А.М. Голиков

КОДИРОВАНИЕ ИНФОРМАЦИИ В РАДИОЭЛЕКТРОННЫХ СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ

Сборник компьютерных лабораторных работ

Томск 2018

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
Томский государственный университет систем управления и радиоэлектроники

А.М. ГОЛИКОВ

КОДИРОВАНИЕ ИНФОРМАЦИИ В РАДИОЭЛЕКТРОННЫХ СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ

Сборник компьютерных лабораторных работ

Томск 2018

УДК 621.39(075.8)

ББК 32.973(я73)

Голиков А.М.

Кодирование информации в радиоэлектронных системах передачи информации: Сборник компьютерных лабораторных работ / А.М.Голиков. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2018. – 318 с.: ил. — (Учебная литература для вузов)

Сборник компьютерных лабораторных работ предназначен для направления подготовки магистров 11.04.02 "Инфокоммуникационные технологии и системы связи" по магистерским программам подготовки: "Радиоэлектронные системы передачи информации", "Оптические системы связи и обработки информации", "Инфокоммуникационные системы беспроводного широкополосного доспупа", "Защищенные системы связи", для направления подготовки магистров 11.04.01 "Радиотехника" по магистерской программе подготовки: "Радиотехнические системы и комплексы", "Радиоэлектронные устройства передачи информации", "Системы и устройства передачи, приема и обработки сигналов", "Видеоинформационные технологии и цифровое телевидение" и специалитета 11.05.01 "Радиоэлектронные системы и комплексы" специализации "Радиолокационные системы и комплексы", "Радиоэлектронные системы передачи информации", "Радиоэлектронные системы космических комплексов", а также бакалавриата направления 11.03.01 "Радиотехника" (Радиотехнические средства передачи, приема и обработки сигналов), бакалавриата 11.03.02 Инфокоммуникационные технологии и системы связи (Системы мобильной связи, Защищенные системы и сети связи, Системы радиосвязи и радиодоступа, Оптические системы и сети связи) и может быть полезна аспирантам. Представлены описания программных комплексов и методики выполнения лабораторных и практических работ. Программные комплексы разработаны на базе ПО MATLAB, LabVIEW, SystemView. Всем заинтересованным лицам или организациям будут предоставлены рабочие программы на CD или DVD.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....7
1. ЦИФРОВЫЕ ВИДЫ МОДУЛЯЦИИ И СИГНАЛЬНОГО КОДИРОВАНИЯ, ИХ СПЕКТРАЛЬНАЯ И ЭНЕРГЕТИЧЕСКАЯ ЭФФЕКТИВНОСТЬ.....14
1.1. Анализ цифровых методов модуляции.....14
1.2. Модемы сотовой связи FSK, MSK GMSK и численный анализ вероятности символьной ошибки с использованием ПО LabVIEW.....30
1.3. Модемы спутниковых систем связи M-QAM, M-PSK и численный анализ вероятности символьной ошибки с использованием ПО LabVIEW.....47
2. ПРОПУСКНАЯ СПОСОБНОСТЬ КАНАЛА СВЯЗИ. КОДИРОВАНИЕ ИСТОЧНИКА.....69
2.1. Пропускная способность канала связи. Объем сигнала и емкость канала связи, условия их согласования.....69
2.2. Исследование кодирования источника дискретных сообщений методами Шеннона-Фано.....71
2.3. Исследование алгоритмов Лемпеля - Зива.....78
2.4. Фрактальные методы кодирования изображений..... 87
2.5.Вейвлет преобразования сигналов и изображений с использованием..... 99
3. ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ..... 122
3.1. Исследование кодов Хемминга, БЧХ (Боуза-Чоудхури-Хоквенгема), Рида-Соломона на базе MATLAB..... 122
3.2. Циклические избыточные коды CRC (Cyclic redundancy check)..... 156
3.3. Сверточные коды. Декодирование сверточных кодов..... 170
3.4. Декодирование сверточных кодов по методу Витерби с использованием ПО MATLAB..... 191
3.5. Турбокодирование. Обобщенная схема турбокодера с параллельным каскадированием. Сверточные турбокоды. Декодирование турбокодов. Характеристики помехоустойчивости сверточных турбокодов . Исследование турбокодов с использованием ПО MATLAB..... 211
3.6. Низкоплотностные коды. Классификация LDPC-кодов. Методы построения проверочных матриц. Алгоритмы декодирования низкоплотных кодов. Оценка сложности алгоритмов декодирования на базе MATLAB и LabVIEW..... 225
3.7. Исследование каскадных кодов.....243

4. СИГНАЛЬНО-КОДОВЫЕ КОНСТРУКЦИИ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ.....265
4.1. Сигнально-кодовые конструкции на основе Треллис кодовой модуляции (TCM) и их анализ с использованием MATLAB..... 292
4.2. Исследование сигнально-кодовой конструкции на базе системы с ортогональным частотным мультиплексированием и пространственно-временным кодированием
ЛИТЕРАТУРА.....318

ВВЕДЕНИЕ

В зависимости от назначения и условий функционирования системы радиосвязи ее эффективность оценивается на основании тех или иных показателей (критериев), основными из которых являются энергетический и спектральный. Соответственно, важнейшими характеристиками любой системы радиосвязи являются энергетическая и спектральная эффективность, характеризующие, соответственно, энергетические затраты и полосу занимаемых частот, необходимые для передачи сообщений.

К сожалению, одновременное достижение предельных значений этих показателей эффективности оказывается невозможным, так что в каждом конкретном случае построения системы радиосвязи приходится руководствоваться компромисными соображениями при оптимизации характеристик и режимов функционирования системы [1-4].

Эффективные методы модуляции и помехоустойчивого кодирования все шире используются в современных цифровых телекоммуникационных системах (ТКС). Переход в ТКС к ансамблям многопозиционных сигналов увеличивает информационную скорость и обеспечивает передачу больших потоков информации. Современная элементная база позволяет применять в ТКС достаточно сложные методы помехоустойчивого кодирования и обеспечивать тем самым высокую верность передачи информации. Исследованию методов модуляции и кодирования в ТКС посвящена обширная литература, отражающая как достижения теории, так и вопросы ее практических приложений [1-18].

Как известно, теория сигналов и теория кодирования длительное время развивались независимо. В последние годы значительно возрос интерес к новому перспективному направлению, возникшему на стыке этих наук. В работах отечественных и зарубежных авторов интенсивно исследуются возможности ТКС, в которых для передачи информации используются ансамбли многопозиционных сигналов в сочетании с помехоустойчивыми кодами, причем, процедуры модуляции/кодирования (демодуляции/декодирования) осуществляются ***совместно.*** При рациональном построении такие ***сигнально-кодовые конструкции*** (СКК) сочетают в себе положительные качества как многопозиционных ансамблей сигналов, так и помехоустойчивых кодов, допускают достаточно простые и реализуемые на практике алгоритмы декодирования и при использовании их в ТКС позволяют существенно продвинуться к теоретическим пределам эффективности. Вопросы синтеза таких систем модуляции/кодирования, анализа их структуры, помехоустойчивости и эффективности, демодуляции/декодирования составляют основное содержание нового перспективного направления в теории связи - ***теории сигнально-кодовых конструкций.***

Один путь базируется на введенном А.Г. Зюко [1, 2] понятии эффективности систем связи. Анализ и сравнение эффективности систем с многопозиционными сигналами и систем с корректирующими кодами приводит в ***информационной теории СКК*** к идее *их комбинации в единой сигнально-кодовой конструкции.* Такой путь является наглядным, убедительным и излагается без строгих математических выкладок. Этот путь может быть использован для популярного изложения оснований теории СКК [1-4].

Статистическая теория связи предлагает большое количество вариантов построения телекоммуникационных систем. Как из этого множества выбрать вариант, наиболее целесообразный в заданных условиях? По каким критериям следует производить этот выбор? Насколько оправдано применение тех или иных новых систем и как совершенствовать существующие системы? Принципиальное решение этих вопросов в конечном итоге сводится к *оптимизации систем связи по критериям эффективности.*

Типовая структура одноканальной системы передачи дискретной информации (СПДИ) приведена на рис. 1.1. Источник вырабатывает сообщения, которые необходимо передавать по каналу СПДИ. Это могут быть последовательности *дискретных* сообщений (данные, телеграфные сообщения и т.д.) либо *непрерывные* сообщения (речь, телевидение, результаты телеизмерений и др.), преобразованные в *цифровую форму.*

Реальные сообщения содержат *избыточность* и для согласования источника с каналом передачи информации используют *кодер источника.* Совместно с *декодером* они образуют *кодек источника.* Методы кодирования источника изложены в работах [2, 3, 7, 9-11, 13-16, 18].

Основные требования к СПДИ формулируются достаточно просто:

1. ***верность,***
2. ***скорость,***
3. ***своевременность доставки информации от отправителя к получателю.***

В системах без помехоустойчивого кодирования верность и скорость зависят от вида используемых сигналов-переносчиков. Применение помехоустойчивого кодирования позволяет повысить верность передачи, но за счет снижения скорости. С другой стороны, выбором сигналов можно добиться повышения скорости передачи информации, но, зачастую, в ущерб верности передачи. Однако, следует учитывать, что применение корректирующих кодов вносит задержку в передачу цифровых данных, что ухудшает такой показатель, как своевременность доставки информации. В каждом конкретном случае телекоммуникационных систем (ТКС) могут быть сформулированы количественные выражения этих требований. Это зависит от назначения ТКС и вида передаваемого

сообщения. В общем случае показатели верности, скорости и своевременности передачи информации находятся в противоречивых соотношениях.

Кроме отмеченных выше, следует упомянуть о *дополнительных требованиях,* которые определяют применение того или иного метода передачи информации:

Спектральная эффективность. Современные ТКС работают в условиях дефицита частотного спектра, отводимого для передачи информации. Это и обусловливает важность методов формирования *компактных спектров сигналов,* передаваемых по линиям связи с ограниченной полосой частот.

Сложность реализации методов модуляции-кодирования. Теория информации указывает путь повышения помехоустойчивости - использование для передачи *длинных последовательностей сигналов и кодов,* для извлечения информации из которых на приемной стороне приходится применять достаточно сложные алгоритмы обработки. Поэтому *конструктивная теория кодирования* направлена на поиск корректирующих кодов, допускающих *реализуемые алгоритмы декодирования* [4].

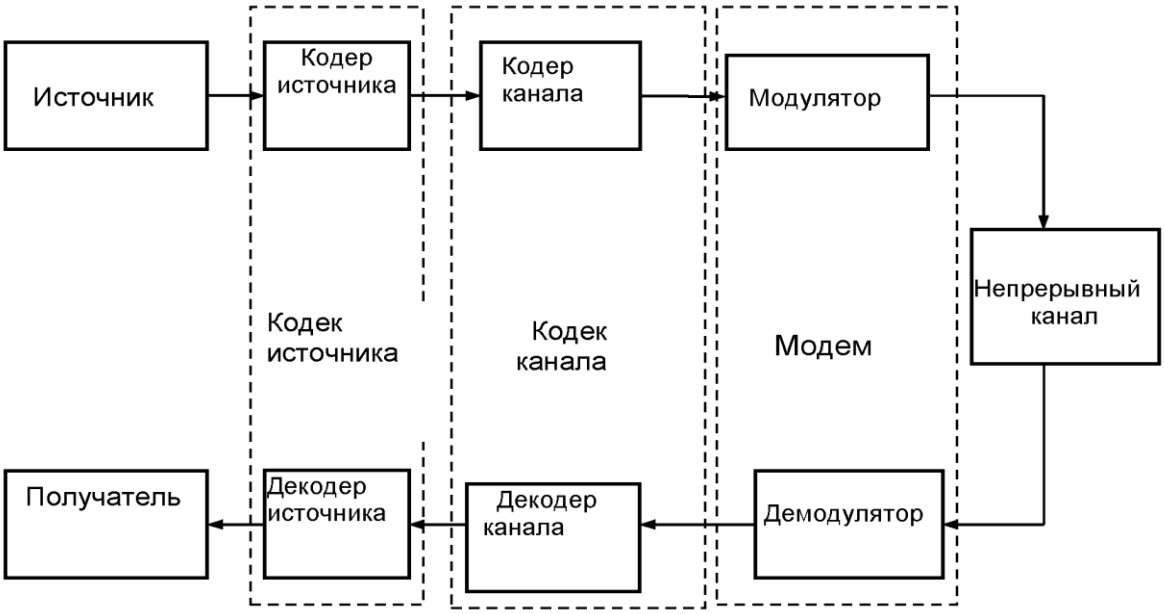


Рис. 1.1. Структурная схема СПДИ

Согласно известных теорем К. Шеннона, в принципе *возможно сколь угодно большое повышение верности передачи информации,* если скорость передачи по каналу $R_{кан}$ не превышает пропускной способности канала $C_к$**.** Достигается это применением достаточно длинных *корректирующих кодов* (КК). С этой целью в структуру КК вводится *избыточность.*

Кодек КК (кодер и декодер канала) показан на рис. 1.1. В реальных условиях длина кода ограничена допустимой сложностью устройств кодирования и, прежде всего,

декодирования, поэтому эффект от применения корректирующих кодов зависит от параметров кода и ограничений на реализацию кодека канала.

Современная теория предлагает широкий набор корректирующих кодов, различных по структуре, принципам построения и корректирующей способности. *Кодем СКК* включает в себя сочетание кодека/модулятора и, соответственно, демодулятора/декодера.

