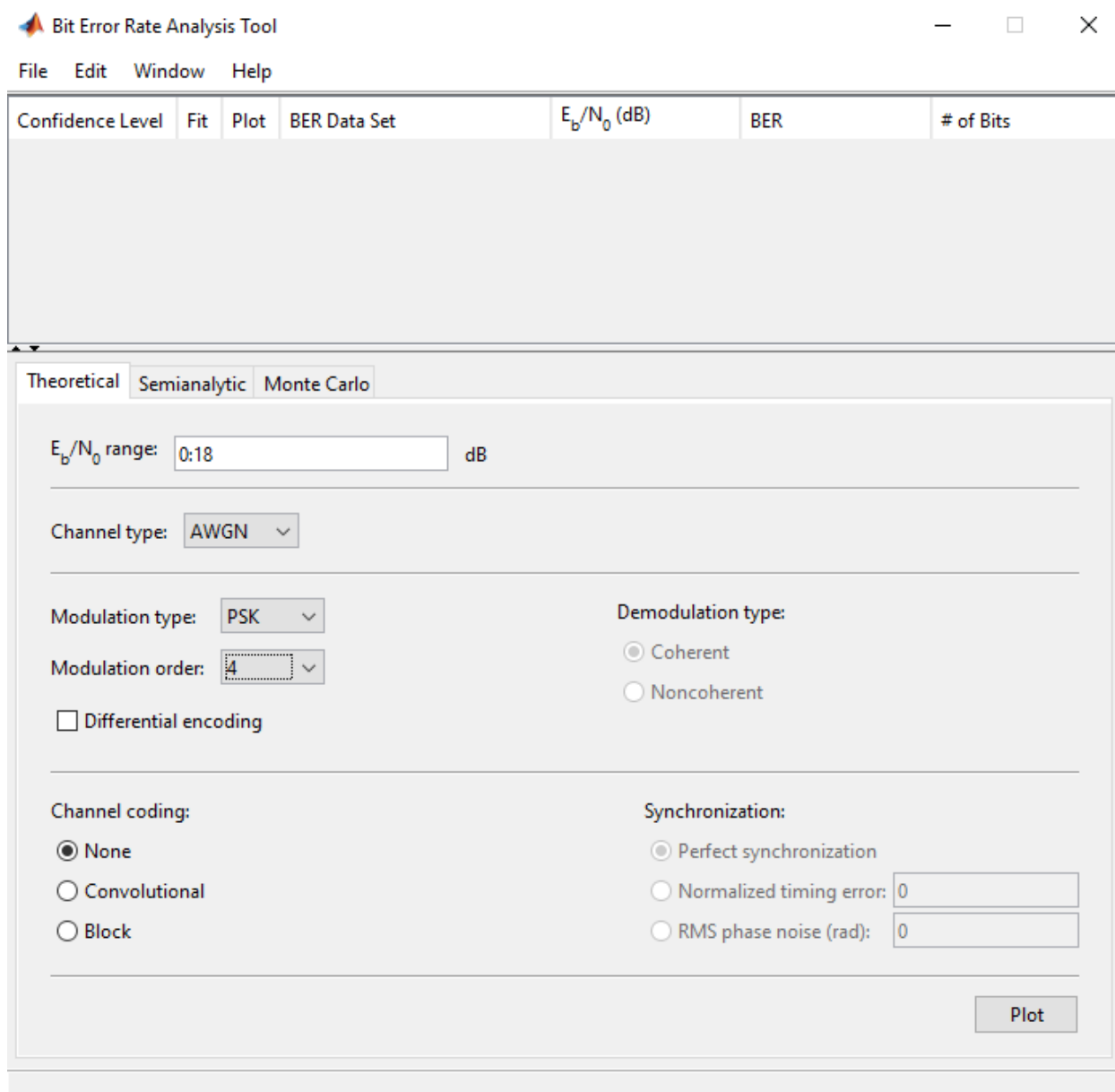


Рисунок 3.25 – Настройки рабочей панели блока Bit Error Rate Analysis

В Bit Error Rate Analysis заложен большой набор функций, которые позволяют:

- строить несколько графиков на одной координатной плоскости;
- строить графики с заданным шагом и в заданном диапазоне значений  $E_b/N_0$ ;
- просматривать как усредненное значение коэффициента ошибок, так и весь разброс его значений, имевший место в процессе эксперимента;
- сохранять результаты моделирования в виде числового файла, для дальнейшей обработки и анализа.

Кроме того, там отображаются результаты уже сделанных симуляций, и команды для работы с графиком зависимости BER от  $E_b/N_0$ . Пример



графика, построенного с помощью Bit Error Rate Analysis (в режиме Theoretical путём нажатия кнопки Plot (рисунок 3.25)), приведен на рисунке 3.26.

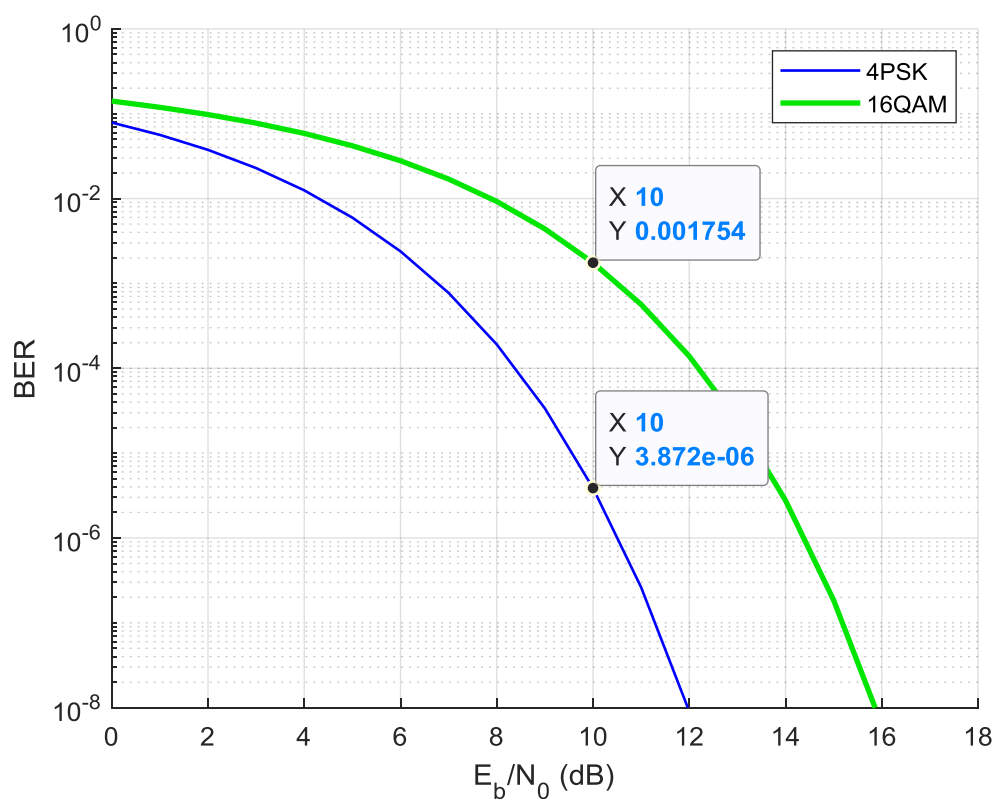


Рисунок 3.26 – График, построенный с помощью Bit Error Rate Analysis

## Контрольные вопросы

1. Каковы преимущества и недостатки моделирования систем при помощи ПК?
2. Назовите несколько современных систем компьютерной математики (СКМ)
3. В чем особенность визуального программирования?
4. Привести последовательность этапов, которые проходит сигнал в процессе симуляции
5. Для чего используется низкочастотная модуляция (baseband modulation)?
6. Что такое кадр (frame) и для чего он используется?
7. Какие возможности предоставляет пользователю Bit Error Rate Analysis?

## **4 Исследование влияния модуляции сигнала на коэффициент битовых ошибок и полосочастот для модуляций M-PSK и M-QAM**

### **4.1 Цель исследования**

Данное исследование ставит своей целью пронаблюдать влияние белого гауссовского шума на многофазные методы модуляции (М-ФМ, М-ДФМ и М-КАМ); а также сравнить эти виды модуляции по эффективности использования спектра и устойчивости к замираниям в канале радиосвязи. Указанные методы модуляции были выбраны как одни из наиболее часто используемых в современных средствах радиосвязи и радиодоступа. В частности, модуляция M-PSK применяется в сотовых системах связи второго и третьего поколения, и наряду с M-QAM модуляцией, используется в стандартах беспроводной связи IEEE 802.11.

Исследование было проведено на базе программных моделей, реализованных в среде программного пакета MATLAB . Упрощенная модель ЦСПС, используемая в исследовании, включает в себя: генератор сигнала, кодер, модулятор, модель канала радиосвязи, демодулятор, декодер, а также ряд вспомогательных блоков. Более подробное описание модели и входящих в её состав блоков приведено в разделе 3.

### **4.2 Сравнение помехоустойчивости разных видов модуляции в условиях белого Гауссовского шума**

Аддитивный белый гауссовский шум (AWGN) является простейшим видом помехи в радиосвязи. Оценка цифрового канала по зависимости коэффициента ошибок (BER) от отношения  $E_b/N_0$  является важнейшим мерилем его качества. На рисунках 4.1, 4.2 показано, как воздействие AWGN отражается на осциллограмме и вектограмме сигнала 16-QAM. Для других многофазных видов модуляции изменения созвездия и осциллограммы будут выглядеть аналогично.

Графические зависимости частоты возникновения битовых ошибок от отношения  $E_b/N_0$  для сигналов модулированных M-PSK полученные с помощью программной модели в системе MATLAB приведены на рисунке 4.3.

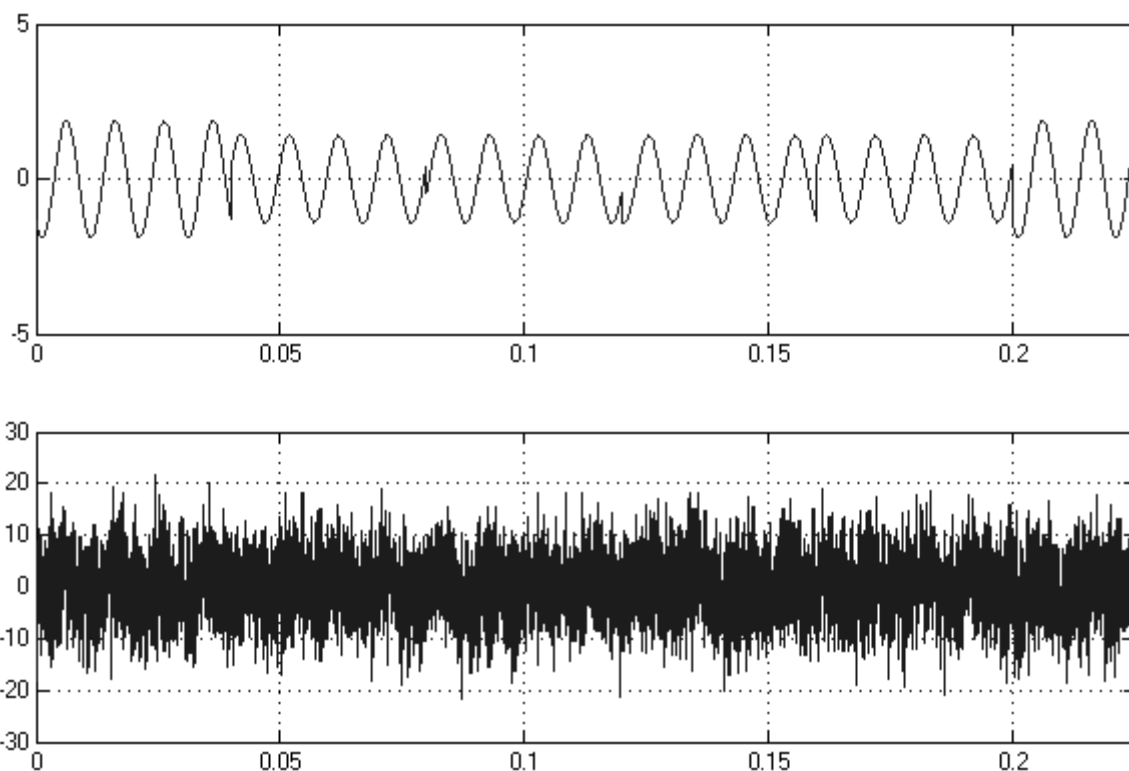


Рисунок 4.1 – Пример воздействия AWGN на модулированный сигнал 16-QAM

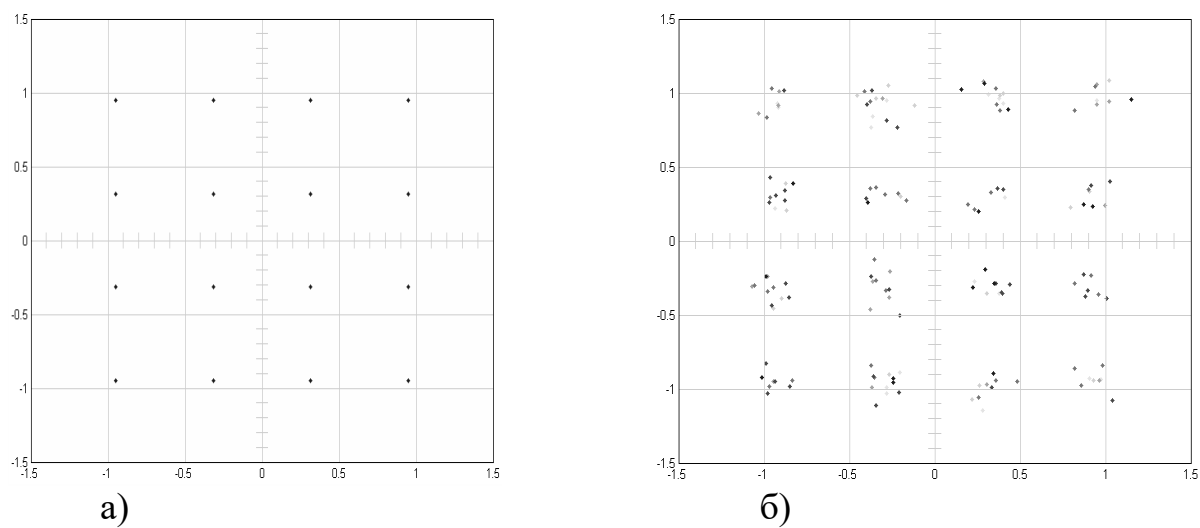


Рисунок 4.2 – Воздействие AWGN на созвездие 16-QAM  
а) до воздействия AWGN, б) после воздействия AWGN

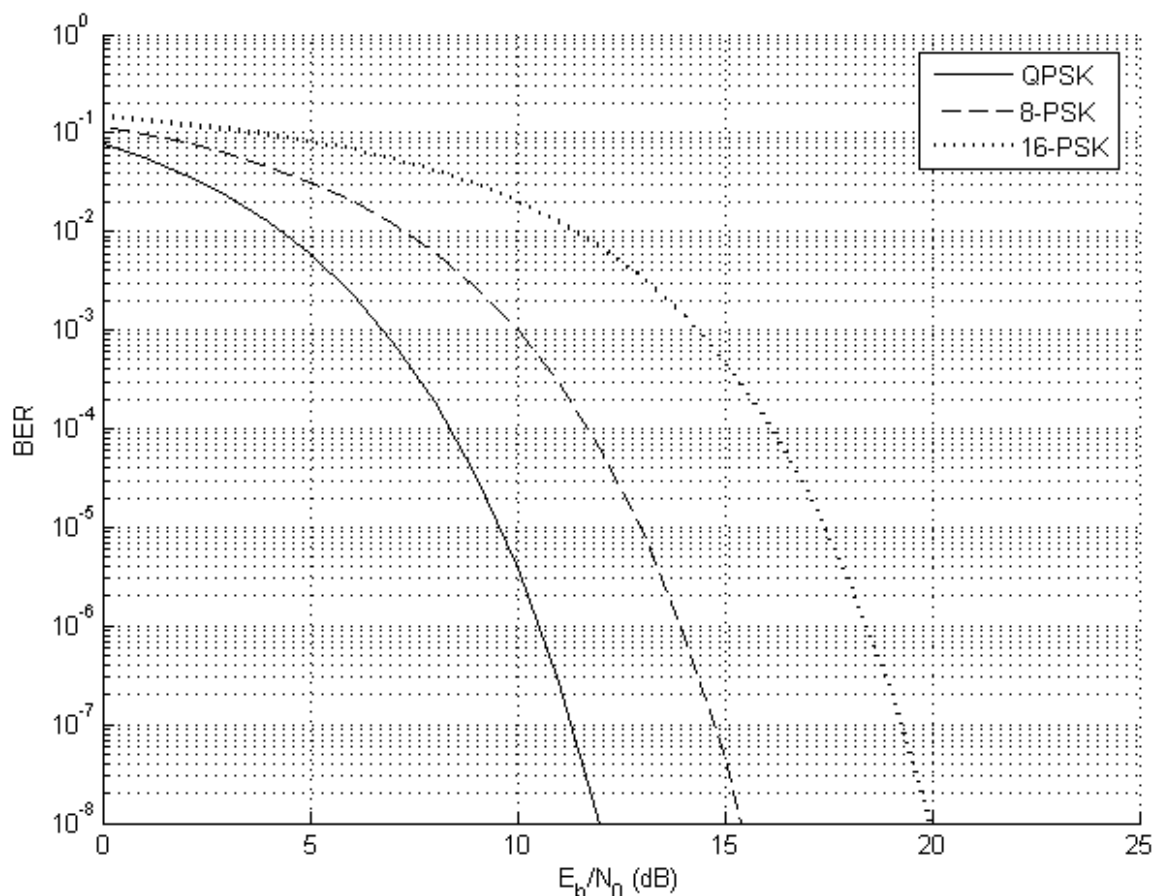


Рисунок 4.3 – Графики зависимости частоты возникновения битовых ошибок от отношения  $E_b/N_0$

Как уже говорилось, в данном случае мы имеем дело с M-позиционной передачей сигналов, то есть в каждый момент времени процессор рассматривает  $k$  бит. Он указывает модулятору произвести один из  $M = 2^k$  сигналов; частным случаем, когда  $k = 1$ , является бинарная передача сигналов. Как видно из графика, по мере роста  $k$  график перемещается в направлении увеличения вероятности ошибки. На первый взгляд, преимущество BPSK очевидно, и может возникнуть вопрос: стоит ли вообще использовать многофазную модуляцию M-PSK, если она приводит к более высокой вероятности ошибки по сравнению с бинарной PSK? Многофазная передача сигналов применяется во многих реальных системах, и объясняется это тем, что зависимость вероятности ошибки от  $E_b/N_0$  не является единственным критерием качества. Модулированный сигнал имеет несколько характеристик, а на рисунке 4.3 явно показана только вероятность ошибки.