В общем случае результат работы системы связи определяется количеством и качеством передаваемой информации. Количество оценивается скоростью передачи информации по каналу (бит/с), а качество - величиной ошибки. Согласно теоремам К. Шеннона, ошибка при соответствующем выборе метода передачи (модуляции/кодирования) может быть сделана произвольно малой. В то же время, скорость передачи не может быть выше некоторого информационного ресурса, называемого *пропускной способностью канала С.*

В работе А.Г. Зюко [2] было предложено считать одним из показателей эффективности системы величину средней скорости $R_{кан}$ при коМесто для формулы.торой обеспечивается заданная верность передачи информации. При этом могут быть определены следующие *показатели эффективности:*

- Информационная эффективность системы, определяющая степень использования пропускной способности канала относительной величиной

$$\eta = \frac{R_{кан}}{C} \tag{1.1}$$

В реальных условиях показатель η всегда меньше единицы. Чем ближе η к единице, тем совершеннее система передачи информации. Достижение необходимых скорости и верности передачи сопровождается определенными затратами других важнейших ресурсов: ***мощности сигнала*** P_c **и** ***полосы частот канала*** F_k**.** Такой подход позволил ввести показатели эффективности использования ресурсов системы, а именно:

- Энергетическая эффективность

$$\beta = \frac{R_{кан}}{P_C / N_0} \tag{1.2}$$

где N_0 - спектральная плотность мощности шума;

- Частотная эффективность

$$\gamma = \frac{R_{кан}}{F} \tag{1.3}$$

где F - полоса частот канала.

Показатели β и γ имеют смысл удельных скоростей, а обратные величины $\beta' = 1/\beta$ и $\gamma' = 1/\gamma$ определяют *удельные расходы соответствующих ресурсов* на передачу информации с единичной скоростью (1 бит/с).

Для гауссовского канала с полосой F, отношением мощностей сигнала и шума $\rho = \frac{P_c}{P_ш}$ и пропускной способностью $C = F\log_2(1+\rho)$ была установлена связь между показателями эффективности

$$\eta = \frac{\gamma}{\log_2(1+\frac{\gamma}{\beta})} \qquad \text{и } \gamma = \rho\beta$$

Для идеальной системы (η) определена также предельная зависимость

$$\beta = \frac{\gamma}{2^{\gamma}-1}. \tag{1.5}$$

Зависимость энергетической эффективности от удельной скорости удобно представить в виде кривой на плоскости $\beta\gamma$ (рис. 1.2). Она отражает наилучший обмен между β и γ в непрерывном канале (НК-предел Шеннона). При этом частотная эффективность γ изменяется в пределах от 0 до ∞ то время как энергетическая эффективность ограничена сверху, поскольку

$$\beta_{\max} = \lim_{\gamma\to 0}\beta = \lim_{\gamma\to 0}\frac{\gamma}{2^{\gamma}-1} = \frac{1}{\ln 2} \approx 1{,}443(1{,}59дБ) \tag{1.6}$$

Аналогичные предельные кривые могут быть построены для других типов каналов [2] (ДНК-дискретно-непрерывный канал, ДСК-дискретный симметричный канал на рис. 1.2).

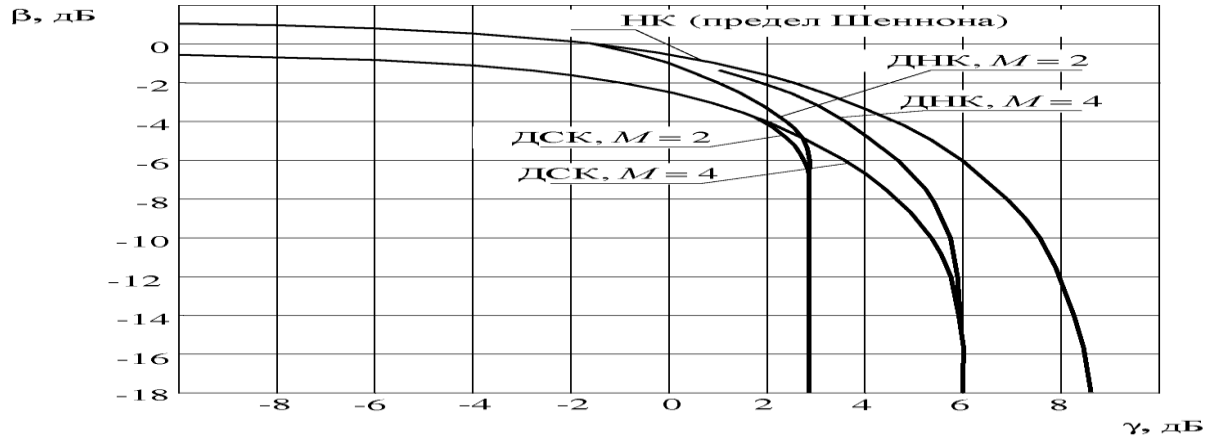


Рис. 1.2. Кривые предельной эффективности систем связи

Полосу частот, занимаемую сигналом конечной длительности не всегда можно определить однозначно. Поэтому далее для анализа и сравнения эффективности будем рассматривать следующую модель процесса передачи информации.

Полагаем, что на интервале T по каналу передается определенное количество независимых отсчетов N_T (по В. А. Котельникову $N_T = 2FT$).

Тогда $D = \frac{N_T}{T}$ число передаваемых отсчетов в единицу времени.

В этой модели могут быть определены следующие показатели:

- Удельная скорость передачи информации

$$RN = R_{кан} / D \ (бит / отсчет), \tag{1.7}$$

-Удельная информационная емкость* канала

$$C_N = \frac{C}{D} \ (бит / отсчеть), \tag{1.8}$$

- Удельная средняя энергия сигнала, затрачиваемая на передачу одного отсчета

$$EN = P_C / D$$

- Удельная эффективность использования пропускной способности канал

$$\eta_N = \frac{R_N}{C_N} = \eta, \tag{1.9}$$

- Удельная энергетическая эффективность (удельная эффективность использования энергии сигнала)

$$\beta_N = \frac{R_N}{E_N / N_0} = \beta, \tag{1.10}$$

- Удельная скорость передачи информации

$$\gamma_N = R_N. \tag{1.11}$$

Переход к удельным показателям эффективности означает, что процесс формирования сигналов рассматривается в ортогональном базисе. При этом удельные показатели

эффективности (1.9) и (1.10) точно совпадают с каноническими (1.4), (1.5), но удается избежать неопределенности при оценке полосы частот [1, 2].

В Главе 1 учебного пособия рассматриваются цифровые виды модуляции и сигнального кодирования, их спектральная и энергетическая эффективность. Приводятся краткие теоретические сведения и практическая их реализация на базе программного обеспечения NI LabVIEW. Даны результаты исследования вероятности ошибки на бит от отношения сигнал шум в канале для различных видов модуляции FSK, MSK и GMSK, а также QAM, M-QAM, PSK, M-PSK, созвездия для многопозиционных методов модуляции, спектры, глазковые диаграммы [11] и джиттер.

В Главе 2 рассматривается пропускная способность канала связи. Кодирование источника. Теорема К.Шеннона о кодировании в дискретном канале без помех. Проводится исследование кодирования источника дискретных сообщений методами Шеннона-Фано, исследование алгоритмов Лемпеля - Зива, Фрактальные методы кодирования изображений, а также вейвлет преобразования сигналов и изображений с использованием ПО MATLAB [13].

В Главе 3 исследуется помехоустойчивое кодирование в телекоммуникационных системах. Исследуются характеристики кодов Хемминга, БЧХ (Боуза-Чоудхури-Хоквенгема) и Рида-Соломона на базе MATLAB. Проведено исследование циклического избыточного кода (Cyclic redundancy check) CRC с использованием ПО MATLAB. Рассмотрены кодеры сверточных кодов и методы их декодирование, включая метод Витерби, исследованы их характеристики с использованием ПО MATLAB. Подробно исследовано турбокодирование, обобщенная схема турбокодера с параллельным каскадированием. Сверточные турбокоды. Декодирование турбокодов. Оценены характеристики помехоустойчивости сверточных турбокодов. Исследование турбокодов проведено с использованием ПО MATLAB. Рассмотрены низкоплотностные коды, дана классификация LDPC-кодов, показаны методы построения проверочных матриц. Реализованы алгоритмы декодирования низкоплотных кодов, проведена оценка сложности алгоритмов декодирования на базе MATLAB и NI LabVIEW, исследованы каскадные коды на базе MATLAB Simulink 2015 [1, 3, 8-11, 16].

В Главе 4 рассмотрены сигнально-кодовые конструкции в телекоммуникационных системах, критерии эффективности систем передачи информации, информационная теория сигнально-кодовых конструкций, системы многочастотной OFDM модуляции - проведено имитационное моделирование в среде LabVIEW, реализованы Треллис кодовая модуляция (ТСМ) и проведен ее анализ с использованием MATLAB, рассмотрены методы

пространственно-временного кодирования (MIMO) и оценена их эффективность на базе NI LabVIEW [12, 16].

В Главе 5 рассмотрены модемы и кодеки современных телекоммуникационных систем, имитационное моделирование системы мобильной связи стандартов GSM на базе ПО MATLAB, имитационное моделирование системы мобильной связи стандарта CDMA на базе ПО MATLAB, имитационное моделирование системы мобильной связи стандарта IEEE 802.11 (WiFi) на базе MATLAB 2015, имитационное моделирование системы мобильной связи стандарта IEEE 802.15.4 ZigBee на базе ПО MATLAB, имитационное моделирование системы мобильной связи стандарта IEEE 802.15.1 (Bluetooth) на базе ПО MATLAB, имитационное моделирование системы мобильной связи стандарта IEEE 802.16 (WiMAX) на базе ПО MATLAB [16, 18], имитационное моделирование системы мобильной связи стандарта IEEE 802.20 LTE на базе ПО MATLAB, имитационное моделирование системы цифрового наземного телевизионного вещания DVB-T [16], имитационное моделирование системы цифрового спутникового телевизионного вещания DVB-S и системы высокоскоростного цифрового спутникового ТВ-вещания DVB-S2 [16], имитационное моделирование системы цифрового кабельного телевизионного вещания DVB-C и системы высокоскоростного цифрового кабельного ТВ-вещания DVB-C2 [16], имитационное моделирование системы цифрового мобильного телевизионного вещания DVB-H и системы высокоскоростного цифрового мобильного ТВ-вещания DVB-H2 [16, 19, 20].

ГЛАВА 1. ЦИФРОВЫЕ ВИДЫ МОДУЛЯЦИИ И СИГНАЛЬНОГО КОДИРОВАНИЯ, ИХ СПЕКТРАЛЬНАЯ И ЭНЕРГЕТИЧЕСКАЯ ЭФФЕКТИВНОСТЬ

1.1. Анализ цифровых методов модуляции

Модулятор

Известно [6, 7], что узкополосный модулированный сигнал с произвольным видом модуляции можно представить в виде:

$$s(t) = I(t)\cos(\omega t) - Q(t)\sin(\omega t), \qquad (1.12)$$

где ω - несущая частота радиосигнала, $I(t)$ и $Q(t)$ называются соответственно *синфазной и квадратурной* составляющими модулирующего сигнала.

Таким образом, для осуществления произвольного вида модуляции сигнала необходимо

выполнить две операции: (1) сформировать синфазную и квадратурную составляющие модулирующего сигнала (вид данных составляющих будет определять вид модуляции)

и (2) выполнить преобразование (1.12).

Выполнение операций (1) и (2) выполняется различными блоками передающего тракта. Операция (1) осуществляется в baseband-процессоре, а операция (2) в квадратурном (IQ) модуляторе.

Baseband-модулятор формирует низкочастотные (baseband) сигналы $I(t)$ и $Q(t)$ из закодированного информационного сигнала (последовательности нулей и единиц). Закон, по которому выполняется данное преобразование, определяет вид модуляции сигнала.

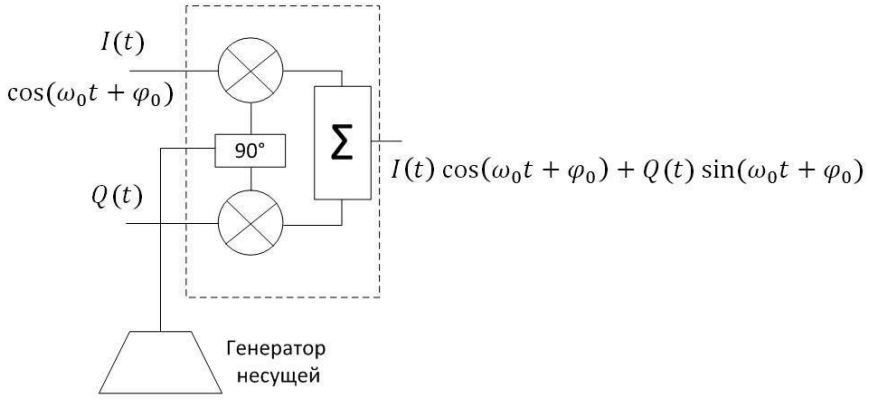


Рис. 1.3. IQ-модулятор

Для IQ-модулятора входными сигналами являются

$$I_{in}(t) = I(t); \qquad Q_{in}(t) = Q(t) \ .$$

Сформированные baseband-процессором из исходного информационного сигнала и определяющие вид и свойства модулированного сигнала. На модулятор также поступает немодулированное несущее колебание вида $\cos(\omega t)$. На выходе модулятора образуется сигнал (1.12) посредством умножения сигнала $I(t)$ на немодулированное несущее колебание, $Q(t)$ на немодулированное несущее колебание, сдвинутое по фазе на 90°, и последующего суммирования.

Видно, что структура IQ-модулятора является инвариантной относительно вида модуляции. Таким образом, осуществление того или иного вида модуляции определяется программой, выполняемой baseband-процессором, а именно алгоритмом формирования квадратурных составляющих из закодированного информационного сигнала. IQ-модулятор работает на высокой (несущей) частоте и, как правило, является аналоговым устройством. Таким образом, структурно передающий тракт можно разделить на цифровую и аналоговую части, разделенные цифро-аналоговым преобразователем (ЦАП), формирующим аналоговый сигнал из последовательности поступающих на него отсчетов.

Возможна также и иная реализация, когда IQ-модулятор выполнен в цифровом виде и преобразование (1.12) осуществляется baseband-процессором. В этом случае на выходе

baseband-процессора формируется сигнал *s(t)* на промежуточной частоте, существенно более низкой, чем несущая. Данный сигнал преобразуется в аналоговый с помощью ЦАП и затем его спектр переносится на несущую частоту с помощью смесителя.

Цифровые виды модуляции [7]

Цифровые виды модуляции (часто цифровая модуляция называется манипуляцией), как и аналоговые, могут быть амплитудными, фазовыми, частотными или комбинированными (например, амплитудно-фазовыми), в зависимости от того, какой из параметров немодулированного несущего колебания $s(t) = A(t)\cos(\omega t) + p(t)$ изменяется в соответствии с изменением информационного сигнала. Так как значения цифрового информационного сигнала являются дискретными (например, {0,1}), дискретным является также и возможный набор значений каждого из параметров. Однако если информационный сигнал проходит через baseband-фильтр для ограничения спектра, его значения уже не являются дискретными, поэтому реально переход от одного дискретного значения параметра колебания (например, изменение амплитуды или фазы) происходит гладко и непрерывно.

Амплитудные виды модуляции (OOK, ASK, M-ASK)

Наиболее простым видом манипуляции сигнала является амплитудная манипуляция. Модулированный сигнал имеет вид:

$$s(t) = A(c(t) + B)\cos(\omega t + \varphi_0),$$

где *c(t)* - информационный цифровой сигнал, *A, B и* φ_0 - постоянные, $B \geq 0$, ω - несущая частота. Пусть множество возможных значений *c(t)* {0,1}, B=0. В этом случае модулированный сигнал имеет вид $s(t) = Ac(t)\cos(\omega t + \varphi_0)$, его амплитуда принимает значение 0 при нулевом значении информационного сигнала и *A* при единичном (рис.1.2). Такой тип манипуляции называется OOK (On-Off Keying, Включено-Выключено) и часто используется в системах сигнализации и охранных системах.

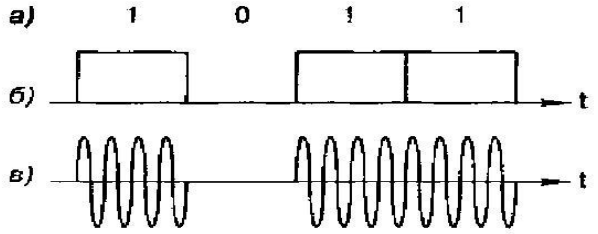


Рис. 1.4. Модуляция OOK: а - информационное сообщение; б - модулирующий цифровой сигнал; в - модулированный радиосигнал

Допустим теперь $B = 1$. В этом случае амплитуда модулированного сигнала принимает значение *A* при нулевом значении информационного сигнала и 2*A* при единичном. Вид модуляции, для которого *B* >0, носит название ASK (Amplitude Shift Keying -

амплитудная манипуляция). OOK является частным случаем ASK при *B*=0. Существует два основных критерия сравнения эффективности различных видов модуляции. Это *критерии спектральной и энергетической эффективности.* Спектральная эффективность характеризует полосу частот, необходимую для передачи информации с определенной скоростью. Энергетическая эффективность описывает мощность, необходимую для передачи информации с заданной достоверностью (вероятностью ошибки). Известно, что спектр модулированного сигнала на радиочастоте с точностью до постоянного множителя совпадает со спектром модулирующего (baseband) сигнала, однако, центр спектра радиосигнала размещен на несущей частоте, а не на нулевой. Поэтому, как правило, анализируются спектральные плотности модулирующих сигналов, центрированные относительно нулевой частоты. Спектральные плотности мощности ASK сигналов для различных baseband-фильтров приведены на рис.1.4. На рис.1.4 показаны соответствующие формы импульсов модулирующего сигнала после прохождения baseband-фильтра. Из сравнения рис.1.4 и рис.1.5 видно, что более гладкая форма импульса модулирующего сигнала приводит к расширению главного лепестка спектральной плотности мощности модулированного сигнала и более быстрому уменьшению амплитуды боковых лепестков.

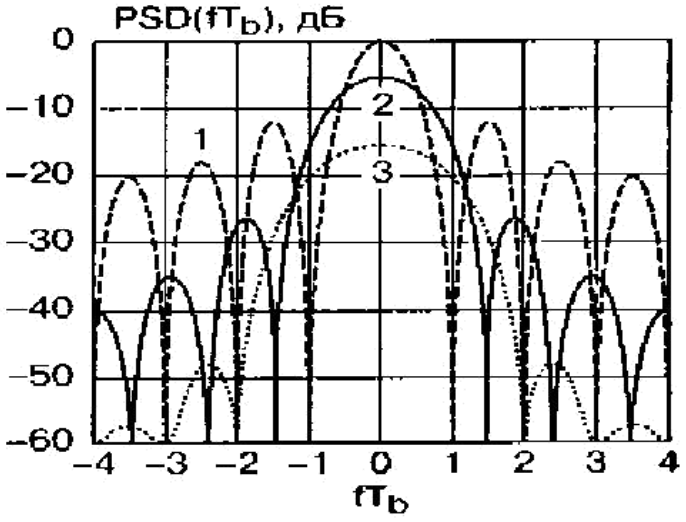


Рис. 1.5. Спектральная плотность мощности ASK-сигнала. Форма импульса модулирующего сигнала: 1 - прямоугольная, 2 - косинусоидальная, 3 - приподнятый косинус.

Множество возможных значений квадратурных компонент *I(t)* и *Q(t)* называется *сигнальным созвездием.* Как правило, данное множество отображают на декартовой плоскости, где по оси абсцисс отложены значения синфазной составляющей *I(t),* а по оси ординат - квадратурной *Q(t).* Точка на плоскости с координатами (x,y) соответствует состоянию сигнала, в котором синфазная составляющая равна x, квадратурная равна y. Таким образом, сигнальное созвездие - это диаграмма возможных состояний сигнала.

Используя общий вид модулированного радиосигнала, можно показать, что амплитуда модулированного радиосигнала в текущем состоянии равна

$$A(t) = \sqrt{I^2(t) + Q^2(t)},$$

а фаза равна углу вектора, указывающего в точку (I,Q), отсчитываемого от оси абсцисс. Для модуляций OOK и ASK сигнальное созвездие изображено на рисунке 1.6.

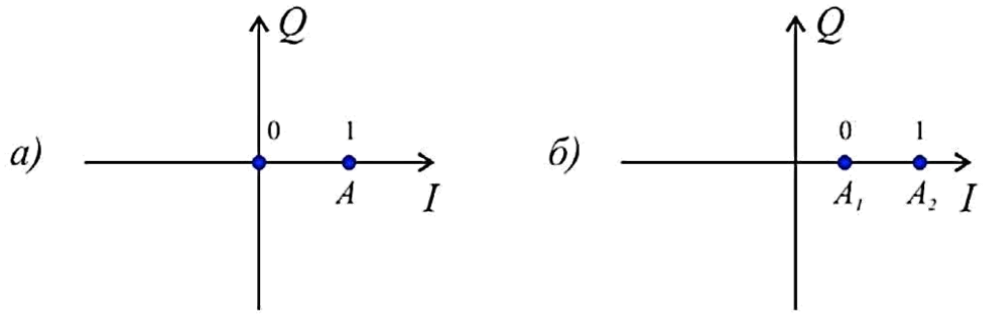


Рис. 1.6. Виды модуляции: а - сигнальное созвездие модуляции OOK, б - сигнальное созвездие модуляции ASK

Многопозиционная амплитудная модуляция (M-ASK)

При модуляции ASK множество возможных значений амплитуды радиосигнала ограничивается двумя значениями (без учета сглаживания baseband-фильтром). Спектральная эффективность может быть существенно повышена, если использовать большее количество значений амплитуды радиосигнала.

Сгруппируем биты исходного информационного сообщения в пары. Каждая такая пара называется символом. Если каждый бит имеет множество значений {0,1}, то каждый символ имеет четыре возможных значения из множества {00, 01, 10, 11}. Сопоставим каждому из возможных значений символа значение амплитуды радиосигнала из множества {0, A, 2A, 3A}. Аналогичным образом можно группировать тройки, четверки и большее количество бит в одном символе. Получится многоуровневый (многопозиционный) сигнал M-ASK с размерностью множества возможных значений амплитуды сигнала $M = \log_2 k$, где k - число бит в одном символе.

Например, сигнал с модуляцией 256-ASK имеет 256 возможных значений амплитуды сигнала и 8 бит в одном символе.

На рисунке 1.7 изображена спектральная плотность мощности восьмиуровневого сигнала 8-ASK и спектральная плотность сигнала ASK с импульсами прямоугольной формы (без baseband-фильтрации). Многопозиционный сигнал имеет меньшую ширину главного лепестка (занимает меньшую полосу частот) и более низкий уровень боковых лепестков, т. е. имеет большую спектральную эффективность по сравнению с двухуровневым сигналом.

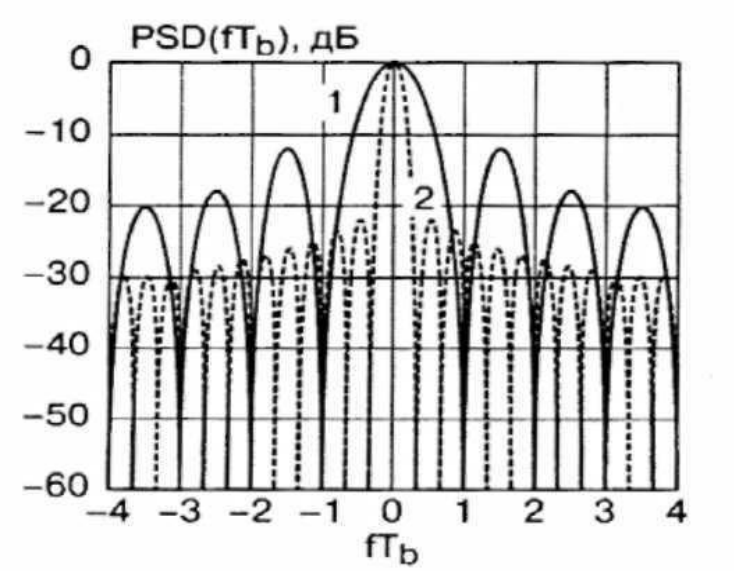


Рис. 1.7. Сравнение спектральных плотностей мощности двухуровневого и восьмиуровневого АМ-сигналов: 1 - сигнал ASK, 2 – сигнал 8-ASK

Амплитудные виды модуляции имеют невысокую энергетическую эффективность (так как средний уровень мощности существенно меньше максимального), требуют высокой линейности и большого динамического диапазона усилителя мощности. Ошибка в амплитуде сигнала из-за нелинейности усилителя приведет непосредственно к символьной ошибке, т. к. значение символа определяется амплитудой сигнала. Отношение максимальной амплитуды сигнала к минимальной достаточно высоко и требует усилителя с большим динамическим диапазоном. Влияние аддитивного шума или помехи непосредственно изменяет амплитуду сигнала, поэтому амплитудные виды модуляции не обладают высокой помехоустойчивостью. Однако они достаточно просты в реализации. Сигнальное созвездие для 8-ASK приведено на рисунок 1.8.

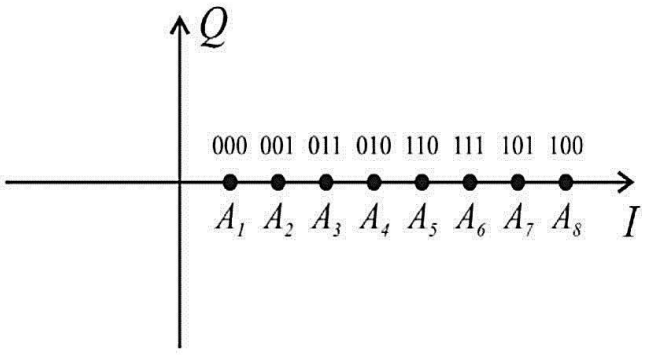


Рис. 1.8. Сигнальное созвездие модуляции 8-ASK

Фазовые виды модуляции (BPSK, QPSK, M-PSK, QAM)

Фазомодулированный сигнал имеет вид:

$$s(t) = A\cos(\omega t + \varphi(t) + \varphi_0), \quad (1.13)$$

где A и φ_0 - постоянные, ω_0 - несущая частота.

Информация кодируется фазой $\varphi(t)$. Так как при когерентной демодуляции в приемнике имеется восстановленная несущая $S_C(t) = A\cos(\omega t + \varphi_0)$, то путем сравнения сигнала с несущей вычисляется текущий сдвиг фазы $\varphi(t)$. Изменение фазы $\varphi(t)$ взаимнооднозначно связано с информационным сигналом $c(t)$.

QPSK (Quadrature Phase Shift Keying) - Квадратурная фазовая модуляция (четырёхуровневая фазовая модуляция). В данной фазовой модуляции используются четыре значения фазы несущего колебания. В этом случае фаза сигнала должна принимать четыре значения: 0°, 90°, 180° и 270°. Однако чаще используются другие значения фаз: 45°, 135°, 225° и 315° с шагом, кратным π / 2.