При разработке системы цифровой радиосвязи учитывается совокупность параметров, которая должна обеспечивать некоторое

компромиссное решение. Вообще, системные компромиссы – неотъемлемая часть всех разработок цифровых систем связи. Одними из условий, к которым должен стремиться разработчик являются:

- увеличение скорости передачи данных до максимально возможной;
- минимизация вероятности появления битовой ошибки;
- минимизации потребляемой мощности, или, что тоже самое, минимизации требуемого отношения энергии одного бита к спектральной плотности мощности шума  $E_b/N_0$ ;
- минимизация ширины полосы пропускания.

Удовлетворить все эти требования одновременно невозможно – очевидно, что первые два противоречат последним двум. Однако, подбор различных параметров таким образом, чтобы они обеспечивали необходимый компромисс между характеристиками системы – задача, решаемая на этапе разработки любой системы связи.

#### 4.2.1 Сравнение M-PSK и M-DPSK на примере BPSK и BDPSK

Как известно, кроме модуляции PSK существует схожий метод линейного кодирования – модуляция DPSK, которая может обнаруживаться как когерентно, так и некогерентно. В некогерентных системах не предпринимается попытки определить действительное значение фазы поступающего сигнала; при когерентном обнаружении решение принимается на основе степени корреляции между эталонным и принятым сигналом.

Произведем сравнение этих двух схем модуляции (PSK и DPSK) на примере их бинарных вариантов. Напомним, что в схеме PSK существует однозначное соответствие между двоичным импульсом и соответствующим ему сигналом; а при использовании DPSK двоичный ноль представляется сигнальным пакетом, фаза которого совпадает с фазой предыдущего посланного пакета, а двоичная единица представляется сигнальным пакетом с фазой, противоположной фазе предыдущего пакета. Такая схема называется дифференциальной, поскольку сдвиг фаз выполняется относительно предыдущего переданного бита, а не относительно какого-то эталонного сигнала.

На рисунке 4.4 приведены графики зависимости BER от  $E_b/N_0$  для трех схем модуляции: BPSK, бинарная DPSK с когерентным обнаружением и бинарная DPSK с некогерентным обнаружением.

Приведенные зависимости подтверждаются математически формулами для, битовых вероятностей ошибок

$$P_{b,BPSK} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (4.1)$$

для некогерентной *DBPSK*

$$P_{b,DPSK} = \frac{1}{2} \exp\left(-\frac{E_b}{N_0}\right) \quad (4.2)$$

для когерентной *DBPSK*

$$P_{b,DPSK} = 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \left[1 - Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\right] \quad (4.3)$$

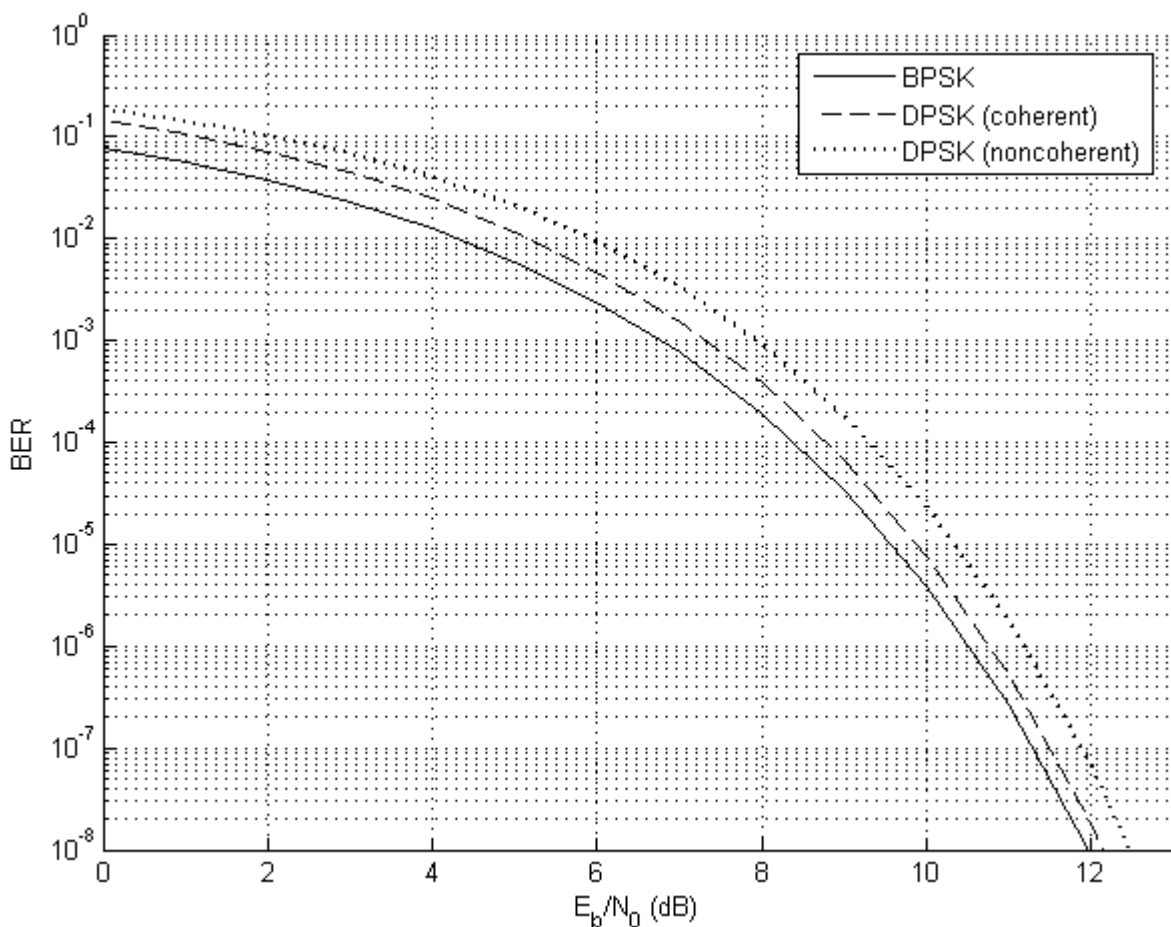


Рисунок 4.4 – Зависимость BER от  $E_b/N_0$  для разных вариантов PSK

Хорошо видно, что модуляция DPSK с некогерентным обнаружением наиболее чувствительна к воздействию гауссовского шума: по сравнению с PSK проигрыш составляет около двух дБ. Если сравнить вероятность ошибки для когерентной DPSK с вероятностью ошибки когерентной схемы BPSK, видно, что при равных  $P_B$  схема DPSK требует приблизительно на 0,5 дБ

большого отношения  $E_b/N_0$ , чем схема BPSK (например, для  $P_B < 10^{-4}$ ); то есть существует незначительное ухудшение достоверности обнаружения по сравнению с когерентным обнаружением сигналов в модуляции PSK. Это вызвано дифференциальным кодированием, поскольку любая отдельная ошибка обнаружения обычно приводит к принятию двух ошибочных решений. Однако, систему DPSK реализовать легче, чем систему PSK, поскольку приемник DPSK не требует фазовой синхронизации. По этой причине иногда предпочтительнее использовать менее эффективную схему DPSK, чем более сложную схему PSK.

Таким образом, мы видим, что из трех рассматриваемых двухпозиционных видов модуляции наиболее удачной с точки зрения помехоустойчивости является схема BPSK с когерентным обнаружением. Полученный результат распространяется и на схемы модуляции большей позиционности; то есть из M-PSK, когерентной M-DPSK и некогерентной M-DPSK наилучшей помехоустойчивостью обладает M-PSK.

#### 4.2.2 Сравнение модуляций M-PSK и M-QAM

Одним из наиболее часто применяющихся в цифровых системах радиосвязи методов модуляции, является многопозиционная QAM модуляция. Произведем сравнение помехоустойчивости для двух многофазных методов модуляции PSK и QAM (рисунок 4.5).

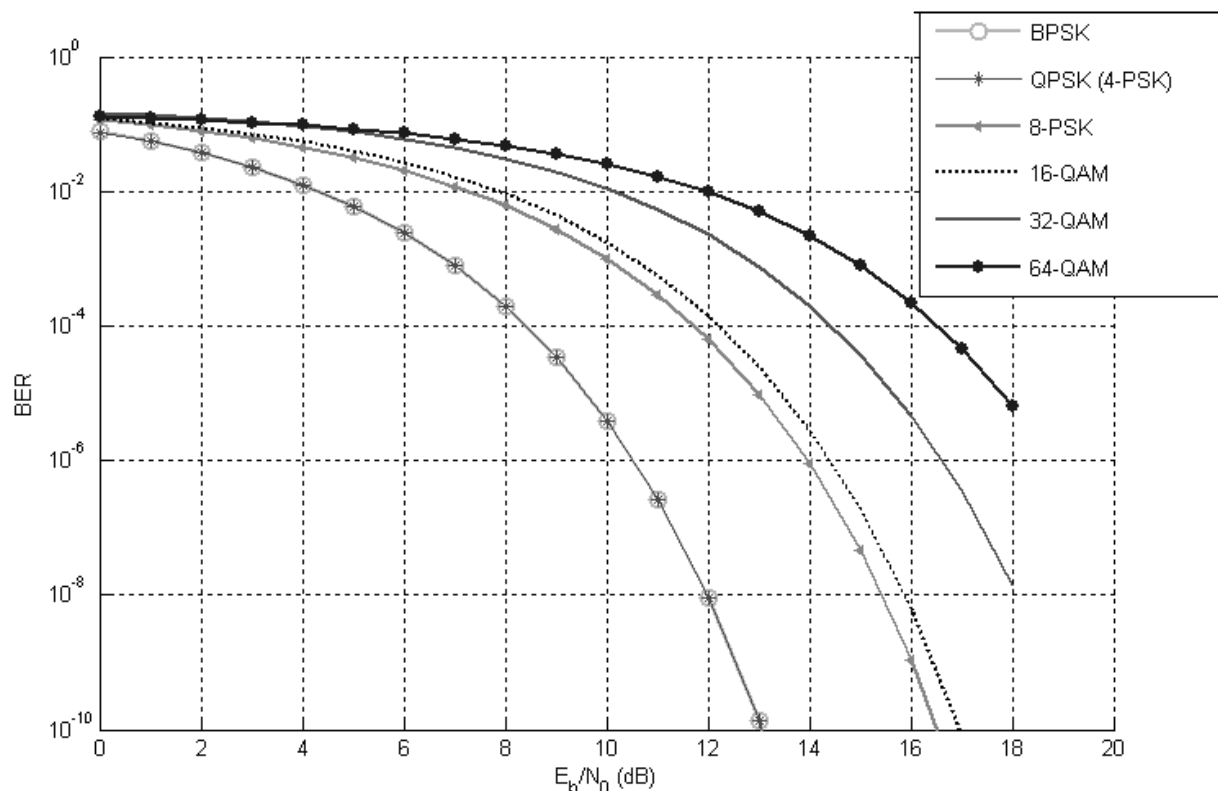


Рисунок 4.5 - Зависимость BER от  $E_b/N_0$  для разных вариантов PSK и QAM



На рисунке 4.5 приведены графики зависимости BER от  $E_b/N_0$  для этих видов модуляции при обнаружении в условиях гауссовского шума. Наблюдаем следующую закономерность: при увеличении позиционности модуляции её помехоустойчивость понижается.

Одной из рабочих характеристик, не представленных на рисунке 4.5 явно, является необходимая ширина полосы частот. Для М-позиционных многофазных кривых, приведенных на рисунке 4.5, рост величины  $k$  позволяет получать большую скорость передачи битов при той же ширине полосы частот или, что то же самое, при фиксированной скорости передачи данных уменьшить необходимую пропускную способность канала. Следовательно, графики вероятности ошибки при многофазной передаче показывают, что М-позиционная передача сигналов представляет средство реализации компромиссов между параметрами системы: скоростью передачи данных и используемой полосой частот. В случае многофазной передачи сигналов эффективность использования частотного диапазона может быть получена за счет вероятности ошибки.

### **4.3 Исследование эффективности использования частотного спектра при различных видах модуляции**

#### **4.3.1 Производительность**

При рассмотрении производительности различных методов модуляции первым параметром, представляющим интерес, является ширина полосы модулированного сигнала. Зависит данный параметр от многих факторов, в том числе от используемого определения ширины полосы и методов фильтрации, применяемых для создания полосового сигнала. Например, ширина полосы пропускания  $W$  для схемы BPSK равна

$$W_{BPSK} = (1 + \alpha)R \quad (4.4)$$

где  $R$  – скорость передачи битов;

$\alpha$  – rolloff factor (ROF), связан с методом фильтрации сигнала, применяемого для создания необходимой девиации частоты. Как правило,  $0 < \alpha < 1$ . Следовательно, ширина полосы непосредственно связана со скоростью передачи битов.

При использовании многофазной передачи сигналов можно добиться значительно более эффективного использования полосы:

$$W_{M-PSK} = \left( \frac{1 + \alpha}{L} \right) R = \left( \frac{1 + \alpha}{\log_2 M} \right) R \quad (4.5)$$

где  $L$  – число битов, закодированных в одной сигнальной посылке,  
 $M$  – позиционность модулятора.

В табл. 4.1 показано отношение скорости передачи данных к ширине полосы пропускания для этих двух схем. Отметим, что данное отношение также называется эффективностью использования полосы  $\gamma = R/W$  и является мерой эффективности, с которой полосу можно использовать для передачи данных.

Итак, теперь мы можем утверждать, что преимущества многоуровневых методов передачи сигналов стали очевидными.

Таблица 4.1 – Эффективность использования полосы частот для M-PSK

Фазовая манипуляция	$\alpha = 0$	$\alpha = 0,5$	$\alpha = 1$
$M = 2$	1,0	0,67	0,5
$M = 4$	2,00	1,33	1,00
$M = 8$	3,00	2,00	1,50
$M = 16$	4,00	2,67	2,00
$M = 32$	5,00	3,33	2,50

### 4.3.2 Плоскость "полоса-эффективность"

Шеннон показал, что пропускная способность канала  $C$  с аддитивным белым гауссовым шумом является функцией средней мощности принятого сигнала  $S$ , средней мощности шума  $N$  и ширины полосы пропускания  $W$ . Выражение для пропускной способности (теорема Шеннона-Хартли) можно записать следующим образом

$$C = W \log_2 \left( 1 + \frac{S}{N} \right) \quad (4.6)$$

Если  $W$  измеряется в герцах, а логарифм берется по основанию 2, то пропускная способность будет иметь размерность бит/с. Теоретически (при использовании достаточно сложной схемы кодирования) информацию по каналу можно передавать с любой скоростью  $R$  ( $R \leq C$ ) со сколь угодно малой вероятностью возникновения ошибки. Если же  $R > C$ , то кода, на основе которого можно добиться сколь угодно малой вероятности возникновения ошибки, не существует.

Зависимость  $C/W$  от  $E_b/N_0$  показана на графике зависимости  $R/W$  от  $E_b/N_0$  на рисунке 4.6.

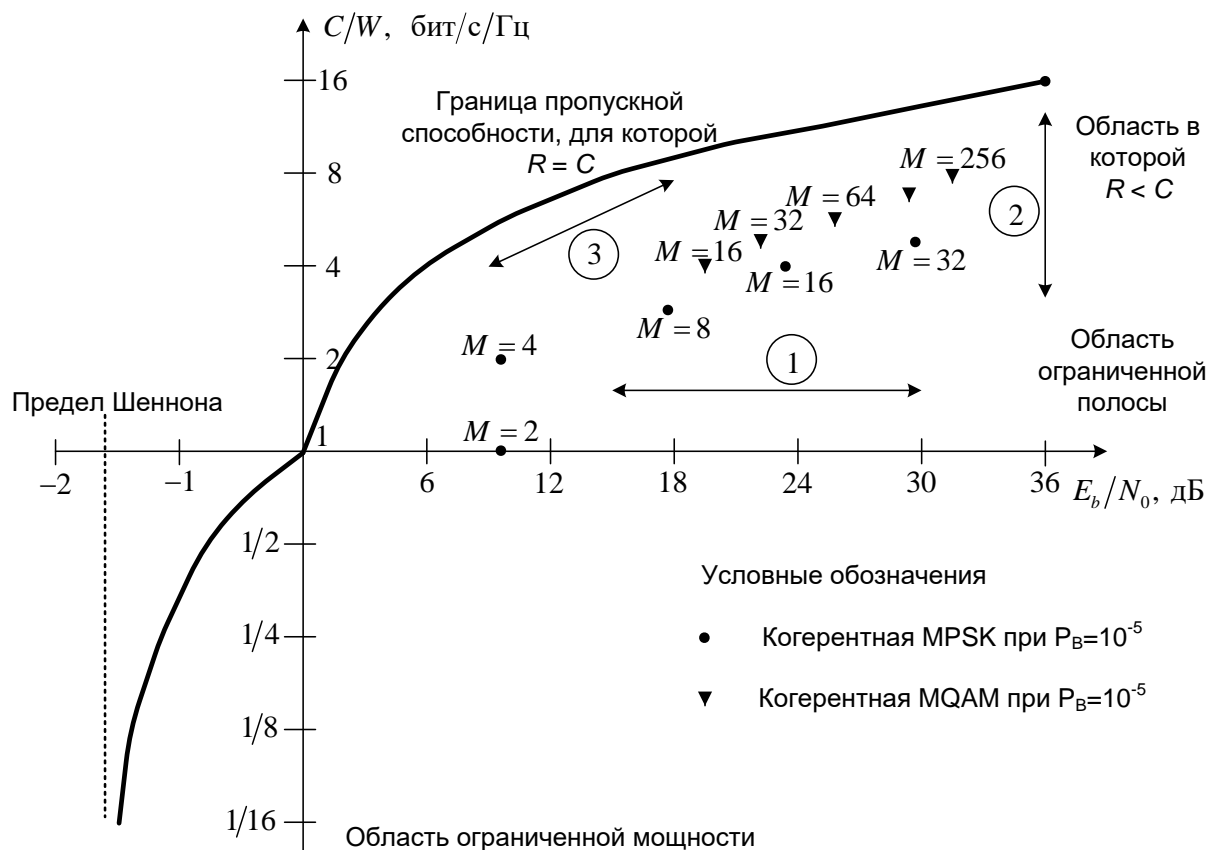


Рисунок 4.6 – Плоскость «полоса - эффективность»

Обозначим эту плоскость как плоскость "полоса-эффективность". Ордината  $R/W$  – это мера объема данных, которые можно передать через единицу полосы частот за данное время; следовательно, она отображает эффективность использования ресурса полосы пропускания. Независимая переменная  $E_b/N_0$  измеряется в децибелах.