Соотношение между сдвигом фазы модулированного колебания из множества {±π/4, ±3π/4} и множеством символов (дибитов) цифрового сообщения {00, 01, 10, 11} устанавливается в каждом конкретном случае стандартом на радиоканал и отображается сигнальным созвездием. Сигнальное созвездие – это представление манипулированных радиосигналов на комплексной плоскости.

Модулированный сигнал можно представить как вектор на графике в полярной системе координат; длина вектора соответствует амплитуде сигнала, а его ориентация — фазе. На рисунке 1.9 представлено сигнальное созвездие для QPSK модуляции.

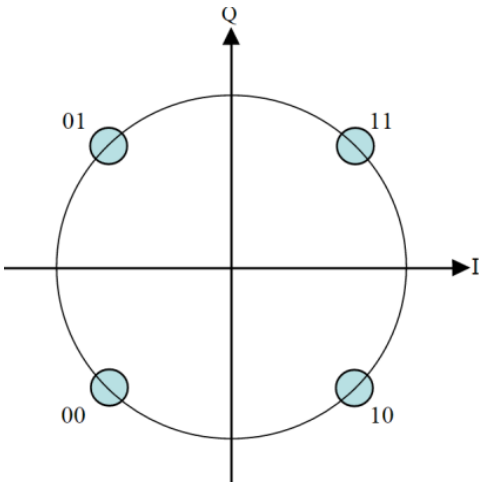


Рис. 1.9. Сигнальное созвездие для QPSK модуляции.

Фаза может переходить из одного состояния в другое. Такие переходы представлены на рисунке 1.10 чёрными стрелками.

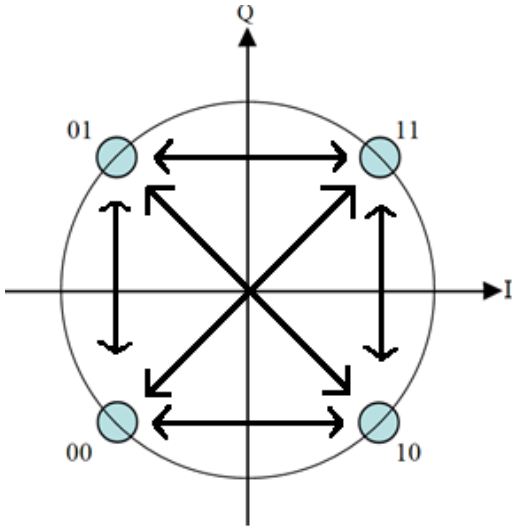


Рис. 1.10. Переходы из одного фазового состояния в другое

Из теории связи известно, что наивысшей помехоустойчивостью обладает двоичная фазовая модуляция BPSK. Однако в ряде случаев за счет уменьшения помехоустойчивости канала связи можно увеличить его пропускную способность.

Квадратурно-амплитудная модуляция - QAM (QAM – Quadrature Amplitude Modulation) может рассматриваться как расширенная многоуровневая ФМ, в которой два исходных сигнала генерируются независимо. Таким образом, здесь имеют место два полностью независимых квадратурных канала, включающие процессы кодирования и детектирования в основной полосе.

На рисунке 1.11 показано сигнально - точеное пространство для системы с 16-QAM и четырьмя уровнями в каждом квадратурном канале. Точки представляют составной сигнал, а штрихи на осях отмечают уровни амплитуды в каждом квадратурном канале.

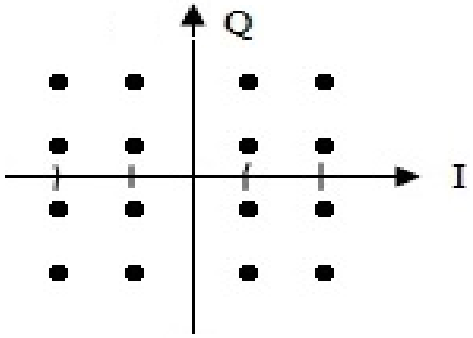


Рис. 1.11. Сигнально-точечное пространство модуляции для 16-QAM

В отличие от ФМ сигналов сигналы QAM не содержат постоянной огибающей. Наличие постоянной огибающей в ФМ объясняется поддержанием отношения уровней в квадратурных каналах. В QAM такие ограничения не вводятся ввиду того, что в каждом канале уровни независимы.

Характеристики ошибок систем QAM и ФМ модуляций сильно отличаются. При достаточно большом числе сигнальных точек системы QAM имеют, как правило, лучшие характеристики, чем системы с ФМ. Основная причина состоит в том, что расстояние между сигнальными точками на диаграмме для системы с QAM больше, чем для соответствующей системы с ФМ.

QAM имеет преимущество над системой ФМ при той же пиковой мощности.

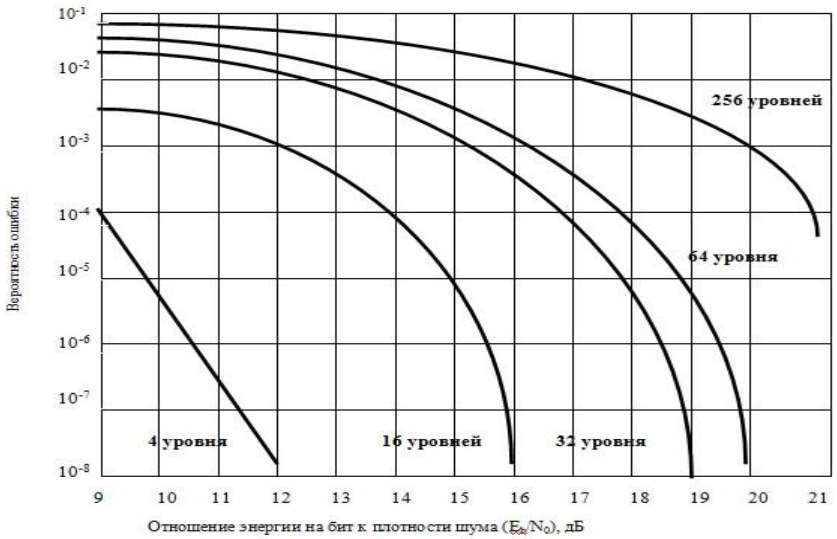


Рис. 1.12. Вероятности ошибок в системах QAM

Достоинство высоких значений номера QAM – это повышенная скорость передачи данных, поскольку таким образом большее количество битов информации может быть передано в течении одного цикла. Однако, с другой стороны, в этом случае большее число уровней амплитуды сигнала располагаются близко друг к другу, повышая тем самым вероятность неразличимости двух уровней, и как следствие – повышая чувствительность системы к шуму.

На рисунке 1.13 представлено отношение параметра SNR к параметру BER (Bit Error Rate – Отношение Бит/Ошибка).

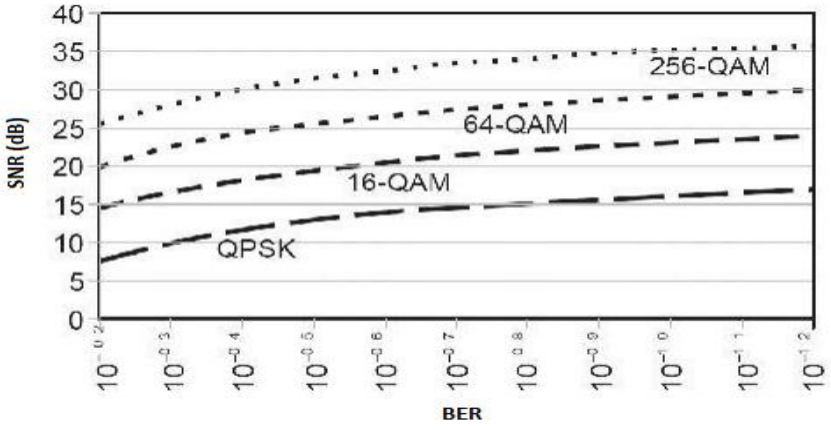


Рис. 1.13. Отношение параметра SNR к параметру BER

Двоичная фазовая модуляция (BPSK - Binary Phase Shift Keying)

Множеству значений информационного сигнала {0,1} ставится в однозначное соответствие множество изменений фазы {0, π}. При изменении значения информационного сигнала фаза радиосигнала изменяется на 180°. Временная форма сигнала и его созвездие показаны на рисунке 1.14.

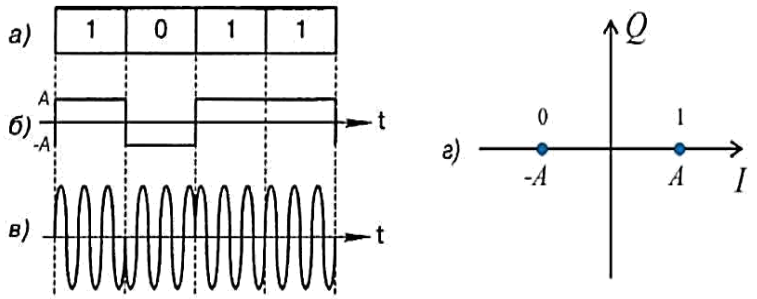


Рис. 1.14. Временная форма и сигнальное созвездие сигнала BPSK: а – цифровое сообщение; б - модулирующий сигнал; в - модулированное ВЧ-колебание; г - сигнальное созвездие

Квадратурная фазовая модуляция (QPSK - Quadrature Phase Shift Keying)

Квадратурная фазовая модуляция является четырехуровневой фазовой модуляцией (M=4), при которой фаза высокочастотного колебания может принимать 4 различных значения с шагом, кратным $\pi / 2$.

Соотношение между сдвигом фазы модулированного колебания из множества {±π/4,±3π/4} и множеством символов (дибитов) цифрового сообщения {00, 01, 10, 11} устанавливается в каждом конкретном случае стандартом на радиоканал и отображается сигнальным созвездием, аналогичным рисунке 1.15. Стрелками показаны возможные переходы из одного фазового состояния в другое.

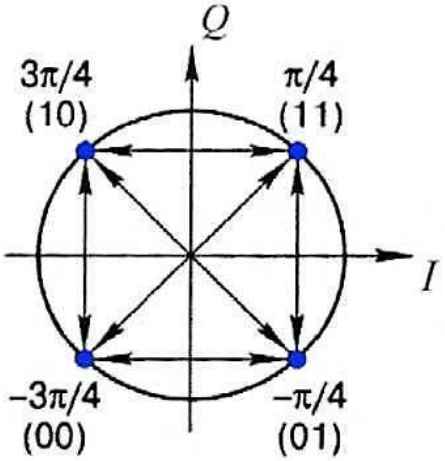


Рис. 1.15. Сигнальное созвездие модуляции QPSK

Из рисунка видно, что соответствие между значениями символов и фазой сигнала установлено таким образом, что в соседних точках сигнального созвездия значения соответствующих символов отличаются лишь в одном бите.

При передаче в условиях шума наиболее вероятной ошибкой будет определение фазы соседней точки созвездия. При указанном кодировании, несмотря на то, что произошла ошибка в определении значения символа, это будет соответствовать ошибке в одном (а не двух) бите информации. Таким образом, достигается снижение вероятности ошибки на бит. Указанный способ кодирования называется кодом Грея.

Каждому значению фазы модулированного сигнала соответствует 2 бита информации, и поэтому изменение модулирующего сигнала при QPSK - модуляции происходит в 2 раза реже, чем при BPSK-модуляции при одинаковой скорости передачи информации. Известно [5], что спектральная плотность мощности многоуровневого сигнала совпадает со спектральной плотностью мощности бинарного сигнала при замене битового интервала T_b на символьный интервал $T_s = T_b \log_2 M$. Для четырехуровневой модуляции $M=4$ и, следовательно, $T_s = 2T_b$. Расстояние между первыми нулями спектральной плотности мощности сигнала QPSK равно $Af = 1/ T_b$, что в 2 раза меньше, чем для сигнала BPSK. Другими словами, спектральная эффективность квадратурной модуляции QPSK в 2 раза выше, чем бинарной модуляции BPSK.

Многопозиционная фазовая модуляция (M-PSK)

M-PSK формируется, как и другие многопозиционные виды модуляции, путем группировки $k = \log_2 M$ бит в символы и введением взаимно-однозначного соответствия между множеством значений символа и множеством значений сдвига фазы модулированного колебания. Значения сдвига фазы из множества отличаются на одинаковую величину. Для примера на рисунке 1.16 приведено сигнальное созвездие для 8-PSK с кодированием Грея.

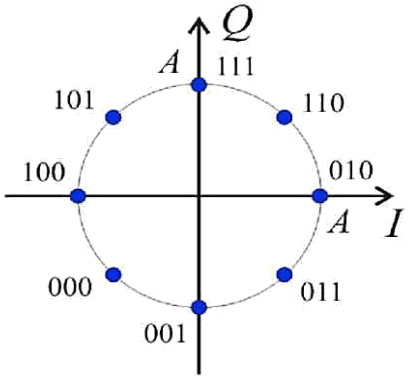


Рис. 1.16. Сигнальное созвездие модуляции 8-PSK

Амплитудно-фазовые виды модуляции (QAM)

Модуляция, при которой происходит одновременное изменение двух параметров

несущего колебания - амплитуды и фазы -называется амплитудно-фазовой модуляцией.

Минимальный уровень символьных ошибок будет достигнут в случае, если расстояние между соседними точками в сигнальном созвездии будет одинаковым, т. е. распределение точек в созвездии будет равномерным на плоскости. Следовательно, сигнальное созвездие должно иметь решетчатый вид. Модуляция с подобным видом сигнального созвездия называется квадратурной амплитудной модуляцией (QAM - Quadrature Amplitude Modulation). QAM является многопозиционной модуляцией. При M=4 она соответствует QPSK, поэтому формально считается для QAM $M > 8$ (т.к. число бит на символ $k = \log_2 M$, $k \; E \; N$, то M может принимать только значения степеней 2: 2, 4, 8, 16 и т.д.). Для примера на рисунке 1.17 приведено сигнальное созвездие 16-QAM с кодированием Грея.

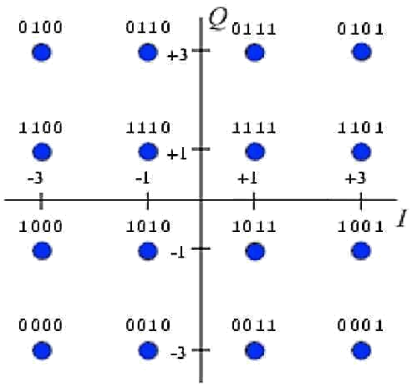


Рис. 1.17. Сигнальное созвездие модуляции 16-QAM

На практике используются большие значения M, вплоть до 1024-QAM. Такие виды модуляции позволяют достичь исключительно высокой спектральной эффективности. Однако, как видно из сигнального созвездия, так как информация кодируется в том числе амплитудой и изменения амплитуды велики, то QAM предъявляет высокие требования к линейности усилителя мощности и его динамическому диапазону, особенно для больших M. Практическое осуществление QAM-модуляции выполняется следующим образом. В памяти процессора хранится таблица значений квадратурных компонент *I(t)* и *Q(t),* имеющихся в сигнальном созвездии и расположенных в порядке возрастания значения соответствующего символа. Процессор анализирует входную последовательность битов, разбивает ее на символы и для каждого символа выбирает соответствующие значения квадратурных компонент из таблицы. Затем выполняется baseband-фильтрация сигналов *I(t)* и *Q(t).*

Сравнение различных видов модуляции

Как указывалось, основными критериями эффективности различных видов модуляции являются критерии спектральной и энергетической эффективности. Энергетическая эффективность характеризует энергию, которую необходимо затратить для передачи информации с заданной достоверностью (вероятностью ошибки). Спектральная

эффективность характеризует полосу частот, необходимую для того, чтобы передавать информацию с определенной скоростью. Кроме данных критериев, виды модуляции сравниваются по устойчивости к различным типам помех и искажений и сложности аппаратной реализации. Увеличение позиций (уровней) модуляции (модуляции M-ASK, M-PSK и M-QAM) увеличивает спектральную эффективность в $k = \log_2 M$ раз. Также отмечено, что наибольшей спектральной эффективностью среди частотных видов модуляции обладает модуляция MSK. MSK является спектрально в 2.6 раза менее эффективной, чем QPSK и в 1.3 раза менее эффективной, чем BPSK. Сравним виды модуляции по критерию энергетической эффективности. Для этого оценим для каждого вида модуляции требуемую энергию для передачи информации с одинаковой вероятностью ошибки на бит. В таблице приводятся зависимости вероятности ошибки на бит от отношения E_b / N_0 для различных видов модуляции.

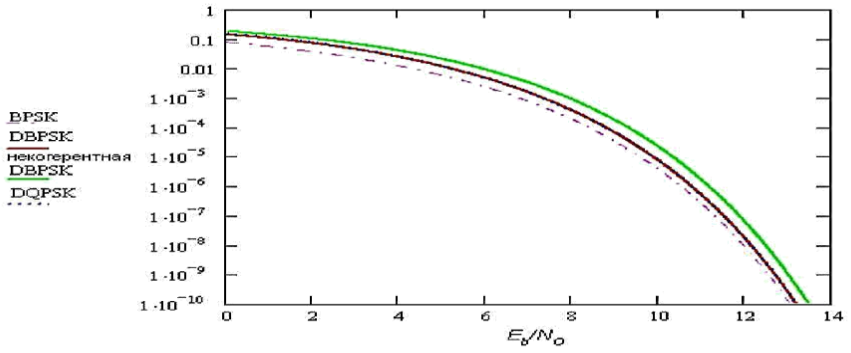


Рис. 1.18. Сравнение энергетической эффективности модуляций с относительным кодированием: DBPSK, DQPSK и некогерентной DBPSK.

Однако BER для BPSK и QPSK описываются одинаковыми формулами, при этом QPSK в 2 раза спектрально эффективнее, чем BPSK. Следовательно, QPSK всегда существенно эффективнее, чем BPSK, и, обыкновенно, имеет смысл использовать QPSK, а не BPSK.

Сравним виды модуляции по критерию энергетической эффективности. Для этого оценим для каждого вида модуляции требуемую энергию для передачи информации с одинаковой вероятностью ошибки на бит. В [2], [10] определены соотношения, связывающие вероятность битовой ошибки с величиной E_b / N_0 для различных видов модуляции:

$$BER = f\left(\frac{E_b}{N_0}\right), \tag{1.14}$$

где BER - вероятность ошибки E_b - энергия, необходимая для передачи одного бита информации, N_0 - спектральная плотность мощности белого шума в канале. Если мощность

передатчика равна P, то величина энергии, приходящаяся на один бит информации, равна $E_b = PT_b$, где T_b - длительность бита. В табл. 1.1 приводятся зависимости вероятности ошибки на бит от отношения E_b / N_0 для различных видов модуляции [1].

Таблица 1.1. Вероятность ошибки для различных видов модулялии

Вид модуляции	Вероятность ошибки на бит (BER)
OOK	$Q(\sqrt{E_b / N_0})$
M-ASK код Грея	$\frac{2(M-1)}{M\log_2 M} Q\left(\sqrt{\frac{\log_2 M}{(M-1)^2}\frac{E_b}{N_0}}\right)$
BPSK	$Q(\sqrt{2E_b / N_0})$
Некогерентная DBPSK	$\frac{1}{2}\exp(-E_b / N_0)$
Когерентная DBPSK	$2Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\left(1-Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\right)$
QPSK код Грея	$Q(\sqrt{2E_b / N_0})$
Когерентная DQPSK при $E_b / N_0 >> 1$	$2\,Q(\sqrt{2E_b / N_0})$
M-PSK код Грея	$\frac{2}{\log_2 M} Q\left(\sqrt{\frac{2E_b \log_2 M}{N_0}\sin^2\left(\frac{\pi}{M}\right)}\right)$
FSK	$Q\left(\sqrt{\left[1-\frac{\sin(2\pi\cdot m)}{2\pi\cdot m}\right]\frac{E_b}{N_0}}\right)$
MSK	$Q(\sqrt{E_b / N_0})$
M-MSK	$\frac{2(M-1)}{M\log_2 M} Q\left\{\sqrt{\log_2 M \frac{E_b}{N_0}}\right\}$
QAM код Грея	для $k = \log_2 M$, k – четное: $BER = \frac{2P_0 - P_0^{\,2}}{\log_2 M}$, где $P_0 = \frac{2(\sqrt{M}-1)}{\sqrt{M}} Q\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{E_b}{N_0}}\right)$ для нечетных k: $BER \le \frac{1}{\log_2 M}\left[1-\left(1-2Q\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{E_b}{N_0}}\right)\right)^2\right]$

В табл. $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt$ - интеграл ошибок (интеграл вероятности),

M - число позиций для многопозиционных видов модуляции, *m* - индекс модуляции для частотной модуляции, *BER* - вероятность ошибки на бит.

Из табл. видно, что с увеличением позиционности модуляции, вероятность битовой ошибки увеличивается (см., например, формулы M-ASK и M-PSK, *Q(x)* является убывающей функцией аргумента). Таким образом, как правило, *при увеличении спектральной эффективности энергетическая эффективность уменьшается.*

Однако BER для BPSK и QPSK описываются одинаковыми формулами, при этом QPSK в 2 раза спектрально эффективнее, чем BPSK. Следовательно, *QPSK всегда существенно эффективнее, чем BPSK, и, обыкновенно, имеет смысл использовать QPSK, а не BPSK.* Физически, это объясняется тем, что в случае QPSK добавляется дополнительная степень свободы: квадратурная составляющая *Q(t).* В случае BPSK используется только синфазная составляющая *I(t).* Квадратурная форма когерентного фазового демодулятора приводит к тому, что два канала детектора обеспечивают *независимый* прием двух бинарных фазомодулированных сигналов. Аналогичное явление имеет место и при сравнении модуляций DBPSK и DQPSK (с относительным кодированием). Хотя выражения для BER несколько отличаются, с высокой степенью приближения они совпадают (рис.24). Модуляции с относительным кодированием имеют небольшой энергетический проигрыш по сравнению с обыкновенными BPSK и QPSK (0.3 - 0.9 дБ). DBPSK с некогерентным детектированием также имеет небольшой проигрыш по сравнению с DBPSK с когерентным детектированием (около 0.5 дБ), рис. 1.20. Под энергетическим выигрышем понимается разница в значении E_b / N_0 при одинаковом значении вероятности ошибки на бит.

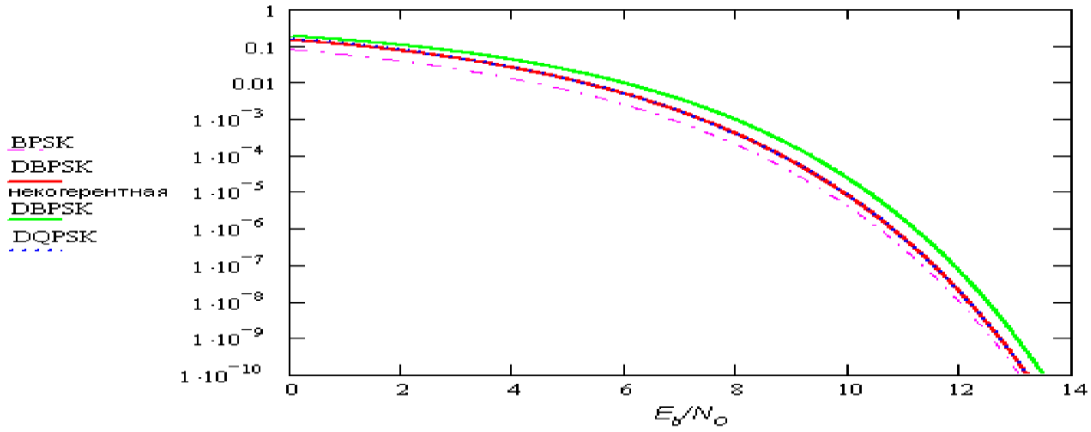


Рис. 1.20. Сравнение энергетической эффективности модуляций BPSK, DBPSK, DBPSK и DQPSK

Таким образом, имеет смысл сравнивать виды модуляции с одинаковым числом позиций. Сравним двухуровневые OOK, BPSK и MSK. Как видно из табл., OOK и MSK

имеют одинаковую эффективность и уступают BPSK (и, соответственно, QPSK) по энергетической эффективности приблизительно 3 дБ (рис.1.21).

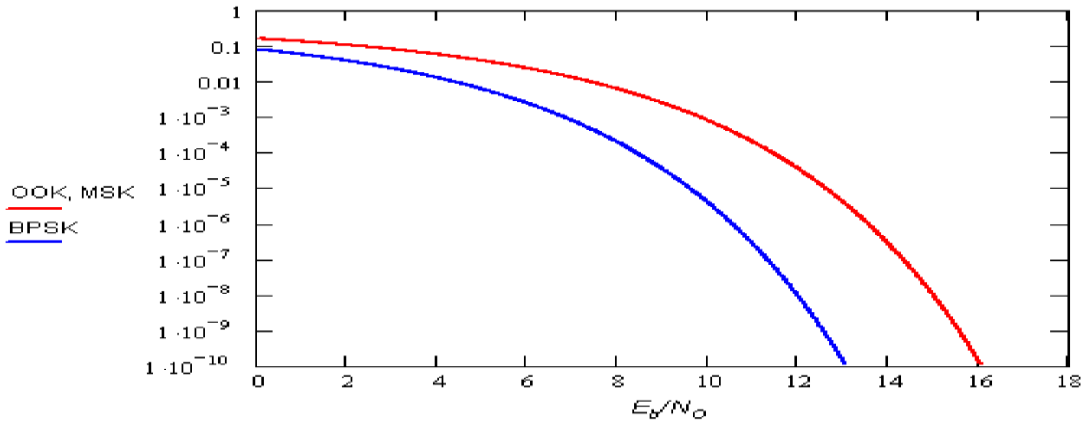


Рис. 1.21. Сравнение энергетической эффективности модуляций OOK, MSK и BPSK

По результатам данного сравнения можно сделать вывод о том, что при числе уровней до 4 включительно QPSK является спектрально и энергетически наиболее эффективным видом модуляции. Однако здесь следует сделать одно существенное замечание относительно модуляции GMSK. Ее спектральная эффективность ниже, чем QPSK, в системах с *линейным усилением.* GMSK, как частотный вид модуляции, позволяет использовать высокоэффективные нелинейные усилители и ограничители, что дает энергетический выигрыш. При прохождении QPSK через подобные устройства, ее спектр расширяется (происходит некоторое восстановление боковых лепестков). Поэтому, в некоторых случаях, GMSK может иметь большую эффективность, чем QPSK. В частности в стандарте GSM выбор сделан в пользу GMSK, а в CDMA - OQPSK. Однако, усовершенствованные виды модуляции QPSK (например, FQPSK) в любом случае превосходят GMSK.