На рисунке 4.6 кривая  $R/C$  – это граница, разделяющая область реальных прикладных систем связи и область, в которой такие системы связи теоретически невозможны. Характеристика эффективности полосы пропускания на рисунке 4.6 устанавливает предельные параметры, которые достижимы для прикладных систем. Отметим, что на рисунке 4.6 проиллюстрирована зависимость эффективности использования полосы частот от  $E_b/N_0$  для систем с одной несущей. Для систем с множественными несущими эффективность использования полосы частот зависит от разнесения несущих (и типа модуляции). В этом случае компромисс – это насколько разнесены несущие (что приводит к повышению эффективности

использования полосы частот) без возникновения неприемлемых помех соседних каналов (adjacent channel interference – ACI). [1, 2]

### **4.3.3 Эффективность использования полосы при выборе схем MPSK и MQAM**

На рисунке 4.6 показаны рабочие точки для манипуляции MPSK при вероятности битовой ошибки  $10^{-5}$ . Предполагается, что до модуляции осуществляется фильтрация по Найквисту (идеальная прямоугольная), так что минимальная двойная полоса пропускания на промежуточной частоте (intermediate frequency – IF)  $W_{IF} = 1/T$ , где  $T$  – длительность символа. Эффективность использования полосы частот описывается соотношением

$$R/W = \log_2 M \quad (4.7)$$

где  $M$  – размер набора символов (позиционность модуляции).

Для реальных каналов и сигналов производительность следует понизить, чтобы учесть увеличение полосы пропускания, требуемое для создания реализуемых фильтров. Отметим, что при модуляции MPSK  $R/W$  растет с увеличением  $M$ . Кроме того, положение рабочих точек MPSK указывает, что для модуляции BPSK ( $M = 2$ ) и квадратичной PSK, или QPSK ( $M = 4$ ), требуются одинаковые значения  $E_b/N_0$ . Иными словами, при том же значении  $E_b/N_0$  эффективность использования полосы частот для схемы QPSK равна 2 бит/с/Гц, в отличие от 1 бит/с/Гц для схемы BPSK. Эта уникальная особенность является следствием того, что QPSK представляет собой эффективную комбинацию двух сигналов в модуляции BPSK, которые передаются на ортогональных компонентах несущей.

На рисунке 4.6 также показаны рабочие точки для когерентной квадратурной амплитудной модуляции (quadrature amplitude modulation – MQAM). Видно, что на фоне остальных модуляций MQAM наиболее эффективно использует полосу частот из всех представленных на рисунке видов модуляции.

### **4.4 Оценка влияния замираний на модулированный сигнал**

В системах радиосвязи сигнал может передаваться от передатчика к приемнику по множеству путей. Это явление, называемое многолучевым распространением (multipath propagation), может вызвать флуктуации амплитуды, фазы и угла прибытия полученного сигнала, что определило название замирание вследствие многолучевого распространения. Если преобладает незамирающий компонент сигнала, такой путь как путь вдоль

распространения вдоль луча прямой видимости, огибающая замирания описывается функцией плотности вероятности Райса

$$p(r_0) = \begin{cases} \frac{r_0}{\sigma^2} \exp\left[-\frac{(r_0^2 + A^2)}{2\sigma^2}\right] I_0\left(\frac{r_0 A}{\sigma^2}\right) & \text{для } r_0 \geq 0, A \geq 0 \\ 0 & \text{для других } r_0, A \end{cases} \quad (4.8)$$

где  $r_0$  – амплитуда огибающей релеевского замирания,

$\sigma^2$  – средняя мощность многолучевого сигнала до обнаружения,

$A$  – максимальное значение незамирающего компонента сигнала, называемого зеркальным компонентом,

$I_0(f)$  – модифицированная функции Бесселя первого рода нулевого порядка.

Распределение Райса часто записывают через параметр  $K$ , который определяется как отношение мощности зеркального компонента к мощности многолучевого сигнала. Математически это записывается как

$$K = \frac{A^2}{2\sigma^2} \quad (4.9)$$

При приближении к нулю амплитуды зеркального компонента функция плотности вероятностей Райса стремится к функции плотности вероятности Релея, имеющей следующий вид

$$p(r_0) = \begin{cases} \frac{r_0}{\sigma^2} \exp\left[-\frac{r_0^2}{2\sigma^2}\right] & \text{для } r_0 \geq 0 \\ 0 & \text{для других } r_0 \end{cases} \quad (4.10)$$

Замирание называется Релеевским, если имеется большое число многократно отражающихся лучей, и нет компонента сигнала вдоль луча прямой видимости; огибающая такого сигнала статистически описывается с помощью релеевской функции плотности вероятности. Иными словами, огибающая сигнала распределена по закону Релею, если путь распространения вдоль луча обзора блокирован, в противном случае имеем распределение Райса.

Помимо параметра  $K$  замирания характеризуются задержкой распространения и доплеровским смещением частоты. Эффект Доплера – отклонение частоты, воспринимаемой приемником, от частоты, переданной передатчиком, которое возникает вследствие относительного движения передатчика и приемника. Если пренебречь эффектами второго и более высоких порядков, смещение частоты равно

$$\Delta f = \frac{Vf_0}{c} \quad (4.11)$$

где  $V$  – относительная скорость (положительная, если расстояние между приемником и передатчиком сокращается),

$f_0$  – номинальная частота передачи,

$c$  – скорость света.

Кроме того, в радиосвязи параметры каналов изменяются во времени, поскольку изменение градиента диэлектрической проницаемости (рефракция радиоволн) и движение передатчика и/или приемника приводят в результате к изменению пути распространения. Скорость изменения таких условий распространения определяет скорость замирания (скорость изменения ухудшения характеристик вследствие замирания).

Для того чтобы оценить негативное влияние Райсовских замираний на модулированный сигнал были проведены несколько экспериментов.

На рисунке 4.7 представлен график, характеризующий воздействие медленных замираний Райса на сигналы модулированные QAM и PSK.

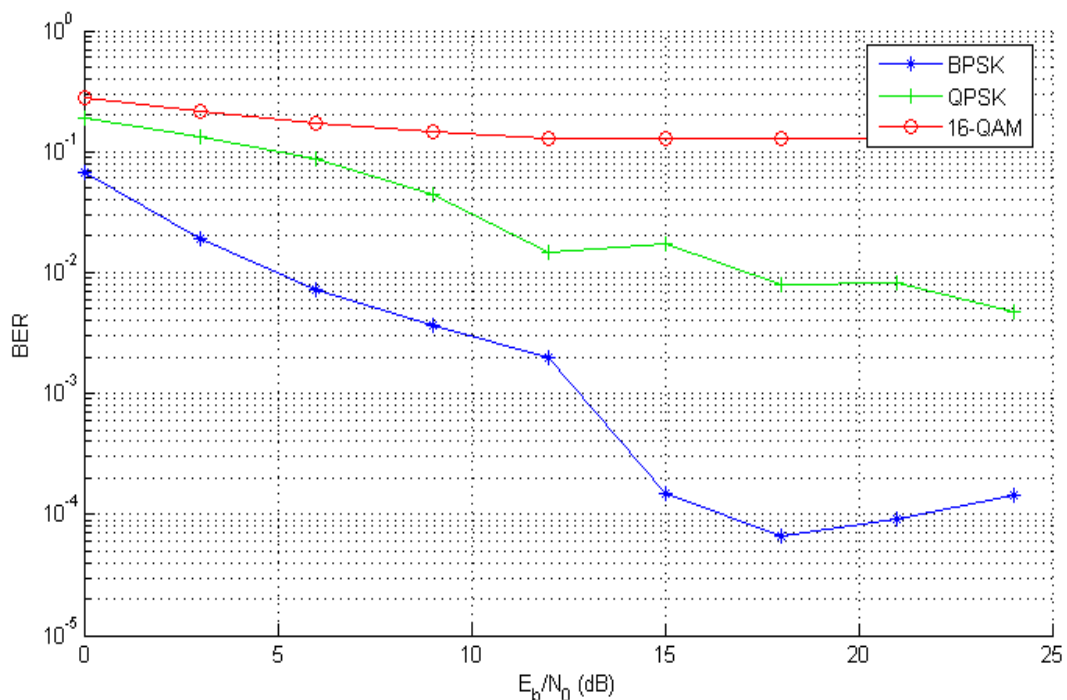


Рисунок 4.7 – Влияние замираний Райса на модулированный сигнал

Для наглядности, на рисунке представлен не самый худший случай замираний:

– величина доплеровского сдвига ( $ds$ ) для замираний была выбрана равной 4 Гц, что согласно формуле (4.11) эквивалентно движению со скоростью приблизительно равной 5 км/ч, если частоту передачи принять

равной 900 МГц (частота работы аппаратуры сотовой связи стандарта GSM-900);

- задержка распространения – 0,0001 секунды;
- $K = 5$ .

Из рисунка 4.7 видно, что при увеличении позиционности модуляции наблюдается резкое ухудшение помехоустойчивости в условиях сочетания Райсовских замираний и гауссовского шума. Вместе с этим, коэффициент ошибки не обязательно уменьшается при увеличении значения  $E_b/N_0$ . Это обусловлено неоднородным воздействием замираний на сигнал в разные моменты времени, в результате чего точки на векторной плоскости находятся в постоянном упорядоченном движении. Это явление характерно для медленных замираний. Движение точек на векторной плоскости иллюстрируют вектограммы, приведенные на рисунке 4.8.

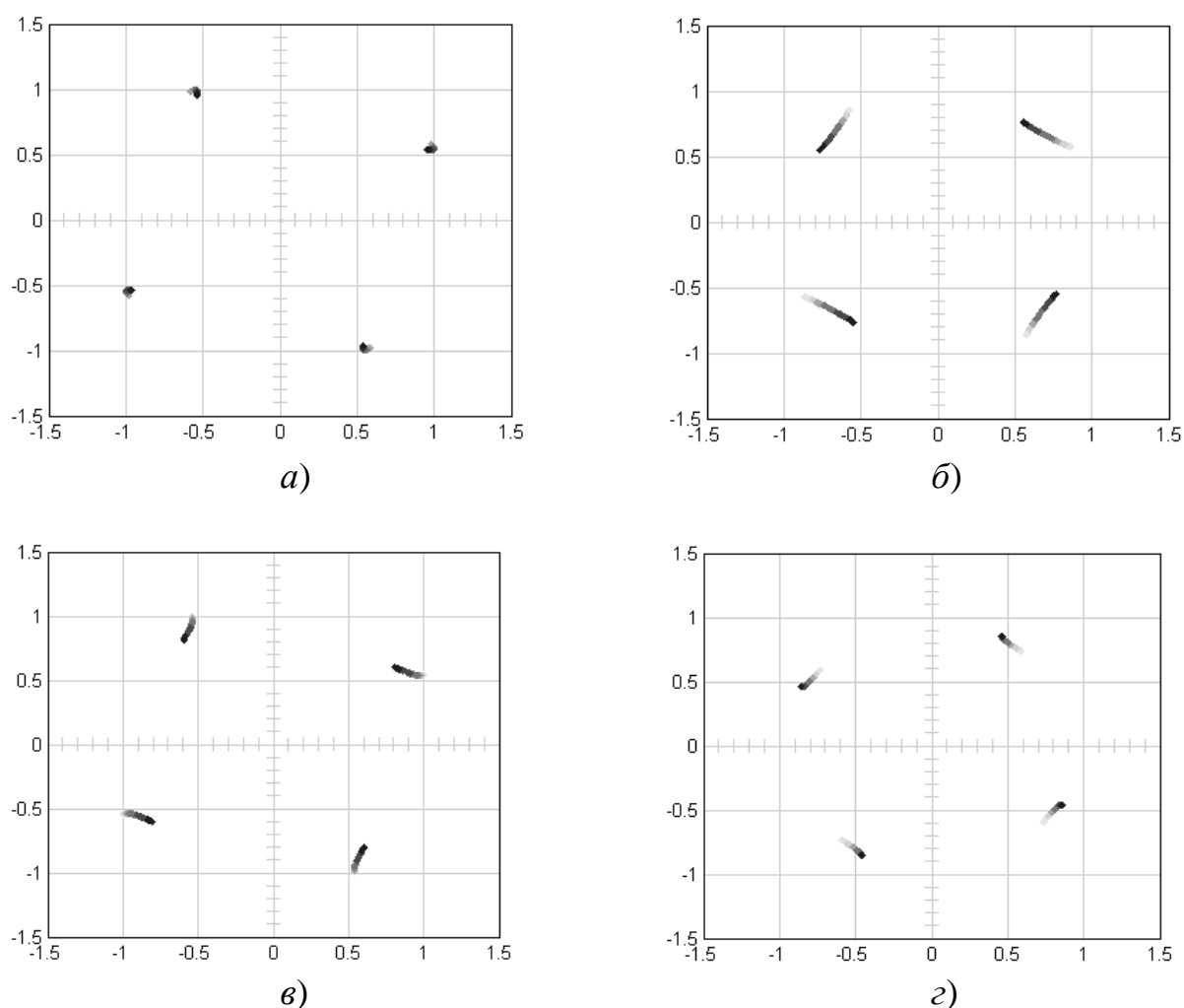


Рисунок 4.8 – Воздействие замираний сигнала по закону Райса на QPSK вектограмму.

Рисунки 4.8 *a-g* иллюстрируют движение точек созвездия по векторной плоскости под воздействием замираний Райса на временном промежутке в 10 секунд в различные моменты времени, при приведенных выше параметрах канала. Скорость и «глубина» этих перемещений зависит от относительной скорости движения приёмника и передатчика, а также от коэффициента  $K$ .

На рисунке 4.9 приведен результат совместного влияния на вектограмму QPSK замираний Райса и гауссовских помех.

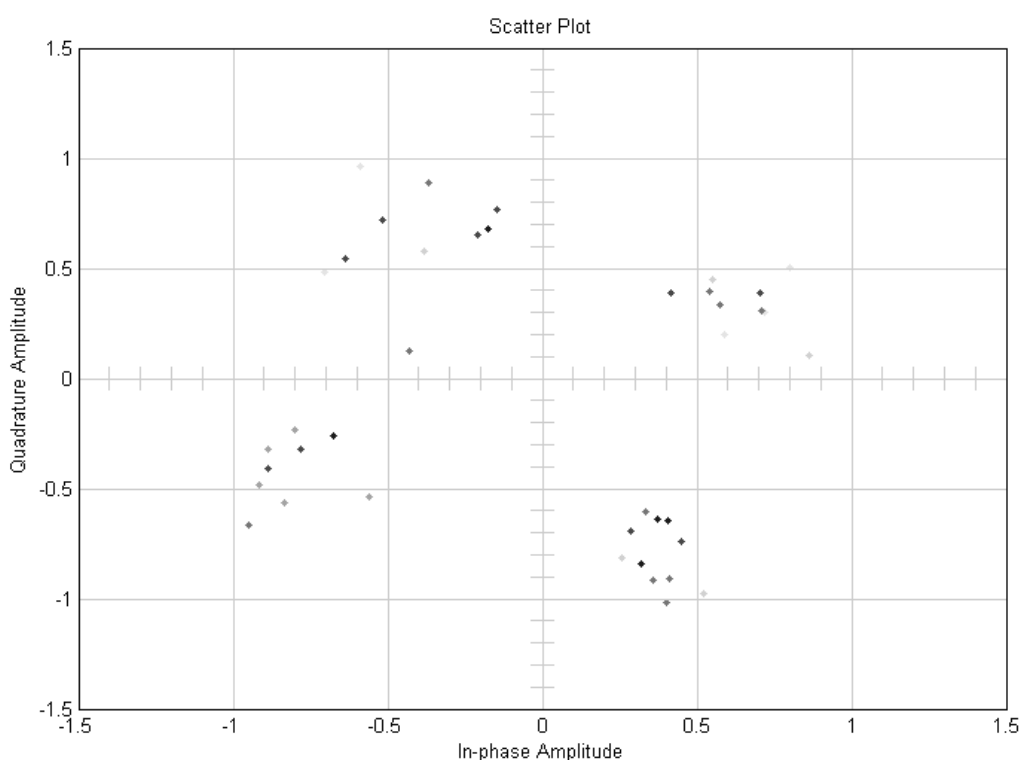


Рисунок 4.9 – Созвездие сигнала QPSK после воздействия Гауссовского шума и замираний Райса

При воздействии замираний Рэлея вероятность ошибки возрастает для всех видов модуляции до критического значения  $P_B = 0,5$ . Это иллюстрирует график, приведенный на рисунке 4.11. На нем показана характеристика модуляции BPSK; очевидно, что для остальных видов многофазной модуляции ситуация еще хуже, поскольку BPSK наиболее помехоустойчивая из них.