Сравним теперь модуляции с числом уровней *M>4.* На рис. 1.22 изображено сравнение энергетической эффективности для амплитудной, фазовой и амплитудно-фазовой манипуляции при *M=16* и *M=64.*

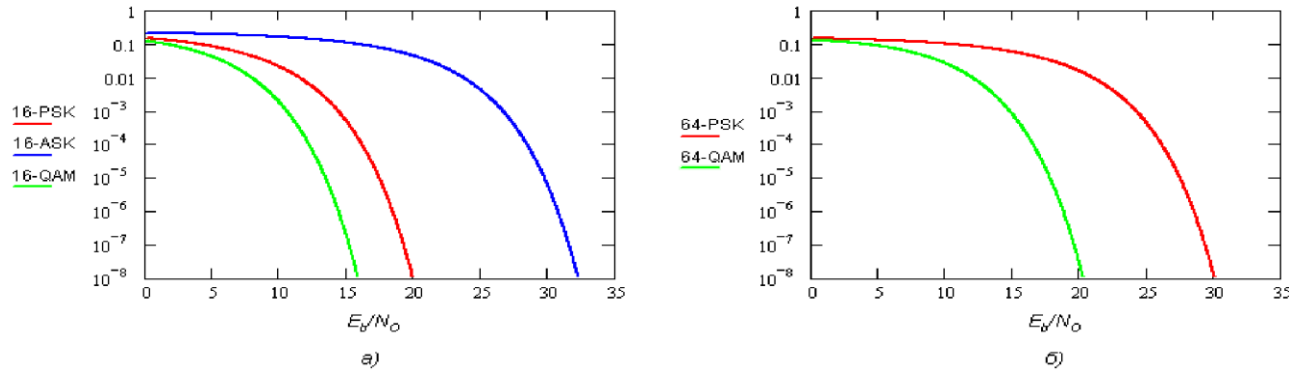


Рис. 1.22. Сравнение энергетической эффективности модуляций M-ASK, M-PSK и M-QAM: а - M=16, б - M=64

Как видно из рис. 1.22, амплитудная модуляция существенно (более 10 дБ при M=16) уступает фазовой и амплитудно-фазовой, поэтому при *M=64* сравнение с ней не проводится. При сравнении M-PSK с M-QAM видно, что M-QAM превосходит по эффективности M-PSK, причем энергетический выигрыш M-QAM увеличивается с ростом *M*. Например, для *M=16* выигрыш составляет около 4 дБ, а при M=64 около 10 дБ. Физически это объясняется тем, что расстояние между соседними точками в сигнальном созвездии M-PSK меньше, чем M-QAM. Сигнальное созвездие M-PSK представляет собой окружность с равномерно распределенными на ней точками, а созвездие M-QAM - квадрат с равномерно распределенными по его площади точками. Чем больше расстояние между точками в созвездии, тем менее вероятна ошибка в детектировании соседнего символа. Многопозиционная частотная модуляция используется гораздо реже, так как *при увеличении числа уровней и сохранении индекса модуляции ее спектр не сужается, а расширяется,* ввиду того, что вводятся новые частоты и ширина спектра растет по закону $\frac{M}{\log_2 M}$, Как видно из табл. 1.1, однако, *при увеличении числа уровней M-MSK, в отличие от всех других видов модуляции, вероятность ошибки на бит уменьшается.* Мы получаем выигрыш в энергетической эффективности за счет уменьшения спектральной эффективности.

Таким образом, при ограниченной полосе, при $M < 4$ наиболее эффективной является модуляция QPSK, а при $M > 4$ - QAM. QPSK является частным случаем QAM при M=4. Можно считать QAM наиболее эффективным видом модуляции при любом числе уровней. Еще больший выигрыш по сравнению с обыкновенными QPSK и QAM дают их усовершенствованные модификации, такие, как модификации Феера [11] (FQPSK, FQAM), модуляция с решетчатым кодированием (TCM), оптимизация формы сигнальных созвездий и использование многомерных сигнальных созвездий.

1.2. Модемы сотовой связи FSK, MSK GMSK и численный анализ вероятности символьной ошибки с использованием ПО LabVIEW

Частотная манипуляция (FSK)

Значениям «0» и «1» информационной последовательности соответствуют определённые частоты синусоидального сигнала при неизменной амплитуде. Частотная манипуляция весьма помехоустойчива, поскольку помехи телефонного канала искажают в основном амплитуду, а не частоту сигнала. Однако при частотной манипуляции неэкономно расходуется ресурс полосы частот телефонного канала. Поэтому этот вид модуляции

применяется в низкоскоростных протоколах, позволяющих осуществлять связь по каналам с низким отношением сигнал/шум.

Существует также подвид этой модуляции GFSK. Принцип работы модулятора GFSK похож на FSK, за исключением того, что сначала полоса импульсов (-1, 1) проходит через фильтр Гаусса для сглаживания, что обеспечивает уменьшения ширины его спектра, а уже после попадает в FSK. Фильтрация Гаусса — один из самых распространенных способов уменьшения ширины спектра.

На рисунке ниже приведен график двоичной бинарной последовательности нулей и единиц и, соответствующий ему, график частотно-манипулированного сигнала. Низкому уровню бинарного двоичного сигнала соответствует частота 1 КГц, а высокому - частота 0,5 КГц несущего сигнала синусоидального типа.

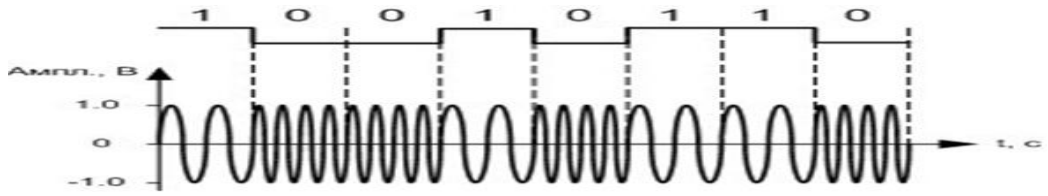


Рис. 1.23. Частотная манипуляция

В программной комплексе можно осуществить многопозиционную частотную модуляцию (MFSK) задав в поле M-FSK нужный уровень от 2 до 64. В бинарной FSK модуляции, т.е. при *M=2*, два двоичных числа представляются сигналами двух различных частот, расположенных около несущей. Бинарная частотная модуляция менее восприимчива к ошибкам, чем амплитудная модуляция.

Более эффективной, но и более подверженной ошибкам, является схема многочастотной модуляции (Multiple FSK - MFSK), в которой используется более двух частот. В этом случае каждая сигнальная посылка представляет более одного бита. Переданный сигнал MFSK (для одного периода передачи сигнальной посылки) можно определить следующим образом:

$$s_i = A\cos(2\pi f_i t), \quad 1 << i << M$$

Здесь

$$f_i = f_c + (2i - 1 - M) f_d,$$

где f_c - несущая частота; f_d - разностная частота; M - число различных сигнальных посылок $= 2^L$; L - количество битов на одну сигнальную посылку.

На рис. 1.24 представлен пример схемы MFSK с M=4. Входной поток битов кодируется по два бита, после чего передается одна из четырех возможных двухбитовых комбинаций.

Для уменьшения занимаемой полосы частот в модуляторах сигналов с фазовой модуляцией применяют сглаживающие фильтры. Применение сглаживающих фильтров приводит к увеличению эффективности использования полосы, но в то же время из-за сглаживания уменьшается расстояние между соседними сигналами, что приводит к снижению помехоустойчивости.

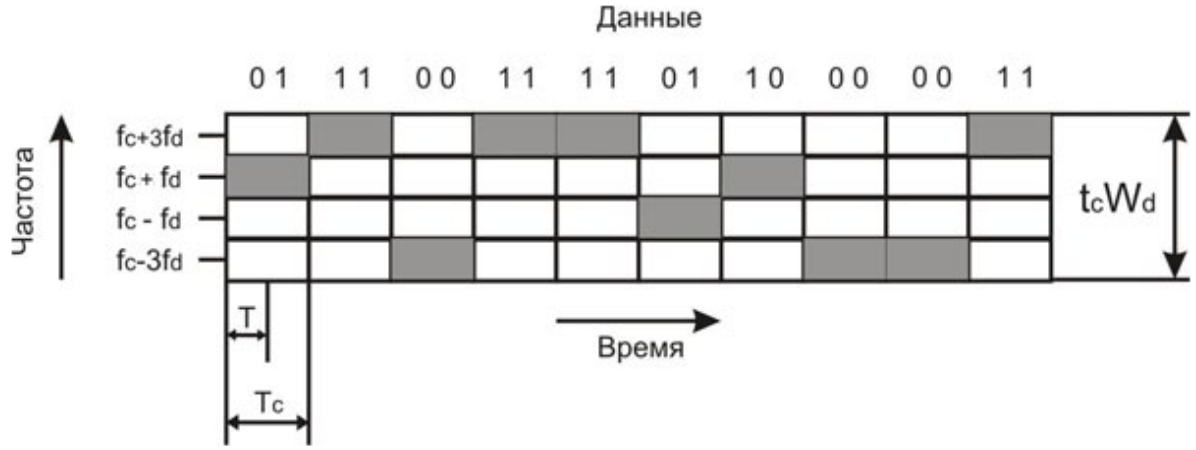


Рис. 1.24. Использование частоты схемой MFSK (M = 4)

Построение графиков глазковой диаграммы и ее вид дает много информации о джиттере сигнала, так же, как и о многих других его параметрах. Например, множество отдельных фронтов и спадов говорит о вероятном присутствии джиттера, зависящего от данных.

Глазковая диаграмма не просто предоставляет множество информации, она удобна простотой применения и тем, что может применяться для измерений в любой цепи с реальными данными.

Рис. 1.25. Параметры глазковой диаграммы

По индикаторной диаграмме можно выполнить ряд важных измерений:

• чем больше открыт глазок, тем легче различать логические 1 и 0;

• ширина открытия глазковой диаграммы (время между пересечениями линий логической 1 с логическим 0 и логического 0, с логической 1) показывает временной интервал, в течение которого сигнал может быть замерен без ошибки из-за межсимвольного влияния;

• псевдослучайной последовательности битов и отображения сигналов на запоминающем осциллографе получается структура, которая называется индикаторной (глазковой) диаграммой (eye diagram). Типичная индикаторная диаграмма приведена на рис. 3.

• высота открытия глазка измеряет запас помехоустойчивости на выходе приемника;

• ширина линий глазка к точках пересечения в углах глазка является мерой флуктуации в системе передачи. Флуктуации вызываются разбросом времени включения и выключения лазера; искажением импульса оптическим волокном и шумом. Флуктуации выражаются в пикосекундах, градусах или в процентах интервала бита;

• толщина линий импульса наверху и внизу глазка пропорциональна шуму и искажениям в системе передачи; время перехода сигнала в схеме глазка с верхнего уровня (логического 0) в нижний (логическая 1) и наоборот указывает времена подъема и спада системы передачи. Они обычно замеряются между отметками 10 и 90%;

• времена подъема и спада важны для оценки чувствительности системы к синхронизации замеров (sample timing). Чем больше времена подъема и спада сигнала, тем более чувствительна система к ошибкам синхронизации;

• чтобы обеспечить Системе максимальную невосприимчивость к шуму, лучшим временем для замеров уровня сигнала является время, когда высота открытия индикаторной диаграммы максимальна.

Джиттер или фазовое дрожание цифрового сигнала данных — нежелательные фазовые и/или частотные случайные отклонения передаваемого сигнала. Возникают вследствие нестабильности задающего генератора, изменений параметров линии передачи во времени и различной скорости распространения частотных составляющих одного и того же сигнала.

В цифровых системах проявляется в виде случайных быстрых изменений местоположения фронтов цифрового сигнала во времени, что приводит к рассинхронизации и, как следствие, искажению передаваемой информации. Например, если фронт имеет малую крутизну или «отстал» по времени, то цифровой сигнал как бы запаздывает, сдвигается относительно значащего момента времени — момента времени, в который происходит оценка сигнала.

Джиттер является одной из основных проблем при проектировании устройств цифровой электроники, в частности, цифровых интерфейсов. Недостаточно аккуратный расчет джиттера может привести к его накоплению при прохождении цифрового сигнала по тракту и, в конечном счёте, к неработоспособности устройства.

Причины возникновения джиттера:

- Фазовые шумы петли ФАПЧ (фазовой автоподстройки частоты) устройства, синхронизируемого внешним сигналом. Джиттер, вызываемый ФАПЧ, проявляется при прослушивании материала с записывающего устройства, синхронизируемого от воспроизводящего устройства.
- АЦП. В современных цифровых системах звукозаписи и воспроизведения основным источником джиттера является АЦП. Нынешние полностью цифровые студийные синхронизаторы достаточно совершенны и часто вносят джиттер меньший, чем АЦП.