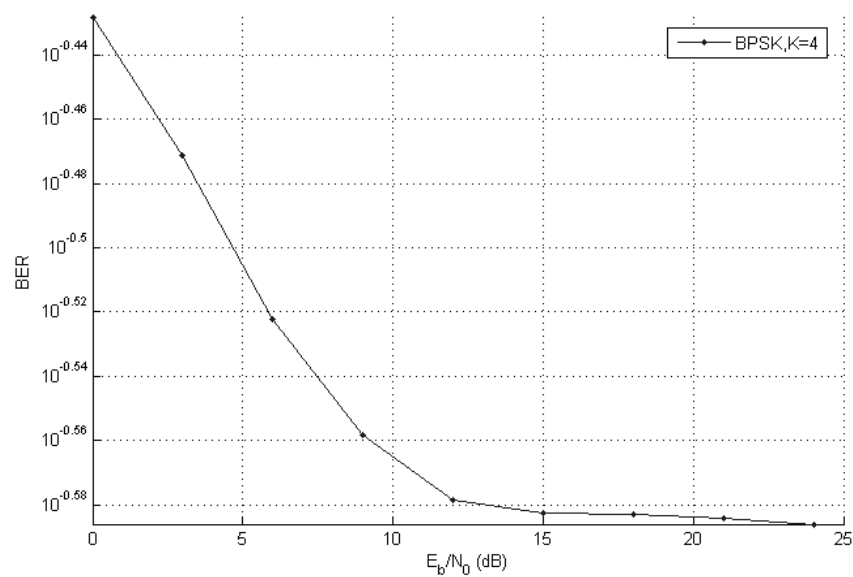


Рисунок 4.10 – Зависимость BER от  $E_b/N_0$  для BPSK в условиях Гауссовского шума и Рэлеевских замираний

## Контрольные вопросы

1. Пояните зависимость вероятности ошибки от отношения  $E_b/N_0$  при 4, 8 и 16 позиционной фазовой модуляции.
2. Назовите основные параметры, которые должны учитываться при разработке систем связи.
3. Пояните различие схем BPSK и DBPSK и сравните их помехоустойчивость.
4. Сравните помехоустойчивость систем M-PSK и M-QAM.
5. Пояните расчет полосы, занимаемой модулированным сигналом с M- позиционной модуляцией.
6. Пояните различие в эффективности использования полосы частот при M- позиционной модуляции с различными значениями ROF.
7. Пояните смысл влияния Райсовских замираний на помехоустойчивость приема сигналов с M- позиционной модуляцией.
8. Пояните смысл влияния Релеевских замираний на помехоустойчивость приема сигналов с M- позиционной модуляцией.

## **5 Исследование влияния способов кодирования сигнала на помехоустойчивость канала радиосвязи**

Важнейшей проблемой современных цифровых систем радиосвязи (ЦСРС) является проблема помехоустойчивости. Основными направлениями улучшения помехоустойчивости являются введение избыточного кодирования при использовании многопозиционных методов модуляции.

В данной работе проводится исследование влияния параметров ЦСРС на вероятность битовых ошибок.

Исследование базируется на блочной модели, реализованной с помощью программного пакета MATLAB. Пакет MATLAB является универсальной средой для моделирования любых систем, в том числе и телекоммуникационных. Метод блочного моделирования, позволяет не описывать математически всю модель целиком на языках программирования, а составить её, используя уже заложенные в программе SIMULINK (расширение программного пакета MATLAB) блоки, варьируя настройки которых, можно добиться нужных параметров системы.

В исследовании используется упрощенная модель ЦСРС, включающая в себя: генератор сигнала, кодер канала, модулятор, модель канала, демодулятор, декодер, а также ряд вспомогательных блоков.

С помощью генератора вырабатывается псевдослучайная последовательность битов, моделирующих цифровой сигнал. Этот сигнал поступает в кодер канала. Кодированный сигнал поступает на вход многопозиционного модулятора. С выхода модулятора сигнал поступает в радиоканал, в котором имитируются шумовые характеристики реального канала связи – белый гауссовский шум, замирания Релея-Райса. Далее выполняется обратная последовательность преобразований.

Для согласования блоков модели используются согласующие устройства (буферы, регистры, устройства памяти и т.д.). Для обеспечения корректной работы модели были выбраны соответствующие параметры – частота дискретизации, скорость передачи, размеры буферов. Наблюдения проводятся по осциллограммам, спектрограммам и вектограммам в различных точках модели. На приемном конце проводится подсчет ошибок и вычисление вероятности их появления.

На базе описанной выше модели проведены исследования влияния способов кодирования и модуляции сигнала на помехоустойчивость цифрового канала радиосвязи, при изменении его характеристик.

### **5.1 Спецификация различных типов кодеров**

Главная задача канального кодирования – улучшение рабочих характеристик (вероятности ошибки,  $E_b/N_0$  или пропускной способности) за

счет полосы пропускания. Канальное кодирование состоит из двух частей: кодирование формы сигнала и структурированные последовательности. Кодирование формы сигнала представляет собой преобразование сигналов в усовершенствованные сигналы, которые дают улучшенные пространственные характеристики (по сравнению с исходными сигналами). Структурированные последовательности подразумевают добавление к данным избыточных разрядов, что позволяет обнаруживать и/или исправлять определенные модели ошибки.

Для двоичных блочных кодов была исследована исправляющая способность циклических кодов БЧХ, которые сравнительно легко реализуются с помощью современных технологий интегральных схем. В качестве двоичного блочного кодера был использован кодер БЧХ (BCH Encoder).

Коды Рида-Соломона – важный класс недвоичных блочных кодов, специально применяемых для коррекции пакетных ошибок. Коды Рида-Соломона особенно привлекательны, поскольку эффективность кода растет с его длиной. При большой длине блока коды можно сконфигурировать таким образом, что время декодирования будет значительно меньше, чем у других кодов с той же длиной блока. Это связано с тем, что декодер работает с целыми символами, а не битами. Следовательно, для 8-битовых символов арифметические операции будут выполняться на уровне байтов. По сравнению с двоичными кодами той же длины это повышает не только сложность логики, но и производительность. В качестве кодера Рида-Соломона использовалась модель устройства Binary RS Encoder.

В течение последних десяти лет наиболее популярной схемой кодирования являлась сверточная, поскольку почти во всех приложениях сверточные коды лучше блочных при той же конструктивной сложности кодера и декодера. Для каналов спутниковой связи схемы прямого исправления ошибок позволяют легко понизить на 5-6 дБ требуемое значение SNR для заданной достоверности передачи. Из этой эффективности кодирования непосредственно вытекает снижение эффективной изотропной излучаемой мощности спутника (effective isotropic radiated power – EIRP), что, соответственно, приводит к снижению веса и стоимости спутника. В качестве сверточного кодера использовалась модель устройства Convolutional Encoder с декодером Витерби (Viterbi Decoder).

Методика чередования (перемежения) позволяет, без потерь в качестве, использовать большинство блочных и сверточных схем кодирования в каналах с импульсными помехами или периодическим замиранием. В данном исследовании была проанализирована работа двух классов перемежителей: блочных и сверточных. В качестве блочных перемежителей были использованы следующие устройства: матричный диагональный перемежитель (Matrix Helical Scan Interleaver), псевдослучайный

перемежитель (Random Interleaver), а так же сверточный перемежитель (Convolutional Interleaver).

## 5.2 Результаты исследования

С помощью специального инструмента анализа *BERTool* программного пакета MATLAB были получены графики зависимости коэффициента битовых ошибок BER от отношения сигнал/шум при различных параметрах радиотракта.

Основные параметры системы:

Скорость информационных бит

100 Мбит/с;

Вид модуляции

QAM-128;

Скорость кода:

BCH

64/127;

RS

(63/127)x7;

Convolutional

1/2;

Глубина перемежения

1000 бит.

1. Исследование влияния различных способов кодирования на коэффициент битовых ошибок (*BER*) при воздействии аддитивного белого гауссовского шума, (рисунок 5.1).

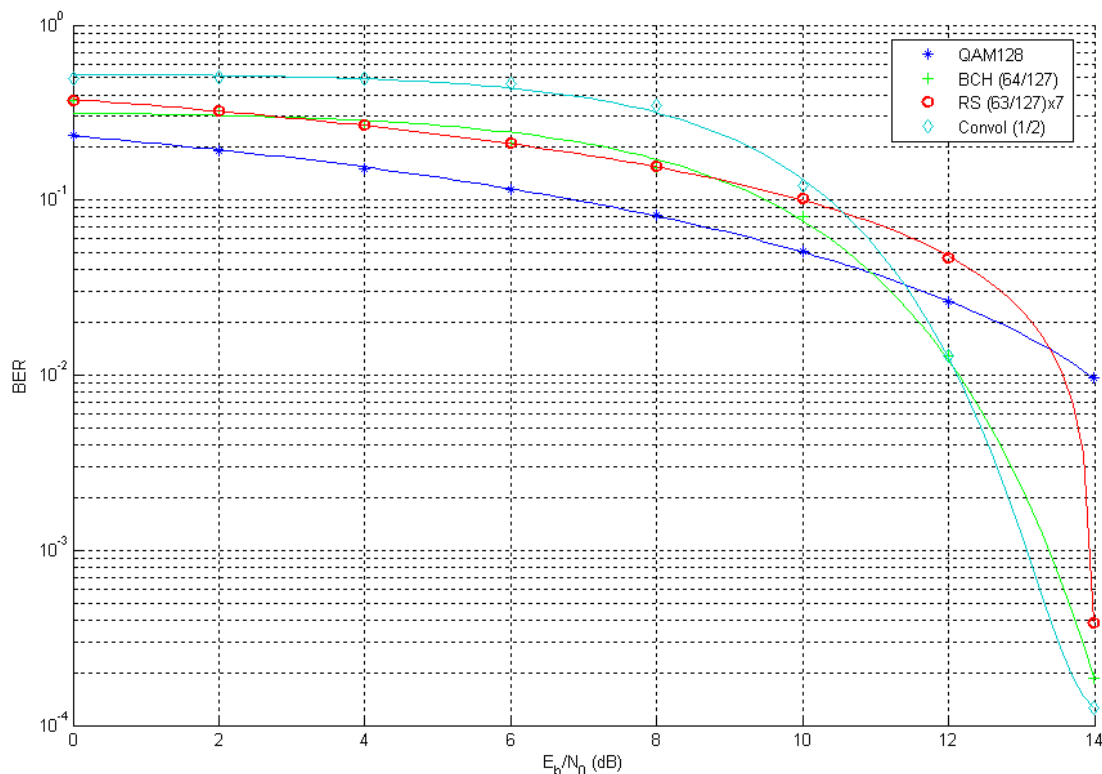


Рисунок 5.1 – Зависимость BER от отношения  $E_b/N_0$  при воздействии АБГШ, при использовании различных видов избыточного кодирования.

Анализ: Распределение ошибок носит случайный характер. Они не группируются и, следовательно, декодерам легче производить их коррекцию. Сверточное кодирование дает заметный выигрыш по коэффициенту BER при большом отношении сигнал/шум. Хотя при малой энергии сигнала декодеры начинают размножать ошибки (особенно сверточный код) и коэффициент BER становится даже больше, чем без использования кодирования.

2. Исследование влияния различных способов кодирования на коэффициент битовых ошибок (BER) при воздействии Аддитивного белого гауссовского шума и Райсовских замираний ( $K = 40$ ), (рисунок 5.2).

Основные параметры среды распространения:

$E_b/N_0$  [0..14] дБ;

$K$  40 ;

Максимальный Доплеровский сдвиг 5 Гц.

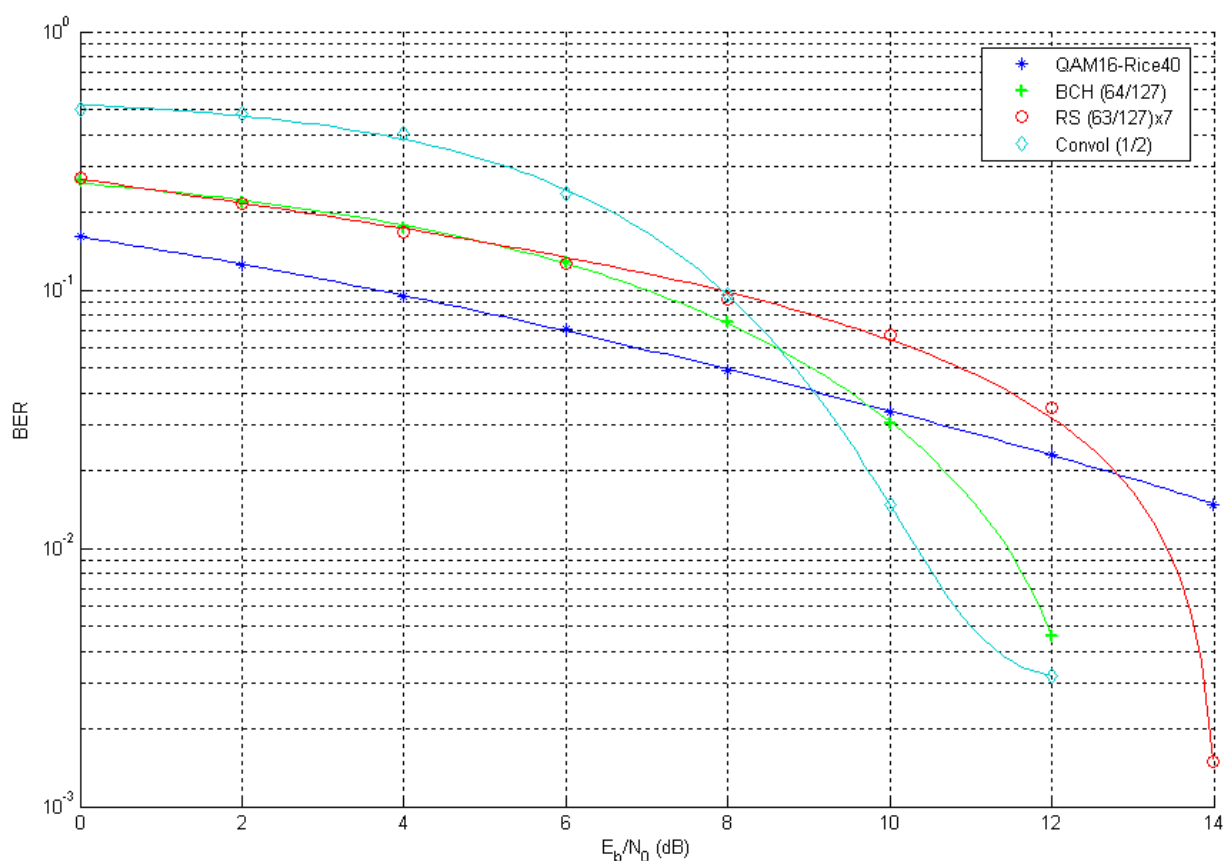


Рисунок 5.2 – Зависимость BER от отношения  $E_b/N_0$  при воздействии АБГШ и Райсовских замираний ( $K = 40$ )

Анализ: Имеет место некоторое группирование ошибок. Отношение энергии сигнала, проходящего в прямом направлении к энергии всех отраженных сигналов ( $K$ ) велико. Перемежителю легко удастся

рассредоточить ошибки во времени. Поэтому коды ведут себя аналогично ситуации при воздействии только АБГШ.

3. Исследование влияния различных способов кодирования на коэффициент битовых ошибок (BER) при воздействии аддитивного белого гауссовского шума и Райсовских замираний ( $K = 20$ ), (рисунок 5.3).

Основные параметры среды распространения:

$E_b/N_0$	[0..14] дБ;
$K$	20 ;
Максимальный Доплеровский сдвиг	5 Гц.

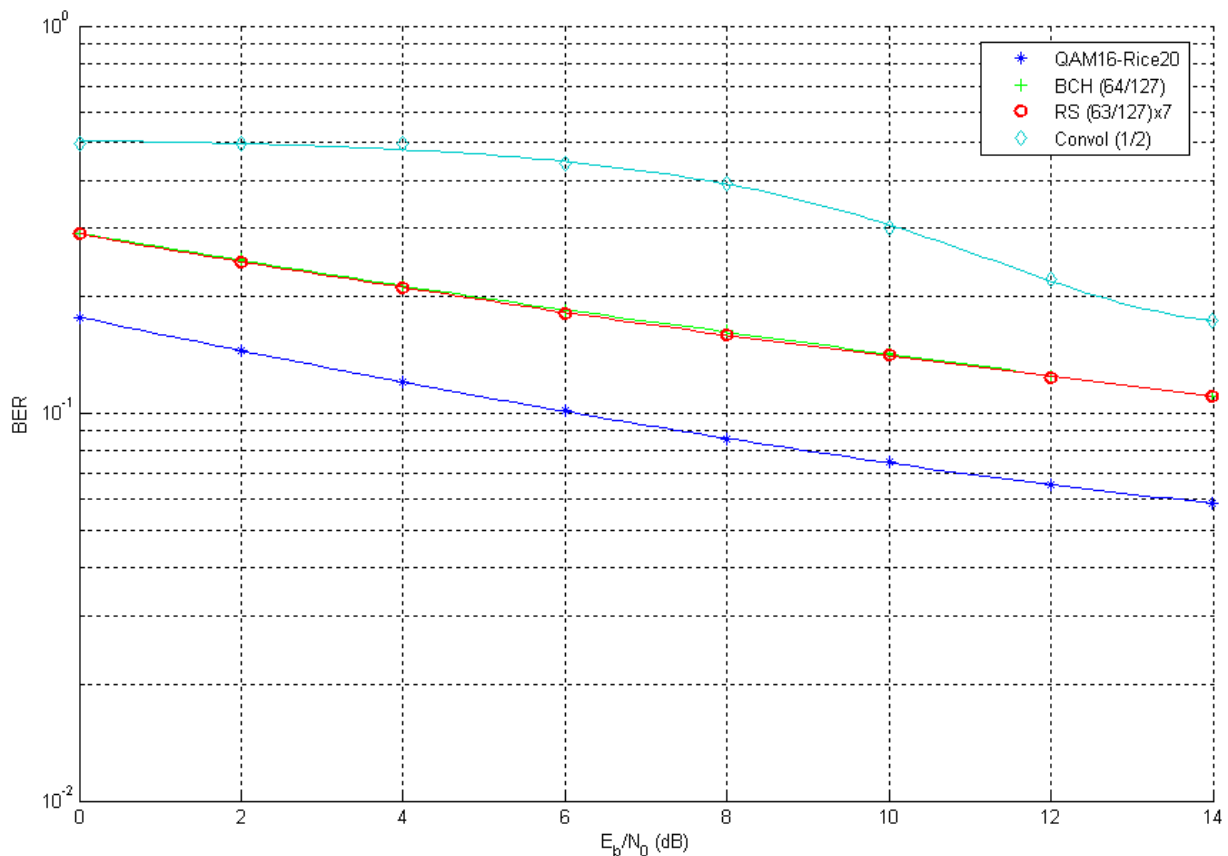


Рисунок 5.3 – Зависимость BER от отношения  $E_b/N_0$  при воздействии АБГШ и Райсовских замираний ( $K = 20$ ) )

Анализ: Очень низкий коэффициент  $K$  не позволяет перемежителю разгруппировать ошибки. Плотность пакетов ошибок настолько велика, что декодеры теряют свою исправляющую способность и начинают активно размножать ошибки.