Описание прибора с FSK модуляцией (манипуляцией)

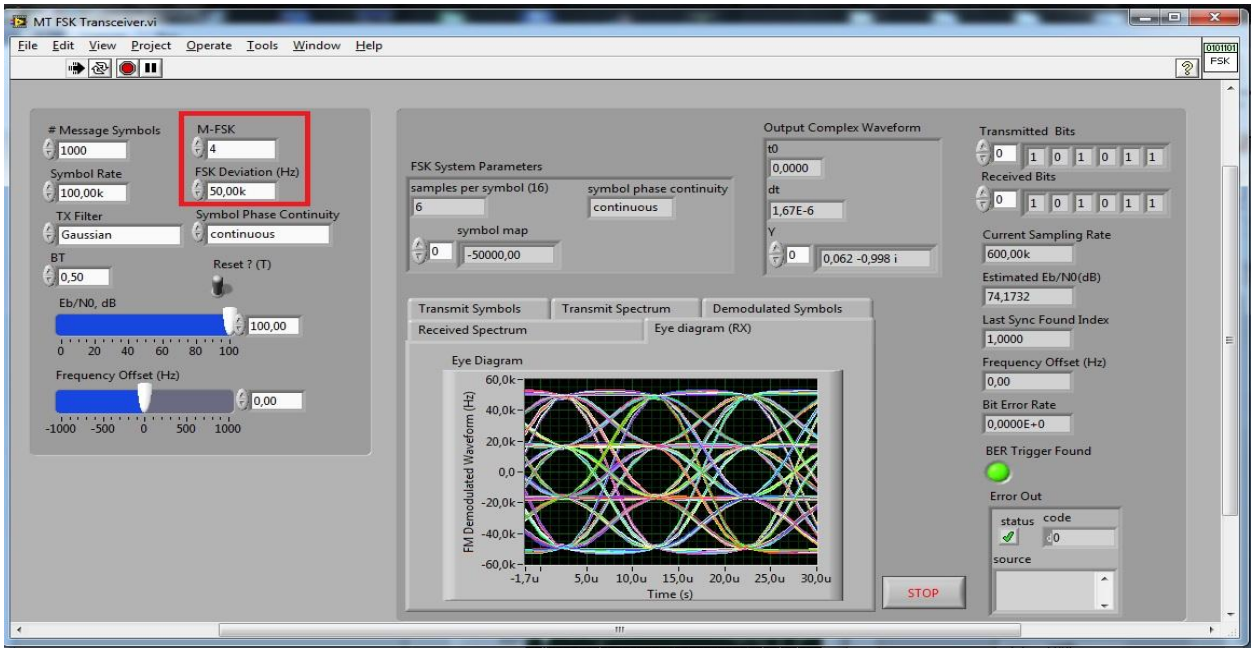


Рис. 1.26. Интерфейс прибора с FSK модуляцией

- Message Symbols – показывает количество передаваемых символов.
- M-FSK – показывает уровень модуляции. Доступны режимы: 2,4,8,16,32,64
- FSK Deviation (Гц) – отклонение по частоте частотной модуляции.
- Eb/N0- отношение сигнал/шум
- Frequency offset –задержка по частоте
- TX Filter – выбор фильтра для передатчика. Доступны следующие: Gaussian, Raised Cosine, Root Raised Cosine

- Symbol Rate – порядок символов.
- Symbol Phase Continuity –продолжительность фазы импульса. Соответственно режимы Continuous(Конечная), Discontinuous (Бесконечная).
- Transmitted Bits – переданные биты.
- Received Bits – принятые биты.
- BER Trigger Found – Вероятность шибки
- BT - безразмерная величина равная $BT = B_{-3дБ} * T$, где $B_{-3дБ}$ - полоса фильтра Гаусса по уровню -3дБ, $T = 1/B_r$ - длительность единичного импульса цифровой информации, передаваемой со скоростью Br бит\с. Например $B_r = 20кбит/с,$ тогда *T=50мс* и при полосе фильтра Гаусса по уровню -3 дБ $B_{-3дБ} = 10\,кГц$ получаем $BT = 10кГц * 50мкс = 0.5$. Таким образом, параметр BT показывает во сколько раз полоса фильтра Гаусса $B_{-3дБ}$ отличается от скорости передачи информации B_r, выраженной в единицах измерения частоты.
- Сокращения: TX – Передатчик, RX – Приемник

Исследование линии передачи с FSK модуляцией

Предварительная фильтрация

Сравним спектры сигналов и их помехоустойчивость при использовании фильтра Гаусса и без фильтра

Без фильтра при соотношении сигнал/шум = 80

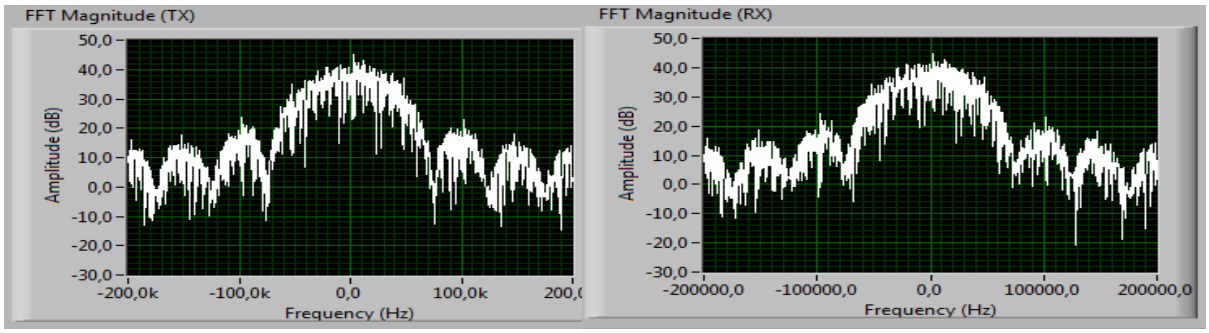


Рис. 1.27. Спектр переданного (слева) и принимаемого (справа) сигнала

Как видно из рисунка 1.27, спектр почти не изменился, но в нем присутствует высокий уровень боковых лепестков, что может повлиять на соседние каналы, также придется выделить более широкий канал в линии передачи, что не всегда является позволительным и выгодным.

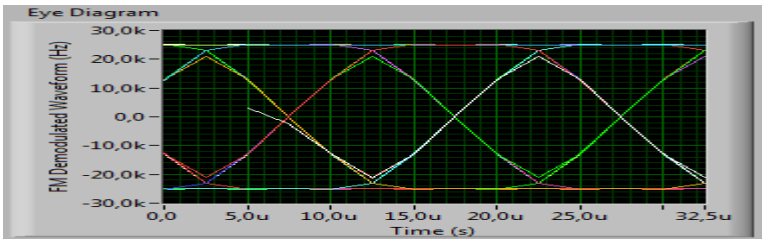


Рис. 1.28. Глазковая диаграмма

Без использования фильтра в сигнал не вносятся дополнительные искажения следовательно джиттер и помехи минимальны.

Применение фильтра Гаусса. При соотношении сигнал/шум = 80, $BT = 0{,}5$

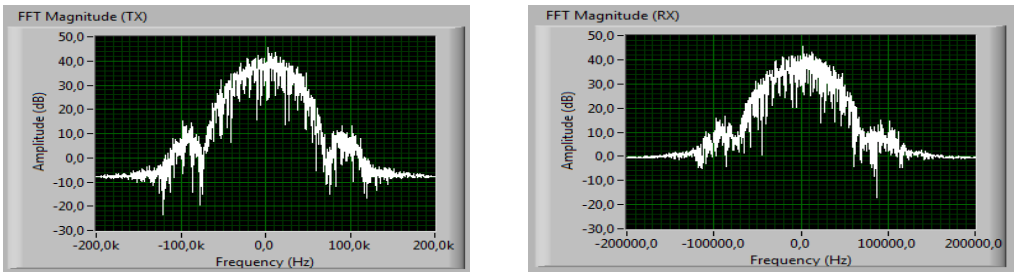


Рис. 1.29. Спектр переданного (слева) и принимаемого (справа) сигнала

Сужение полосы занимаемых частот удалось достигнуть за счет предварительной фильтрации модулирующего сигнала фильтром низкой частоты с Гауссовской импульсной характеристикой.

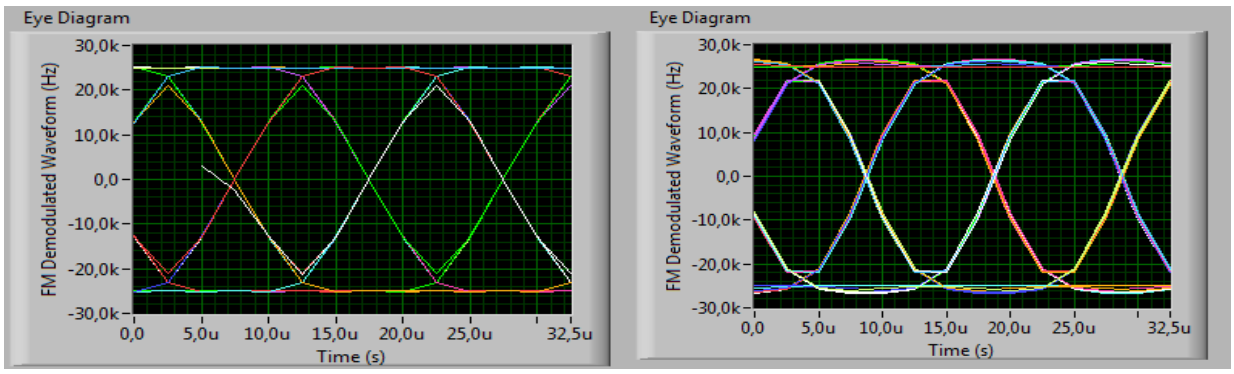


Рис. 1.30. Глазковая диаграмма без фильтра (слева) и с фильтром Гаусса (справа)

Сравнивая две глазковые диаграммы без использования фильтра и с фильтром Гаусса видно, что помехоустойчивость снизилась, появились помехи и искажения, а также увеличился джиттер.

Графики

Фильтр raisedcosine : M-fsk 4 (красный) и 2 (синий)

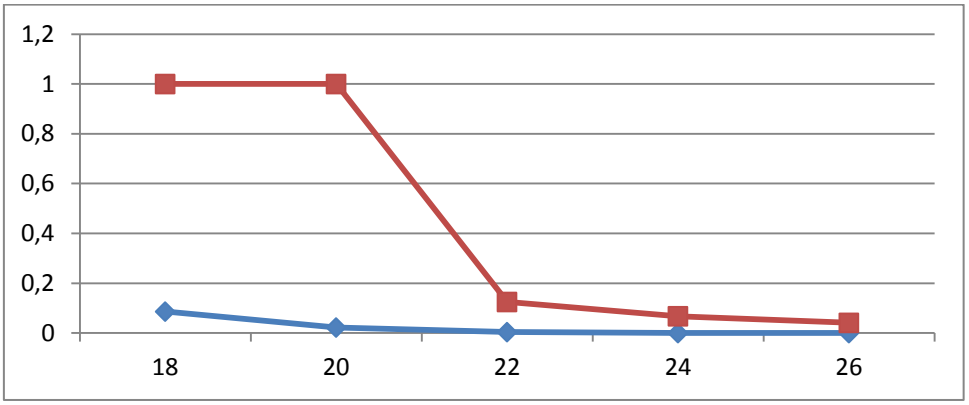


Рис. 1.31. Зависимость вероятности ошибки от отношения сигнал/шум

Фильтр rootraisedcos: M-fsk2 (красный) и 4 (синий)

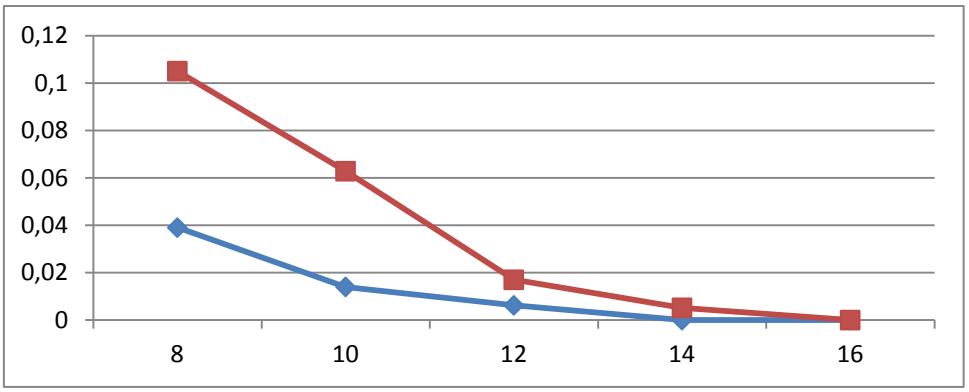


Рис. 1.32. Зависимость вероятности ошибки от отношения сигнал/шум

Фильтр Gaussian: M-fsk2 (красный) и 4 (синий)

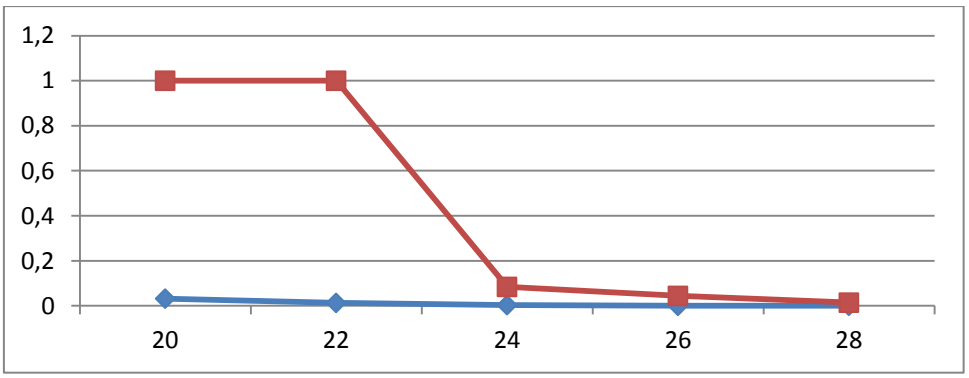


Рис. 1.33. Зависимость вероятности ошибки от отношения сигнал/шум

Частотная манипуляция с минимальным сдвигом (MSK)

MSK это частный случай сигналов с частотной манипуляцией с непрерывной фазой CPFSK с минимальным индексом частотной манипуляции m=0.5, поэтому ширина главного

лепестка спектра MSK сигналов - минимальная из всех возможных сигналов с частотной манипуляцией.