4. Исследование влияния различных способов кодирования на коэффициент битовых ошибок (BER) при воздействии аддитивного белого гауссовского шума и Релеевских замираний, (рисунок 5.4).

Основные параметры среды распространения:

$E_s/N_0$

[0..14] дБ;

Максимальный Доплеровский сдвиг

1 кГц.

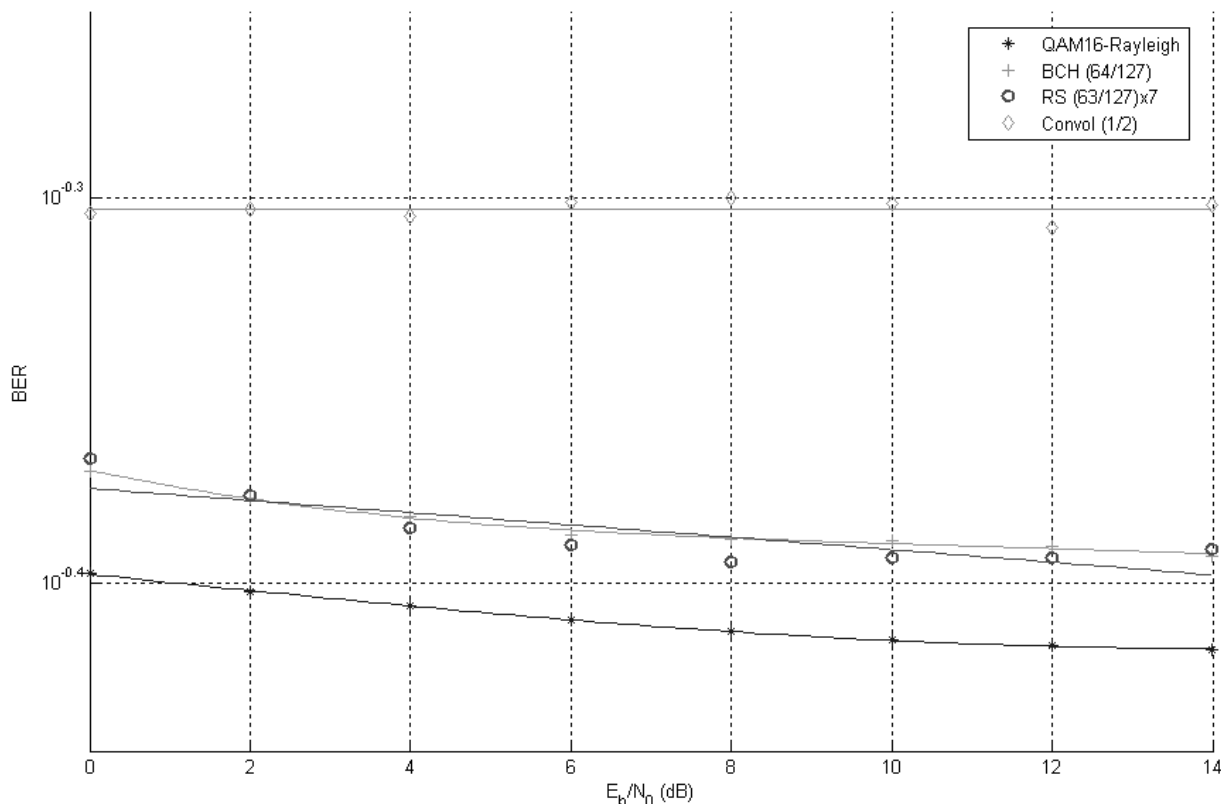


Рисунок 5.4 – Зависимость BER от отношения  $E_b/N_0$  при воздействии АБГШ и Релеевских замираний

Анализ: Релеевские замирания являются наиболее тяжелым случаем как для демодуляции так и для декодирования. Большие пакеты ошибок с выхода модулятора не позволяют исправлять ошибки в кодовых комбинациях даже с использованием большой глубины перемежения. Самым неэффективным кодом в данной ситуации является сверточный код. Это происходит вследствие его главной особенности – сверточные коды со степенью кодирования  $1/n$  сохраняют в памяти предыдущие  $K - 1$  бит, где  $K$  означает длину кодового ограничения. С такой памятью кодирование каждого входного бита данных зависит не только от значения этого бита, но и от предшествующих ему  $K - 1$  бит.



По результатам исследования были получены графики зависимости коэффициента битовых ошибок BER от отношения  $E_b/N_0$  при различных параметрах радиотракта.

Наиболее эффективным кодом при работе с каналом с аддитивным белым гауссовским шумом является сверточный код. На сегодняшний день сверточное кодирование с обратной связью является наиболее широко применяемым в системах с коррекцией ошибок. Циклический код находит применение в системах с обнаружением ошибок, вследствие его превосходной обнаруживающей способности.

При работе с радиотрактами, где имеют место замирания, используются специальные методы эквалайзинга и перемежения большой глубины совместно с блочным и сверточным кодированием.

## Контрольные вопросы

1. Назначение генератора псевдослучайной последовательности в рассматриваемых моделях канала.
2. Чем отличаются блочные и сверточные коды.
3. Назначение перемежения символов в каналах связи.
4. Поясните зависимость вероятности ошибки от отношения  $E_b/N_0$  в канале с АБГШ для различных кодов.
5. Поясните зависимость вероятности ошибки от отношения  $E_b/N_0$  в канале с АБГШ и Райсовских замираний для различных кодов.
6. Поясните зависимость вероятности ошибки от отношения  $E_b/N_0$  в канале с АБГШ и Релеевских замираний для различных кодов.

## 6 Задание и выполнение работы

### 6.1 Задание на работу

Заданы модели систем связи с:

- битовой скоростью передачи  $R_b$ , Мбит/с;
- модуляцией 4, 8 PSK, 16, 64, 256 QAM;
- фильтрами с коэффициентом сглаживания ROF;
- каналом с шумом с отношением  $E_b/N_0$ , dB.

Таблица 6.1 – Варианты заданий на работу

№ вар	Скорость $R_b$ , Мбит/с	Модуляция		ROF	
1	0.1	4 PSK	16-QAM	0.2	0.55
2	1.8	8 PSK	64-QAM	1.0	0.5
3	0.2	4 PSK	64-QAM	0.85	0.35
4	0.7	4 PSK	256-QAM	0.5	0.9
5	0.4	8 PSK	16-QAM	0.35	0.75
6	1.5	8 PSK	16-QAM	0.9	0.4
7	0.6	16-QAM	64-QAM	0.45	0.95
8	1.3	16-QAM	256-QAM	0.6	1.0
9	0.8	16-QAM	8 PSK	0.55	0.95
10	1.7	16-QAM	4 PSK	1.0	0.3
11	1.0	64-QAM	4 PSK	0.65	0.25
12	0.3	64-QAM	8 PSK	0.3	0.6
13	1.2	64-QAM	16-QAM	0.75	0.25
14	0.9	64-QAM	256-QAM	0.6	0.2
15	1.4	256-QAM	64-QAM	0.35	0.95
16	0.5	256-QAM	16-QAM	0.4	0.7
17	1.6	256-QAM	8 PSK	0.95	0.2
18	1.1	256-QAM	4 PSK	0.5	1.0
19	0.1	4 PSK	16-QAM	0.2	0.55
20	1.8	8 PSK	64-QAM	1.0	0.5
21	0.2	4 PSK	64-QAM	0.85	0.35
22	0.7	4 PSK	256-QAM	0.5	0.9
23	0.4	8 PSK	16-QAM	0.35	0.75
24	1.5	8 PSK	16-QAM	0.9	0.4
25	0.6	16-QAM	64-QAM	0.45	0.95
26	1.3	16-QAM	256-QAM	0.6	1.0
27	0.8	16-QAM	8 PSK	0.55	0.95
28	1.7	16-QAM	4 PSK	1.0	0.3
29	1.0	64-QAM	4 PSK	0.65	0.25
30	0.3	64-QAM	8 PSK	0.3	0.6
31	1.2	64-QAM	16-QAM	0.75	0.25
32	0.9	64-QAM	256-QAM	0.6	0.2

## 6.2 Выполнение работы

Перед выполнением работы необходимо изучить теоретические сведения в размере соответствующих глав учебного пособия по данному курсу. Основой для выполнения контрольной работы являются модели, выполненные в MATLAB R2018b SIMULINK, представленные в разделе Модели.

Для выполнения работы на компьютере должен быть установлен MATLAB R2018b. В начале работы следует запустить основное окно программы MATLAB R2018b (рисунок 6.1), щелкнув на его ярлыке.

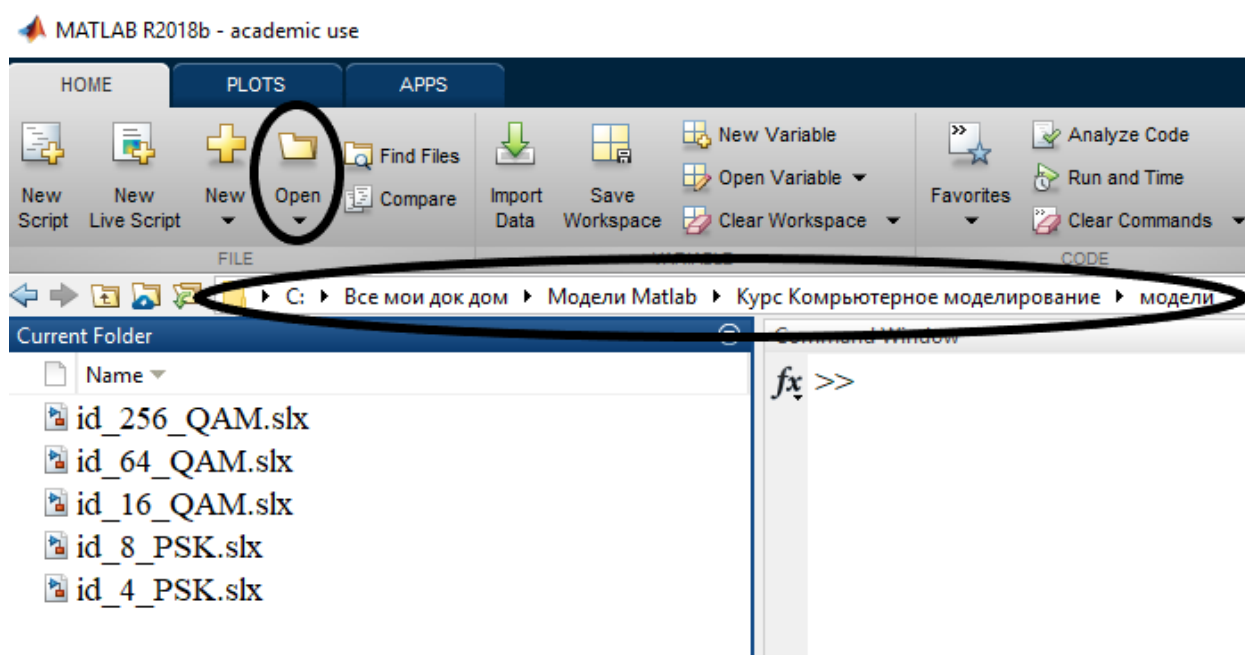


Рисунок 6.1 – Основное окно программы MATLAB R2018b

Затем необходимо открыть нужные файлы, например, “id\_4\_PSK” “id\_16\_ QAM ”, пользуясь либо кнопкой Open, либо проводником в левой части экрана, указав путь к каталогу Модели.

После открытия нужной модели, её нужно скопировать, нажав на клавиатуре клавишу Prt Sc. Далее открыть графический редактор, например, Paint. В открывшемся окне редактора правой кнопкой мыши (ПКМ) вставить содержимое экрана в окно. Затем выбирается нужная часть экрана и копируется и вставляется в документ Word. Таким образом, создаётся Screen shot экрана.

Screen shot модели, необходимой для выполнения рассматриваемого варианта контрольной работы, представлен на рисунке 6.2. Аннотация в нижней части рисунка создаётся двойным щелчком ЛКМ.

Модель, представленная на рисунке 6.2, собрана из совокупности элементов, образующих упрощенную модель системы радиосвязи. Описание элементов модели приведено в лекционном материале по курсу.

В модели в соответствии с рассматриваемым вариантом задания *должны быть установлены*: скорость в Bernoulli binary generator – Sample time =  $1/(R, \text{ bit/s})$ ; Rolloff factor в Raised Cosine Transmit and Receive Filter;  $E_b/N_0$  и Symbol period =  $1/(R/\log_2 M)$  в AWGN Channel.

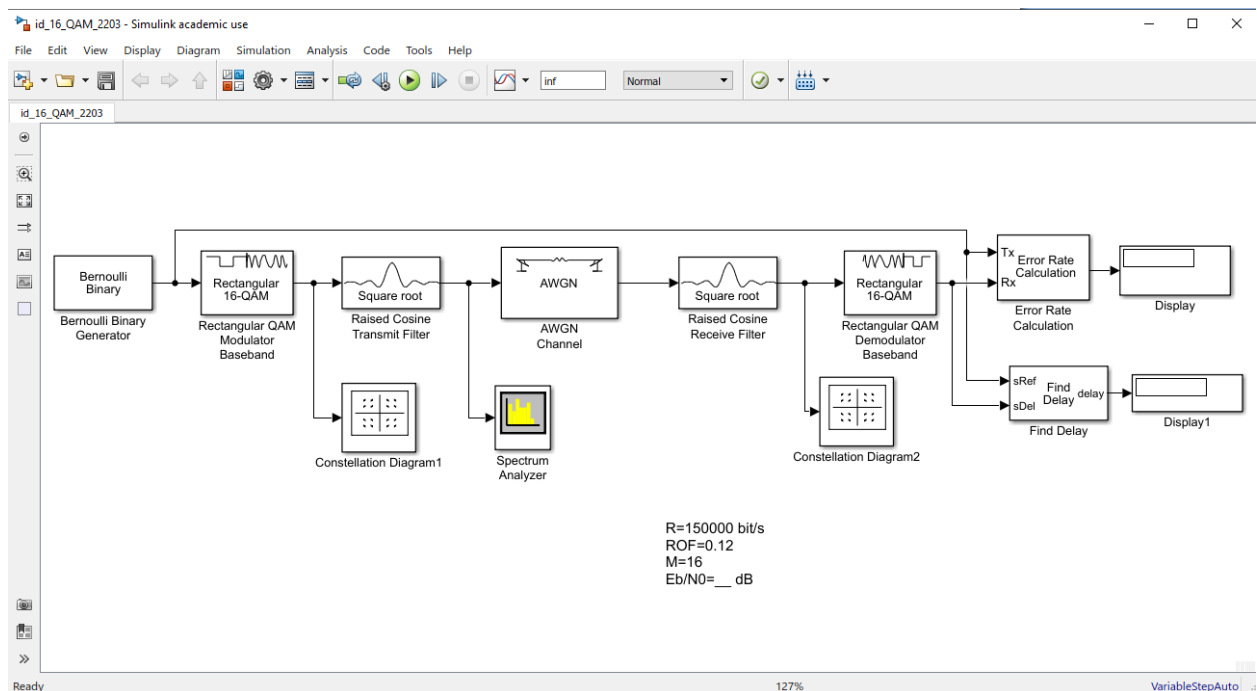




Рисунок 6.2 – Модель системы связи с модуляцией 16- QAM на панели Simulink

Цифровой сигнал, генерируемый блоком Bernoulli binary generator с заданной скоростью, поступает на вход фазового модулятора QPSK Modulator Baseband. Далее осуществляется ограничение спектра модулированного сигнала в Raised Cosine Transmit Filter. После фильтрации модулированный сигнал поступает в канал AWGN Channel, где на него накладывается аддитивный белый гауссовский шум. Уровень шума в канале регулируется параметром  $E_b/N_0$ . После прохождения канала сигнал подается на вход Raised Cosine Receive Filter. После фильтра сигнал подаётся на вход

фазового демодулятора QPSK Demodulator Baseband, где происходит обратное преобразование аналогового модулированного сигнала в цифровой; после чего производится подсчет вероятности ошибок в Error Rate Calculation.

Для наблюдения за сигналом в различных точках модели используются блоки графического отображения: дисплей Display, спектроскоп Spectrum Analyzer и векторграф Constellation Diagram.

### 6.2.1 Основные сведения для работы с моделью

- Запуск и остановка работы модели осуществляется кнопками Run  и Stop , расположенными на рабочей панели Simulink.
- Изменение скорости передачи производится в блоке Bernoulli binary generator, после открытия двойным щелчком ЛКМ на блоке модели, параметров блока Block Parameters в строке Sample time – 1/R.
- Изменение величины ROF производится в блоках Raised Cosine Transmit Filter и Raised Cosine Receive Filter в параметрах блока Block Parameters в строке – Rolloff factor.
- Изменение отношения сигнал/шум производится в блоке AWGN Channel в параметрах блока Block Parameters в строке –  $E_b/N_0$ .
- Окна вектограммы и спектрограммы появляются на экране автоматически после запуска модели. Их также можно открыть двойным щелчком ЛКМ в нужном блоке.

### 6.2.2 Порядок выполнения работы

1 Изучить структурные схемы моделей, пояснить назначение элементов схемы. Ознакомиться с основными сведениями по работе с моделью.

2 Пронаблюдать и привести скриншоты вектограмм (с заполненной легендой) на выходе модулятора для заданных видов модуляции. По вектограммам определить и записать расстояние между соседними точками созвездий. По полученным расстояниям сделать вывод, какая из двух систем более помехоустойчивая. Пояснить полученные результаты.

3 Изучить влияние позиционности модуляции на полосу частот, занимаемую модулированным сигналом.

3.1 При одинаковых значениях Rolloff factor (любом из двух заданных) по спектрограммам (привести скриншоты с курсорами, установленными на уровне минус 20 дБ и с заполненной легендой) для двух видов модуляции определить и записать полосы частот, занимаемые модулированными сигналами.

3.2 Сравнить их с полосами частот, определёнными по теоретической формуле  $\Delta = \frac{R(1+ROF)}{\log_2 M}$ . Пояснить полученные результаты.

4 Изучить влияние величины Rolloff factor (ROF) на полосу частот, занимаемую модулированным сигналом.

4.1 Для одного из двух заданных видов модуляции при двух заданных значениях Rolloff factor по спектрограммам (привести скриншоты с курсорами, установленными на уровне минус 20 дБ и с заполненной легендой) определить и записать полосы частот, занимаемые модулированными сигналами.

4.2 Сравнить их с полосами частот, определёнными по теоретической формуле  $\Delta = \frac{R(1+ROF)}{\log_2 M}$ . Пояснить полученные результаты.

5 Изучить влияние аддитивного белого Гауссовского шума (AWGN) на параметры вектограмм. Вектограммы наблюдаются после блока AWGN Channel на выходе Raised Cosine Receive Filter.

5.1 Установите в блоке AWGN Channel  $E_b/N_0 = 10$  дБ и для двух заданных видов модуляции приведите скриншоты вектограмм с заполненной легендой. Для этих вектограмм запишите значения Avg MER (dB).

5.2 Измените установленный уровень шума в канале  $E_b/N_0 = 20$  дБ и для двух заданных видов модуляции приведите скриншоты вектограмм с заполненной легендой. Для этих вектограмм запишите значения Avg MER (dB).

5.3 Сравните выполненные измерения и поясните полученные результаты.

6 Построение зависимостей коэффициента ошибок (BER) от отношения  $E_b/N_0$ .

6.1 Для определения исследуемого диапазона отношения  $E_b/N_0$  откройте специализированный графический пользовательский интерфейс Bit Error Rate Analysis и постройте графики для двух заданных видов модуляции, в режиме Theoretical путём нажатия кнопки Plot (рисунок 3.9).

6.2 По полученным теоретическим графикам определить для значений  $BER = 10^{-2}$ ,  $10^{-4}$  и  $10^{-6}$  четыре значения  $E_b/N_0$  для каждого вида модуляции.

6.3 Откройте окно модели для первого вида модуляции, установите одно из заданных значений ROF и первое из определённых выше значение  $E_b/N_0$ , заполните аннотацию на модели и запустите симуляцию. Симуляцию остановить при общем количестве бит не менее  $10^7$  (нижнее окно дисплея на выходе Error Rate Calculation). После окончания симуляции сделать скриншот модели.

6.4 Прodelайте п. 6.3 для остальных трёх значений  $E_b/N_0$  (скриншот модели не делать).

6.5 Откройте окно модели для второго заданного вида модуляции, установите одно из заданных значений ROF и первое из определённых выше значение  $E_b/N_0$ , заполните аннотацию на модели и запустите симуляцию. Далее проделайте тоже, что и в п.п. 6.3 и 6.4.

6.6 По четырём точкам постройте графики зависимости BER от  $E_b/N_0$ , сравните их с теоретическими (п.6.1) и поясните полученные результаты.

### 6.3 Пример выполнения работы

Таблица 6.1 – Вариант задания

№ вар	Скорость $R_b$ , Мбит/с	Модуляция		ROF	
1	0.16	4 PSK	16-QAM	0.27	0.68

Выполнение заданного варианта работы проводится в соответствии с подразделом 6.2.2 порядок выполнения работы.

2 Пронаблюдать и привести скриншоты вектограмм (с заполненной легендой) на выходе модулятора для заданных видов модуляции. По вектограммам определить и записать расстояние между соседними точками созвездий. По полученным расстояниям сделать вывод, какая из двух систем более помехоустойчивая. Пояснить полученные результаты.



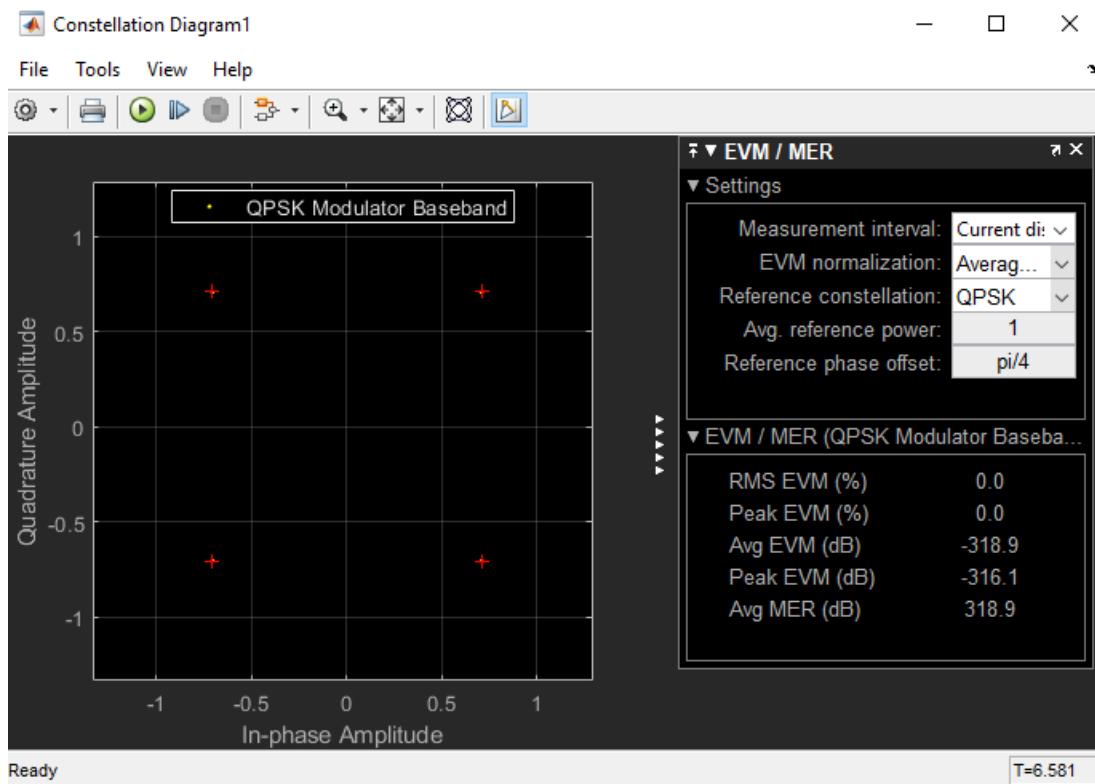


Рисунок 6.3 – Вектограмма на выходе модулятора QPSK (4-PSK)

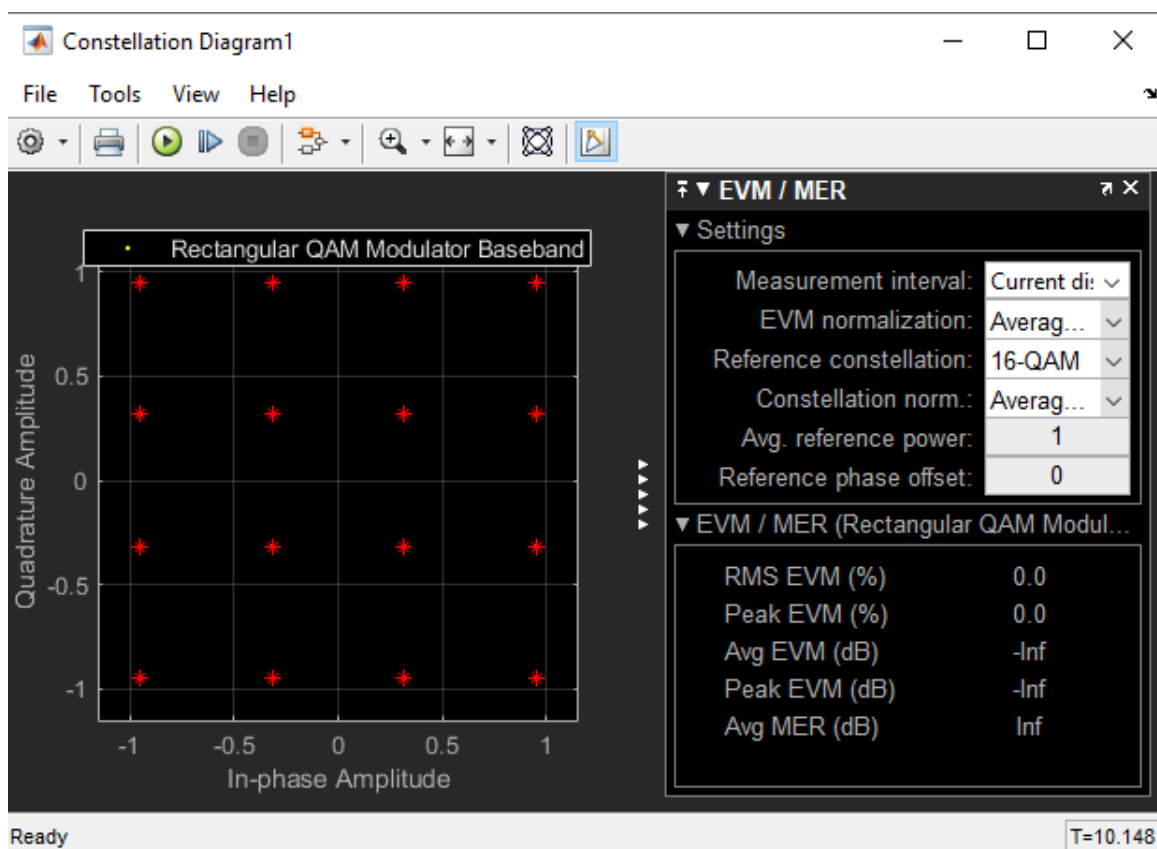


Рисунок 6.4 – Вектограмма на выходе модулятора 16-QAM

3 Изучить влияние позиционности модуляции на полосу частот, занимаемую модулированным сигналом.

3.1 При одинаковых значениях Rolloff factor (любом из двух заданных) по спектрограммам (привести скриншоты с курсорами, установленными на уровне минус 10 дБ и с заполненной легендой) для двух видов модуляции определить и записать полосы частот, занимаемые модулированными сигналами.

3.2 Сравнить их с полосами частот, определёнными по теоретической формуле  $\Delta = \frac{R(1+ROF)}{\log_2 M}$ . Пояснить полученные результаты.

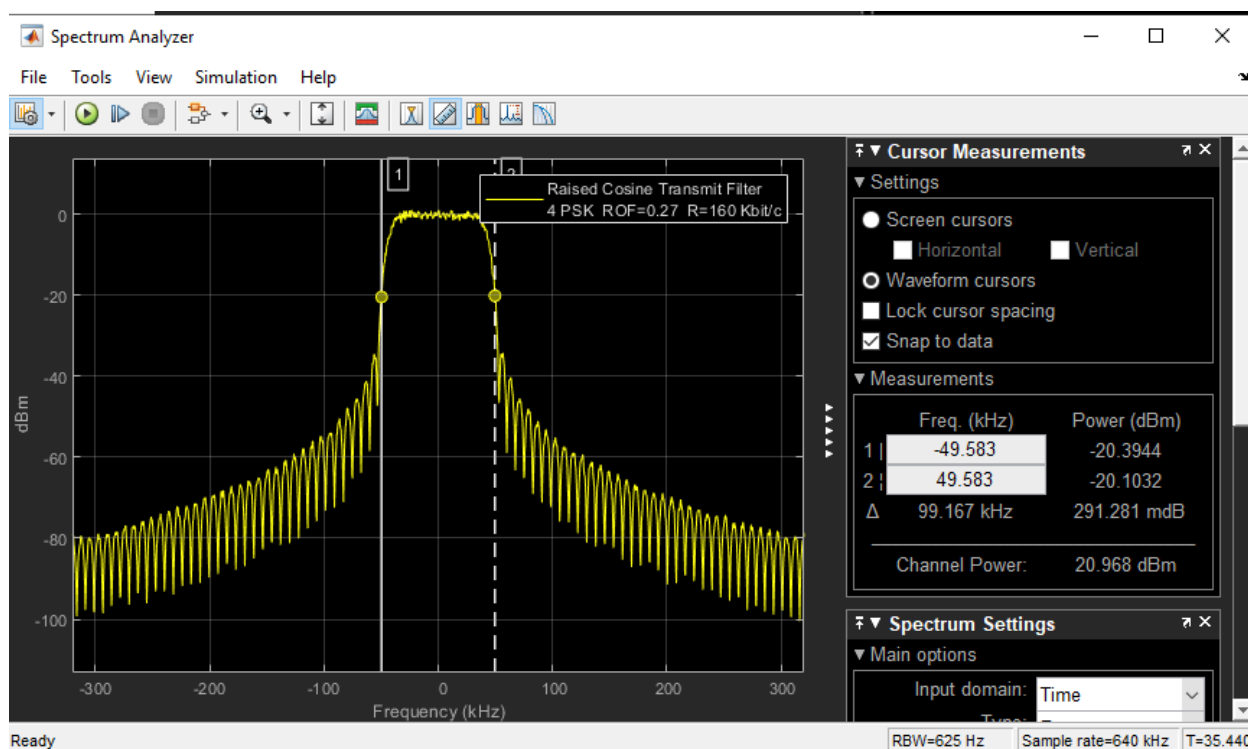


Рисунок 6.5 – Спектрограмма модулированного сигнала для 4-PSK

4 Изучить влияние величины Rolloff factor (ROF) на полосу частот, занимаемую модулированным сигналом.

4.1 Для одного из двух заданных видов модуляции при двух заданных значениях Rolloff factor по спектрограммам (привести скриншоты с курсорами, установленными на уровне минус 10 дБ и с заполненной легендой) определить и записать полосы частот, занимаемые модулированными сигналами.

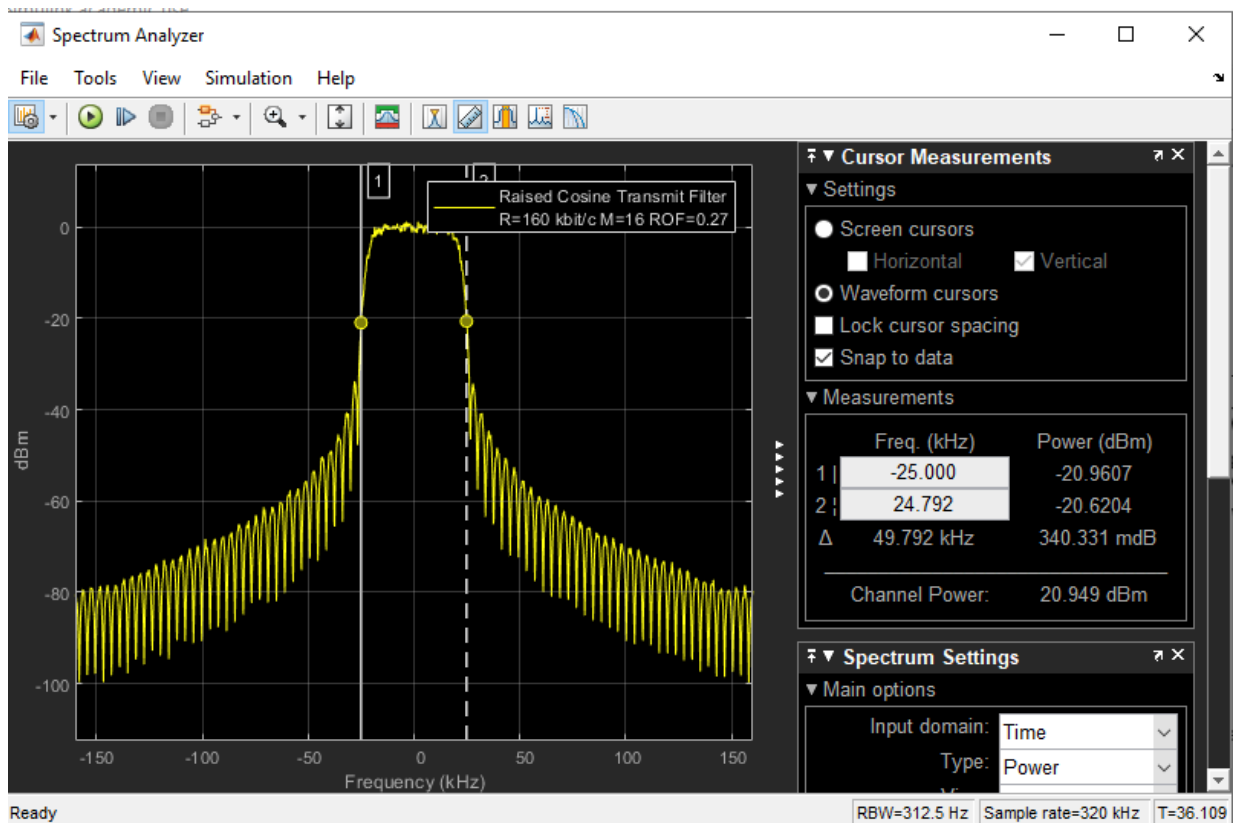


Рисунок 6.6 – Спектрограмма модулированного сигнала для 16-QAM

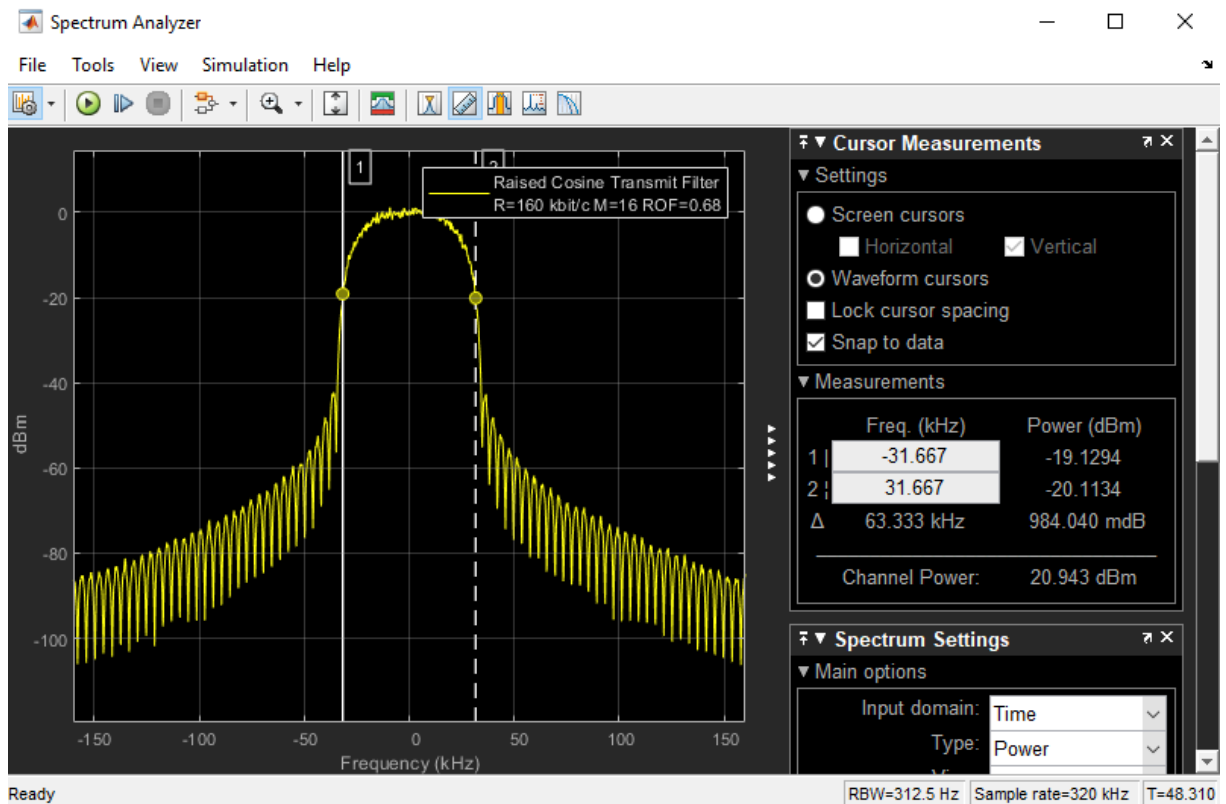


Рисунок 6.7 – Спектрограмма модулированного сигнала для 16-QAM

4.2 Сравнить их с полосами частот, определёнными по теоретической формуле  $\Delta = \frac{R(1+ROF)}{\log_2 M}$ . Пояснить полученные результаты.

Для  $ROF = 0.27$  использовать рисунок 6.6.

5 Изучить влияние аддитивного белого Гауссовского шума (AWGN) на параметры вектограмм. Вектограммы наблюдаются после блока AWGN Channel на выходе Raised Cosine Receive Filter.

5.1 Установите в блоке AWGN Channel  $E_b/N_0 = 10$  дБ и для двух заданных видов модуляции приведите скриншоты вектограмм с заполненной легендой. Для этих вектограмм запишите значения Avg MER (dB).

5.2 Измените установленный уровень шума в канале  $E_b/N_0 = 20$  дБ и для двух заданных видов модуляции приведите скриншоты вектограмм с заполненной легендой. Для этих вектограмм запишите значения Avg MER (dB).

5.3 Сравните выполненные измерения и поясните полученные результаты.

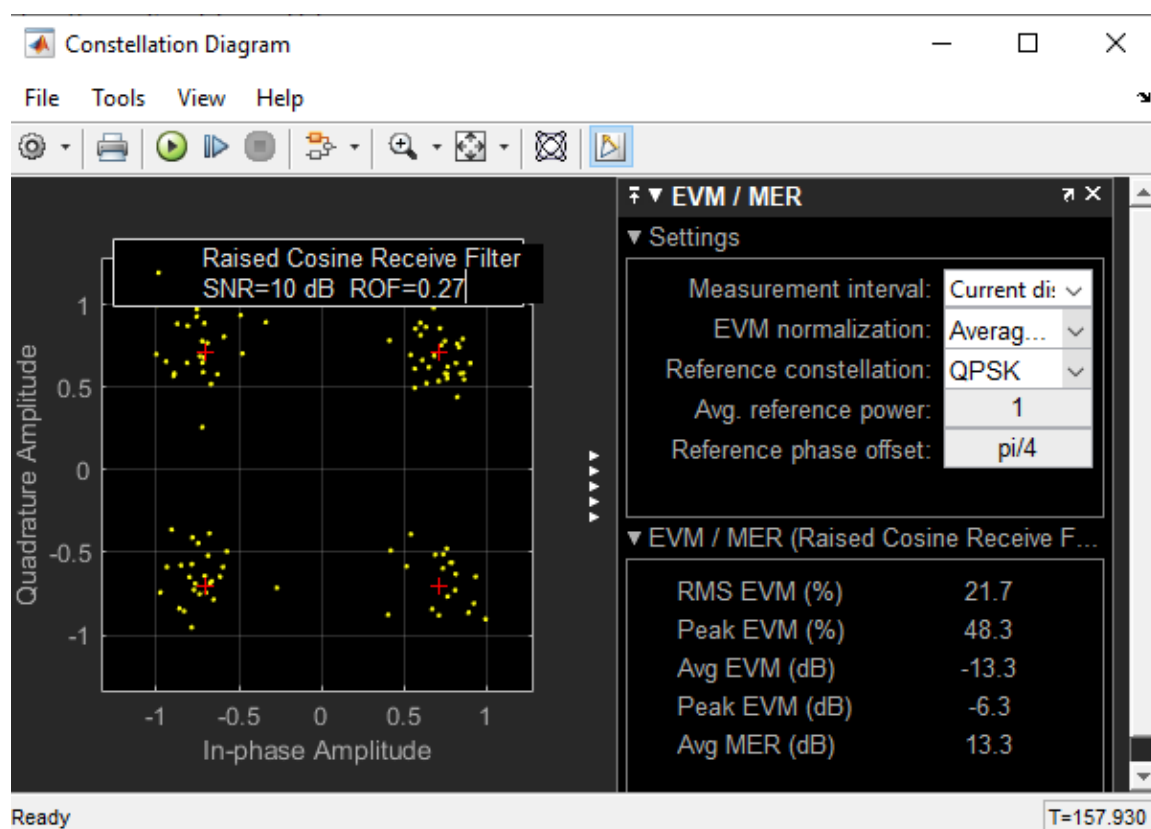


Рисунок 6.8 – Вектограмма модулированного сигнала для 4-PSK при  $E_b/N_0 = 10$  дБ

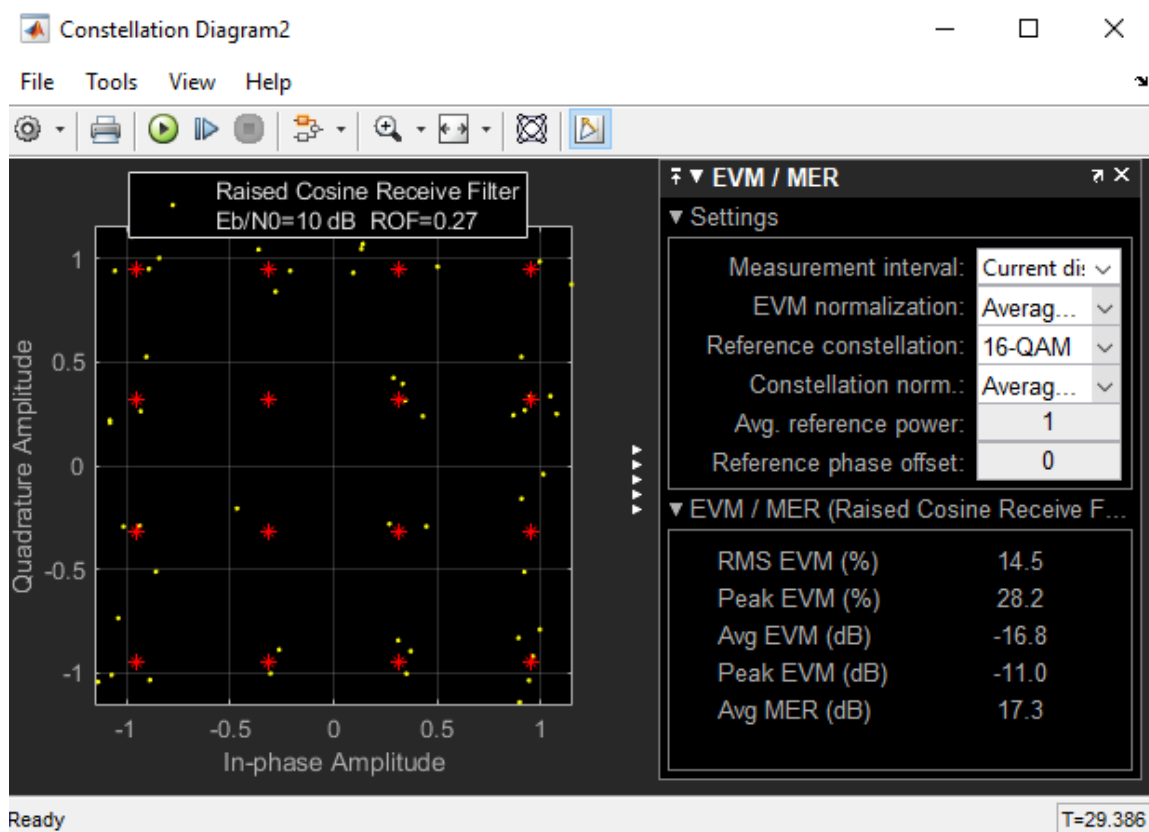


Рисунок 6.79 – Вектограмма модулированного сигнала для 16-QAM при  $E_b/N_0 = 10\text{дБ}$

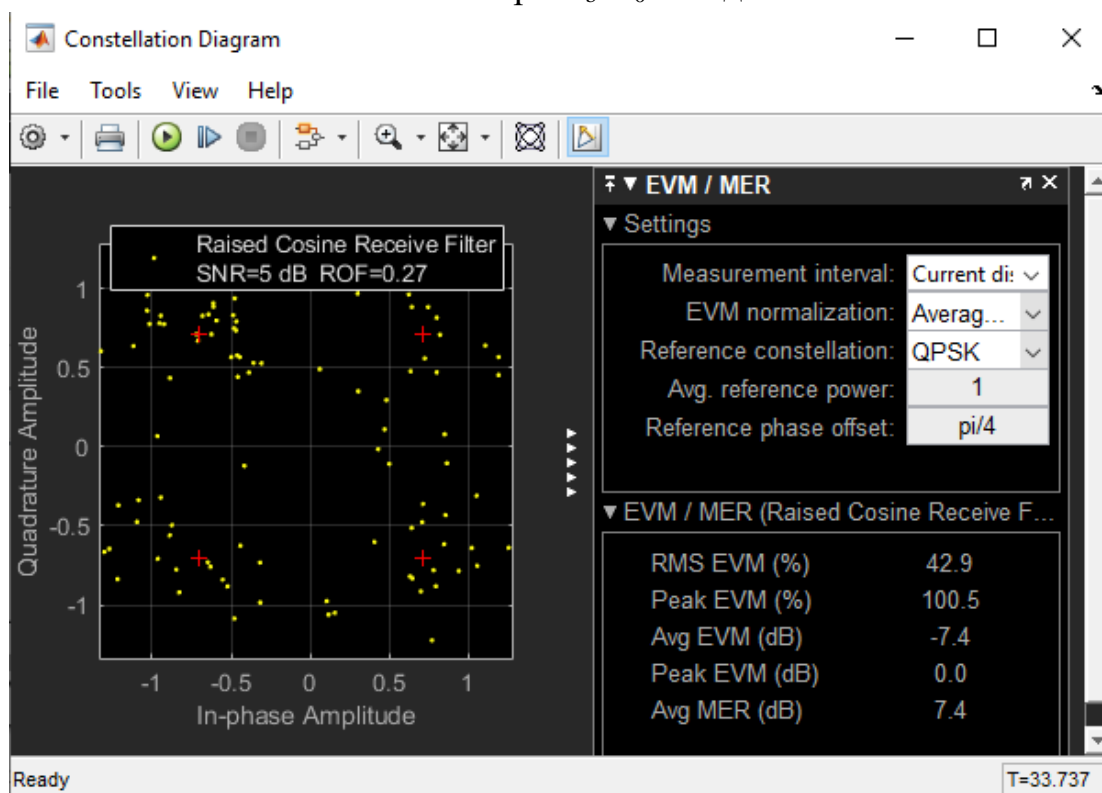


Рисунок 6.10 – Вектограмма модулированного сигнала для 4-PSK при  $E_b/N_0 = 5\text{ дБ}$

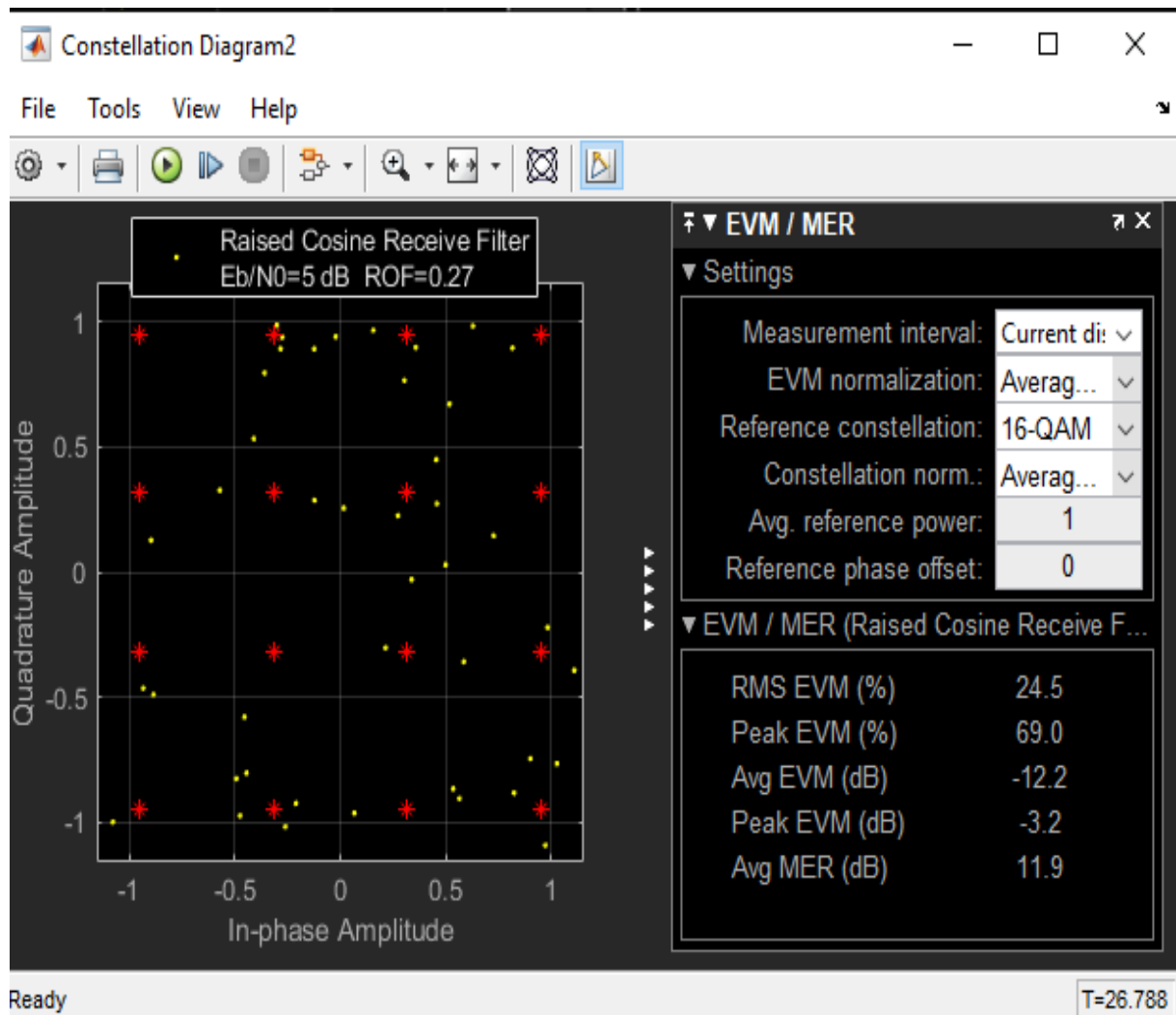


Рисунок 6.11 – Вектограмма модулированного сигнала для 16-QAM при  $E_b/N_0 = 5$  дБ

6 Построение зависимостей коэффициента ошибок (BER) от отношения  $E_b/N_0$ .

6.1 Для определения исследуемого диапазона отношения  $E_b/N_0$  откройте специализированный графический пользовательский интерфейс Bit Error Rate Analysis и откройте графики для двух заданных видов модуляции, в режиме Theoretical путём нажатия кнопки Plot.

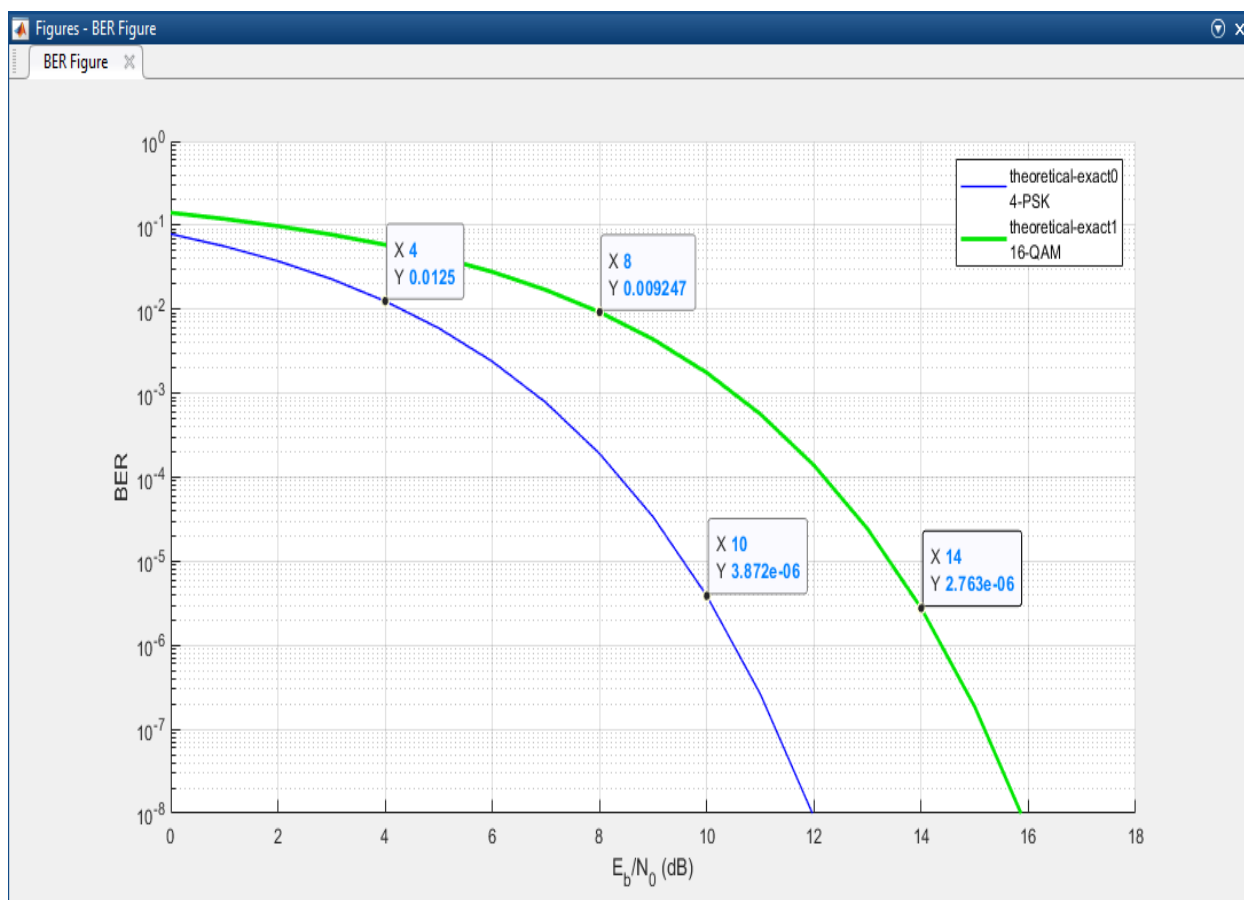


Рисунок 6.12 – Теоретическая зависимость BER от  $E_b/N_0$  для двух заданных видов модуляции

6.2 По полученным теоретическим графикам (рисунок 6.10) определим исследуемый диапазон и четыре значения отношения  $E_b/N_0$ : для 4-PSK это 4, 6, 8 и 10 дБ; для 16- QAM это 8, 10, 12 и 14 дБ.

6.3 Откройте окно модели для первого вида модуляции 4-PSK, установите одно из заданных значений  $ROF = 0.27$  и первое из определённых выше значение  $E_b/N_0 = 4$  дБ, заполните аннотацию на модели и запустите симуляцию. Симуляцию остановить при общем количестве бит не менее  $10^7$  (нижнее окно дисплея на выходе Error Rate Calculation). После окончания симуляции сделать скриншот модели.

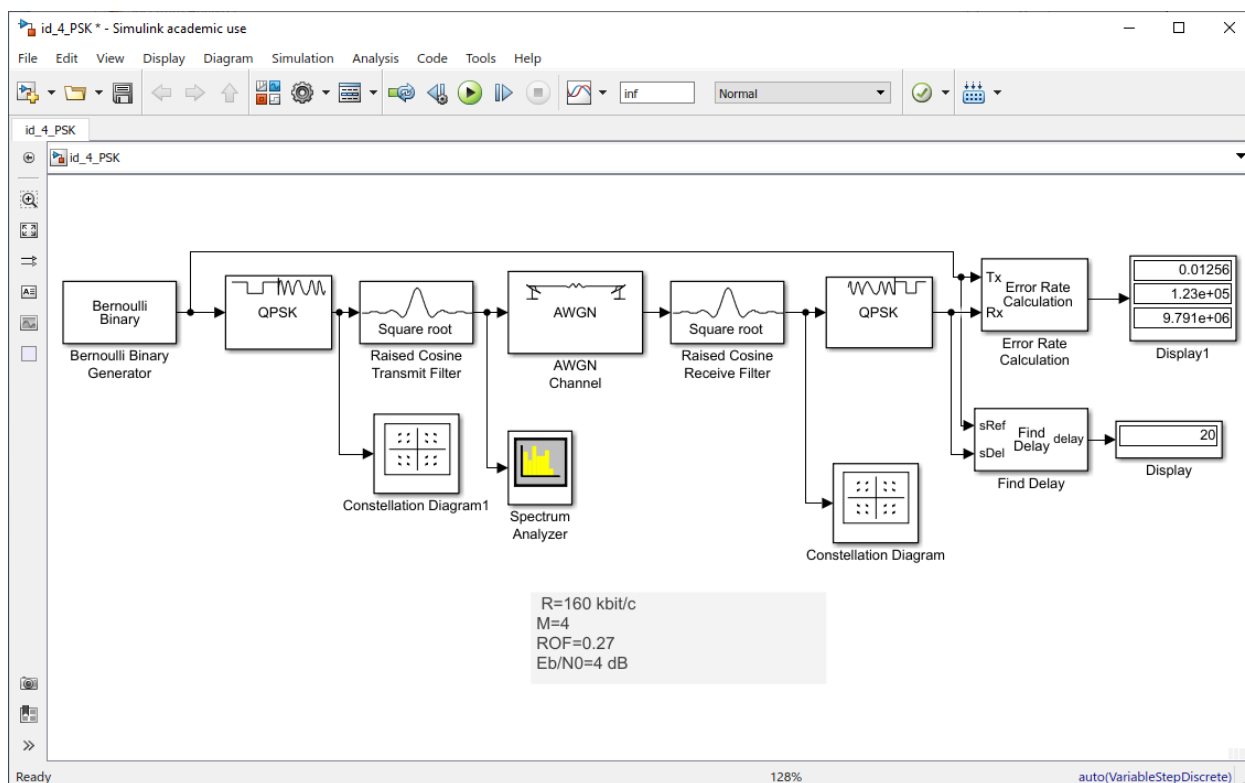


Рисунок 6.13 – Модель системы связи с модуляцией 4-PSK на панели Simulink

6.4 Прodelайте п. 6.3 для остальных трёх значений  $E_b/N_0$  и запишите значения BER (скриншоты модели не делать).

Таблица 6.1 – Зависимость BER от  $E_b/N_0$  при модуляции 4-PSK

$E_b/N_0$ , дБ	4	6	8	10
BER	$1,256 \times 10^{-2}$	$2,4 \times 10^{-3}$	$1,87 \times 10^{-4}$	$3,0 \times 10^{-6}$

6.5 Откройте окно модели для второго заданного вида модуляции 16-QAM, установите одно из заданных значений  $ROF = 0.27$  и первое из определённых выше значение  $E_b/N_0 = 8$  дБ, заполните аннотацию на модели и запустите симуляцию. Далее проделайте тоже, что и в п.п. 6.3 и 6.4.



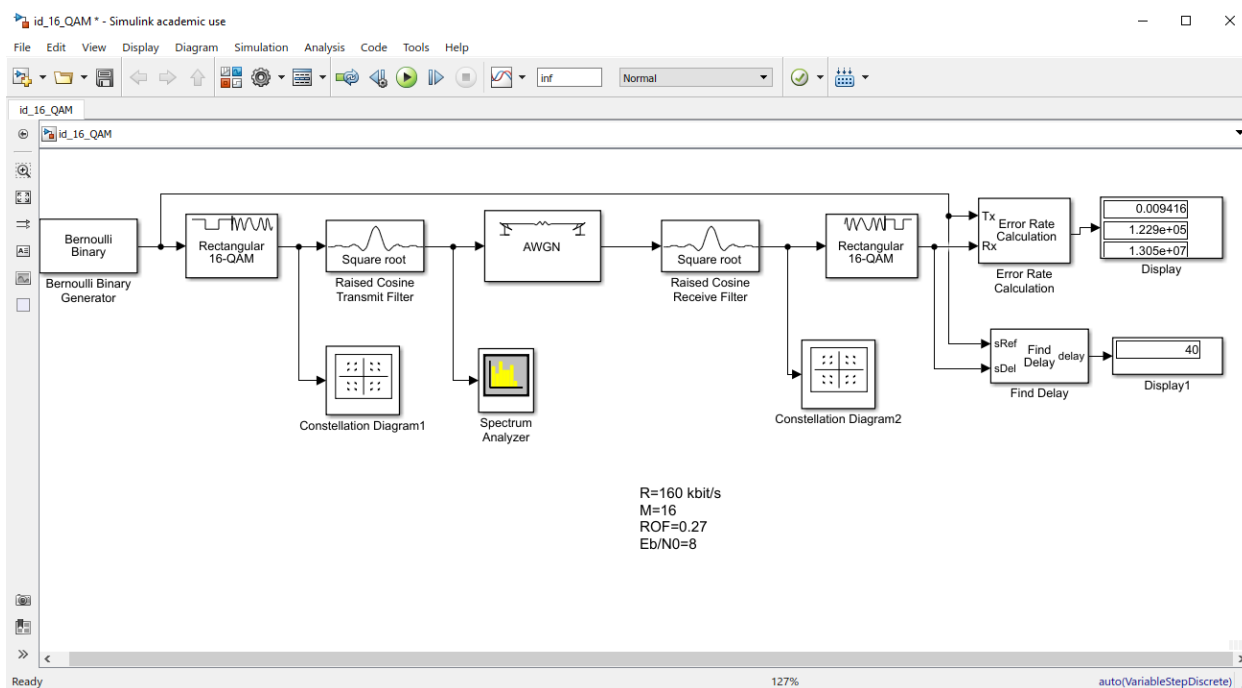


Рисунок 6.14 – Модель системы связи с модуляцией 16- QAM на панели Simulink

Таблица 6.2 – Зависимость BER от  $E_b/N_0$  при модуляции 16- QAM

$E_b/N_0$ , дБ	8	10	12	14
BER	$9,4 \times 10^{-3}$	$1,83 \times 10^{-3}$	$1,5 \times 10^{-4}$	$3,6 \times 10^{-6}$

6.6 По четырём точкам постройте графики зависимости BER от  $E_b/N_0$  , сравните их с теоретическими (п.6.1) и поясните полученные результаты.

## ЗАКЛЮЧЕНИЕ

В данной работе представлено исследование влияния методов модуляции, канального кодирования и реальных характеристик радиотракта на помехоустойчивость приема в системах цифровой радиосвязи.

Методы модуляции M-PSK и M-QAM успешно применяются в системах радиорелейной и спутниковой связи, устройствах беспроводного абонентского доступа, сетях сотовой связи и т.д. Важной особенностью, обуславливающей применение этих видов модуляции, является возможность достижения компромисса между помехоустойчивостью системы и скоростью передачи информации в ней, а также относительная простота реализации устройств. В данной работе были проведены исследования характеристик модулированных сигналов с точки зрения помехоустойчивости и эффективности использования частотного диапазона. Исследование проведено на базе компьютерной модели, реализованной в системе MATLAB. В работе также продемонстрированы возможности MATLAB для моделирования систем связи, на примере простейшей модели системы телекоммуникаций.

Свойства многофазовых сигналов были рассмотрены в условиях белого Гауссовского шума и замираний Релея и Райса. Было показано, что из трех методов фазовой модуляции – PSK и DPSK с когерентным и некогерентным обнаружением – наиболее помехоустойчивым является модуляция PSK, что обуславливает широкое практическое применение её вариантов BPSK и QPSK в сотовых сетях второго поколения (IS-95, W-CDMA) и некоторых реализациях стандарта IEEE 802.11. Однако, помехоустойчивость в некоторых случаях является менее приоритетной задачей по сравнению с простотой исполнения системы. В этих случаях предпочтение может отдаваться менее сложной, с точки зрения практической реализации, модуляции DPSK. Этот вид модуляции немногим уступает PSK в помехоустойчивости (в пределах 1-2 дБ), из-за чего применяется например в системах транкинговой связи стандарта PWT, в стандарте сотовой связи IS-136 ( $\pi/4$  DQPSK) и в реализациях 802.11 не требующих высоких скоростей передачи информации (DBPSK, DQPSK).

Для обеспечения наибольших скоростей передачи применяется модуляция вида M-QAM. Сигналы QAM являют собой пример более эффективного использования спектра, чем у сигналов M-PSK. В частности,

16-ти позиционная QAM модуляция по сравнению с 8-PSK при помехозащищенности худшей примерно на 1,5 дБ характеризуются скоростью передачи информации большей на одну треть. Но при этом, при увеличении позиционности, помехоустойчивость QAM модуляции значительно понижается, что требует введения более мощного избыточного кодирования. Таким образом, применение QAM наиболее оправдано в высокоскоростных системах передачи данных, работающих в условиях малых помех. Примеры областей применения QAM: стандарт IEEE 802.11 (скорости 24Мбит/с – 54Мбит/с) – 16-QAM, 64-QAM; спутниковые системы связи; радиорелейные линии связи.

Исследование влияния замираний Райса и Релея на модулированные сигналы показало, что замирания крайне негативно влияют на помехоустойчивость радиоканала, приемлемая помехоустойчивость достигается исключительно при большом отношении сигнал/шум и незначительном влиянии замираний. На практике эта проблема решается применением комплексных методов борьбы с замираниями (избыточное кодирование в сочетании с перемежением, эквалайзинг, использование RAKE приемников и пр.), рассмотрение которых выходит за рамки данного исследования.

Вторая часть исследования посвящена влиянию способов кодирования сигнала на помехоустойчивость канала радиосвязи. Здесь подробно рассмотрено поведение различных типов кодов, проведен сравнительный анализ кодеров и декодеров по эффективности, сложности реализации и областям применения. Так было показано, что циклические блочные коды в основном используются при обнаружении ошибок в системах с циклической проверкой четности с избыточностью (CRC). Они позволяют обнаруживать одиночные и множественные ошибки при внесении очень малой избыточности (в протоколе Ethernet – около 4%). Для систем с исправлением ошибок используют сверточное кодирование. Почти во всех приложениях сверточные коды лучше блочных при той же конструктивной сложности кодера и декодера. Для каналов спутниковой связи схемы прямого исправления ошибок позволяют легко понизить на 5-6 дБ требуемое значение SNR для заданной достоверности передачи. Коды Рида-Соломона особенно привлекательны, поскольку эффективность кода растет с его длиной. При большой длине блока коды можно сконфигурировать таким образом, что время декодирования будет значительно меньше, чем у других кодов с той же длиной блока. Это связано с тем, что декодер работает с целыми

символами, а не битами. На практике, для повышения эффективности кодирования используют каскадные коды – сочетание сверточного кодирования и кодов Рида-Соломона.

Среда MATLAB является удобным инструментом для практического подтверждения теоретических выкладок. В программе Simulink, входящей в пакет MATLAB очень качественно реализованы модели реальных устройств и процессов, с помощью которых можно проверить достоверность теоретических результатов не используя реальных устройств и сред, и не описывая модели в виде сложных и больших программ. Удобный интерфейс и богатые функциональные возможности MATLAB позволяют не только на практике подтверждать описанные теоретически модели, но и исследовать совершенно новые системы и ставить сложнейшие эксперименты, подтверждая или опровергая существующие гипотезы.

Проведенное исследование на практике доказало теоретически полученные зависимости. Сконструированная модель системы радиосвязи является базой для продолжения исследований в области телекоммуникаций. В дальнейшем можно использовать данную модель в качестве ядра по следующим направлениям:

- Исследование реальных систем цифровой радио и проводной связи;
- Построение моделей новых систем и отдельных устройств для практического обоснования современных теоретических исследований;
- Создание моделей для интерактивного обучения студентов на лабораторных занятиях.

## СПИСОК ЛИТЕРАТУРЫ

- 1 Вишневский В.М., Портной С.Л., Шахнович И.В. Энциклопедия WiMAX. Путь к 4 G. М.: Техносфера, 2009. – 472 с.
- 2 Тихвинский В.О., Терентьев С.В., Высочин В.П. Сети мобильной связи LTE/LTE Advanced: Технологии 4G, приложения и архитектура. – М.: Издательский дом Медиа Паблишер, 2014. 348 с.
- 3 Г.П. Катунин, Г.В. Мамчев, В.И. Носов, В.П. Шувалов Телекоммуникационные системы и сети: Учебное пособие. В трёх томах. Том 2. – Радиосвязь, радиовещание, телевидение. УМО по направлению «Телекоммуникации»/ Под ред. Профессора В.П. Шувалова. – 4-е изд., перераб. и доп.– М.: Горячая линия - Телеком, 2017. – 564 с.
- 4 Носов В.И., Тимощук Р.С., Дроздов Н.В. Моделирование систем связи в среде MATLAB: Учебное пособие. – Новосибирск.: СибГУТИ 2007. – 178 с.
- 5 И.В. Черных SIMULINK среда создания инженерных приложений. / Под общей редакцией канд. технич. таук В.Г. Потёмкина. – М.: Диалог МИФИ, 2004. – 491 с.
- 6 Дьяконов В.П. Simulink 5/6/7. Самоучитель. – М.: ДМК-Пресс, 2008. – 784 с.
- 7 [www.youtube.be/gi8\\_vEHsh6w](http://www.youtube.be/gi8_vEHsh6w)
- 8 [www.exponenta.ru](http://www.exponenta.ru)
- 9 [www.matlab.ru](http://www.matlab.ru)
- 10 [www.mathworks.ru](http://www.mathworks.ru)
- 11 [www.youtube.com/user/MATLABinRussia/playlists](http://www.youtube.com/user/MATLABinRussia/playlists)
- 12 Носов В.И. Радиорелейные линии синхронной цифровой иерархии. Часть 1 – Многоуровневые кодеры, модемы и эквалайзеры. Учебное пособие. УМО по специальности связь. – Новосибирск.: СибГУТИ, 2003. – 156 с.
- 13 А.Б. Сергиенко Цифровая обработка сигналов: Учебное пособие. – СПб.: Питер, 2002. – 608 с.
- 14 В.И. Носов Основы цифровой передачи сигналов и построения радиорелейных линий синхронной цифровой иерархии Учебное пособие. УМО по специальности связь. – Новосибирск.: СибГУТИ, 2005. – 256 с

**Владимир Иванович Носов**

**МОДЕЛИРОВАНИЕ СИСТЕМ СВЯЗИ В СРЕДЕ MATLAB SIMULINK**

Учебное пособие

Редактор: Б.Н. Маглицкий  
Корректор:

---

Лицензия №020475, январь 1998г.

Формат бумаги 62х84 1/16, отпечатано на ризографе, шрифт №10,  
изд. л. . Заказ №\_\_\_\_\_, тираж – \_\_\_\_ экз.

Типография СибГУТИ

630102, Россия, Новосибирск, ул. Кирова, 86