Представляет собой способ модуляции, при котором не происходит скачков фазы и изменение частоты происходит в моменты пересечения несущей нулевого уровня. MSK уникальна потому, что значение частот соответствующих логическим «0» и «1» отличаются на величину равную половине скорости передачи данных. Другими словами, индекс модуляции равен 0,5.

Частотная модуляция с минимальным разносом частот MSK позволяет уменьшить ширину полосы частот, занимаемых цифровым радиосигналом в эфире. Однако даже этот вид модуляции не удовлетворяет всем требованиям, предъявляемым к современным радиосистемам мобильной связи. Обычно сигнал MSK в радиопередатчике дофильтровывают обычным фильтром. Именно поэтому появился еще один вид модуляции с еще более узким спектром радиочастот в эфире.

Сужение полосы занимаемых частот удалось достигнуть за счет предварительной фильтрации модулирующего сигнала фильтром низкой частоты с Гауссовской импульсной характеристикой. Ширина спектра сигнала GMSK определяется произведением длительности передаваемого символа на полосу пропускания Гауссовского фильтра BT. Именно полосой пропускания B и отличаются различные виды GMSK друг от друга. GMSK используется в стандарте GSM сотовой связи.

Формирование GMSK радиосигнала осуществляется таким образом, что на интервале одного информационного бита фаза несущей изменяется на 90°. Это наименьшее возможное изменение фазы, распознаваемое при данном типе модуляции. Непрерывное изменение фазы синусоидального сигнала дает в результате частотную модуляцию с дискретным изменением частоты. Применение фильтра Гаусса позволяет при дискретном изменении частоты получить "гладкие переходы". В стандарте GSM применяется GMSK-модуляция с величиной нормированной полосы BT - 0,3, где B - ширина полосы фильтра по уровню минус 3 дБ, T - длительность одного бита цифрового сообщения. Основой формирователя GMSK-сигнала является квадратурный (1/Q) модулятор. Схема состоит из двух умножителей и одного сумматора. Задача этой схемы заключается в том, чтобы обеспечить непрерывную, очень точную фазовую модуляцию. Один умножитель изменяет амплитуду синусоидального, а второй косинусоидального колебания. Входной сигнал до умножителя разбивается на две квадратурные составляющие. Разложение происходит в двух обозначенных "sin" и "cos" блоках.

Модуляцию GMSK отличают следующие свойства, которые предпочтительны для

подвижной связи:

- постоянная по уровню огибающая, которая позволяет использовать эффективные
- передающие устройства с усилителями мощности в режиме класса С;
- компактный спектр на выходе усилителя мощности передающего устройства,
- обеспечивающий низкий уровень внеполосного излучения;
- хорошие характеристики помехоустойчивости канала связи.

Импульсная характеристика Гауссовского фильтра описывается следующей формулой:

$$h(t) = B\sqrt{\frac{2\pi}{\ln(2)}}e^{\frac{-(B_t\pi)^2}{\ln(2)}},$$

где B — полоса пропускания фильтра по уровню 3 дБ.

GMSK с фильтром Гаусса с BT = 1, вырождается в классическую MSK. Для формирования сигналов GSM стандартно используется фильтр Гаусса с BT = 0.3.

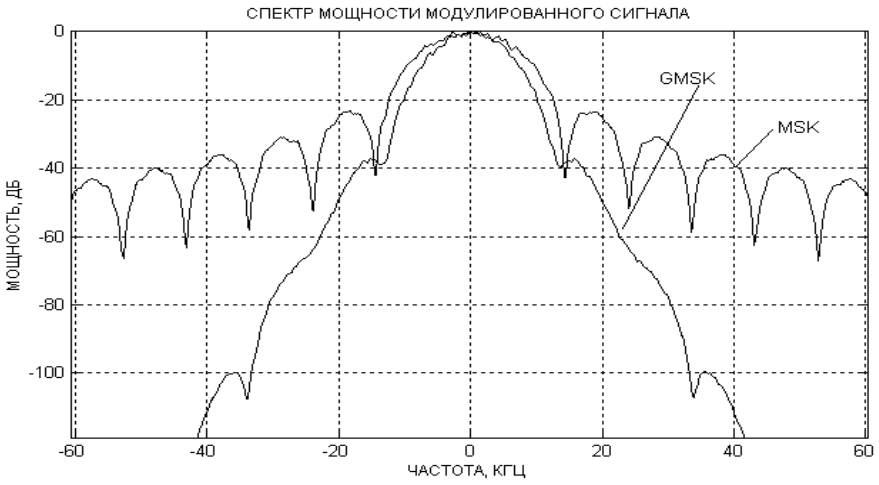


Рис. 1.34. Сравнение спектров двух видов модуляции с фильтром Гаусса(GMSK)

и без фильтра(MSK)

В точках взятия отсчетов сигнал GMSK зависит от предыдущих значений передаваемого сигнала. Это вызвано действием гауссовского фильтра, формирующего спектр сигнала GMSK. В результате помехоустойчивость сигнала GMSK ниже по сравнению даже с помехоустойчивостью сигнала MSK. Конкретное значение помехоустойчивости сигналов GMSK сильно зависит от произведения BT. Пример зависимости вероятности ошибки

приема сигнала GMSK в зависимости от отношения сигнал/шум на входе решающего устройства приведен на рисунке 1.35.

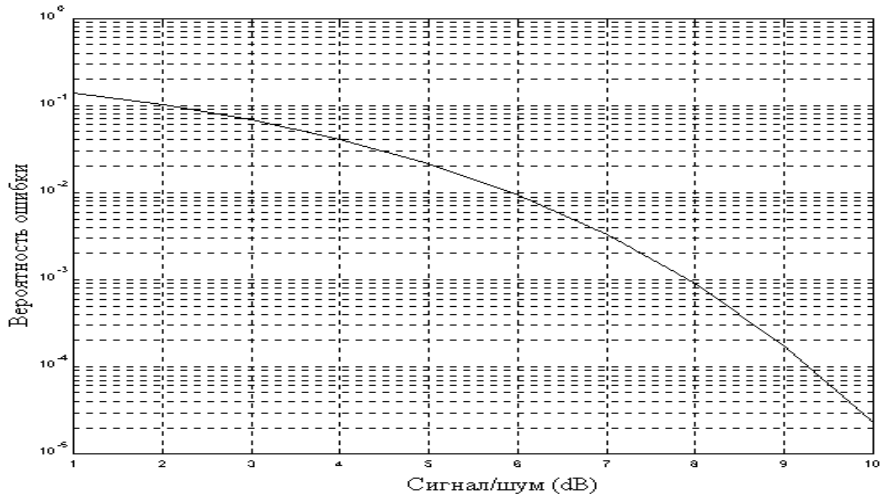


Рис. 1.35. График зависимости вероятности ошибки GMSK в зависимости от отношения сигнал/шум на входе решающего устройства

При индексе модуляции m = 0.5 за время передачи одного символа фаза несущего колебания успевает измениться на угол ±90°. Решетка переходов фазы в MSK сигнале на протяжении двух символьных периодов приведена на рисунке 1.36.

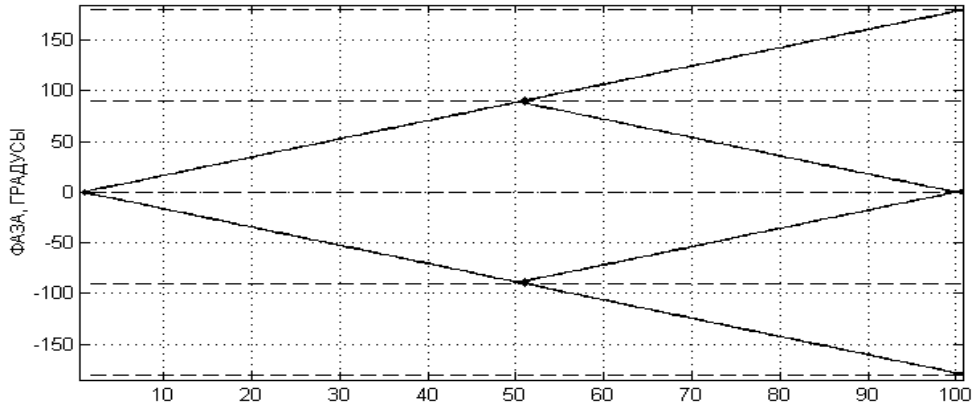


Рис. 1.36. Решетка переходов фазы в MSK сигнале

Два возможных значения фазы несущего колебания в отсчетной точке на одном временном интервале отличаются от двух возможных значений фазы несущего колебания отсчетной точке на соседнем интервале на 90°. Рассмотренная ситуация может быть проиллюстрирована на векторной диаграмме, приведенной на рисунке 1.37.

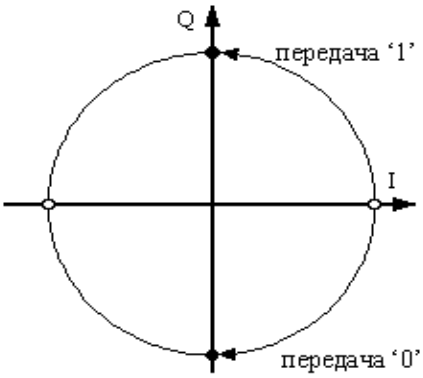


Рис. 1.37. Решетка переходов фазы в MSK сигнале.

Конец вектора, отражающий амплитуду несущего колебания на рисунке 4, в сигнале частотной модуляции может двигаться строго по окружности. На этом рисунке пара состояний сигнала показана пустыми кружочками, а пара состояний сигнала на соседнем символе — заполненными.

Рассмотренные диаграммы показывают, что при разработке радиоприемного устройства можно применить схему фазового детектора. Принимаемые двоичные символы в отсчетных точках будут отличаться друг от друга по фазе на 180°.

Описание прибора с MSK модуляцией (манипуляцией)

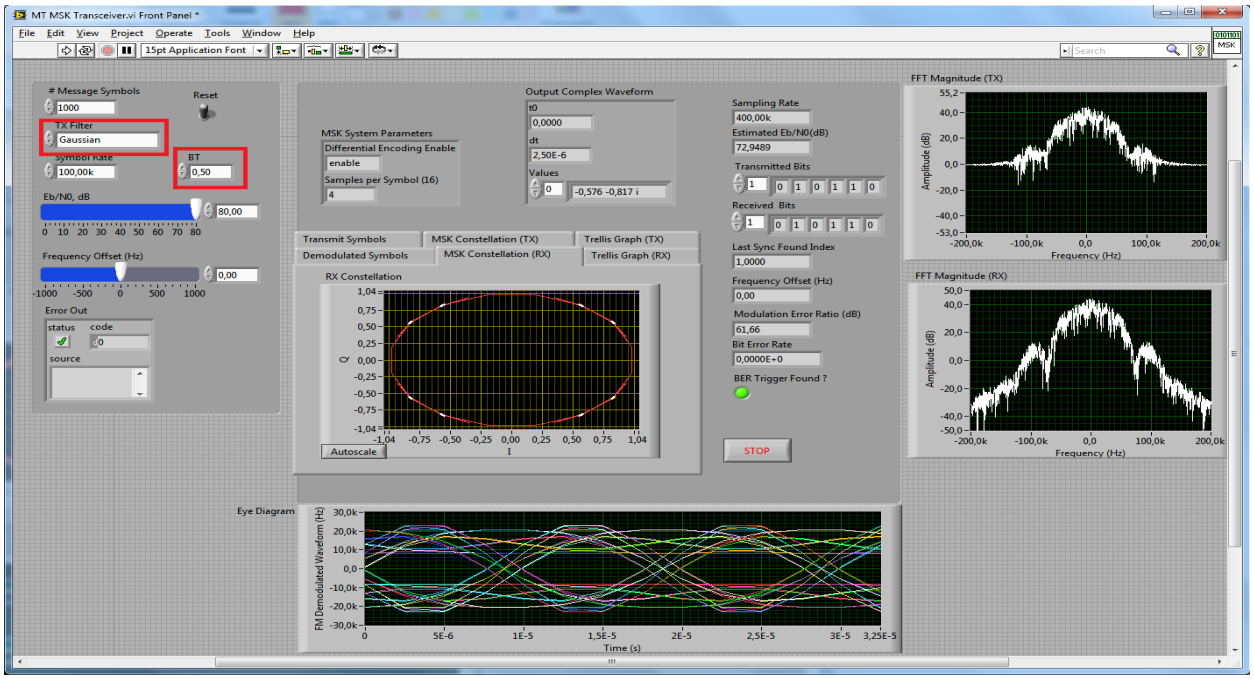


Рис. 1.38. Интерфейс программы с GMSK модуляцией

- Message Symbols – показывает количество передаваемых символов.
- Eb/N0- отношение сигнал/шум
- Frequency offset –задержка по частоте

- TX Filter – выбор фильтра для передатчика. Доступны следующие: Gaussian, Raised Cosine, Root Raised Cosine
- Symbol Rate – порядок символов.
- Transmitted Bits – переданные биты.
- Received Bits – принятые биты.
- BER Trigger Found – Вероятность шибки
- BT - безразмерная величина равная $BT = 0{,}5$
- Сокращения: TX – Передатчик, RX – Приемник

Исследование линии передачи с MSK модуляцией

В качестве характеристического параметра *GMSK* используют произведение *BT,* где *B* - ширина спектра импульса *h(t)* по уровню 3 дБ, а *T*- длительность одного бита. На рис. 1.39 изображена спектральная плотность мощности на выходе идеального

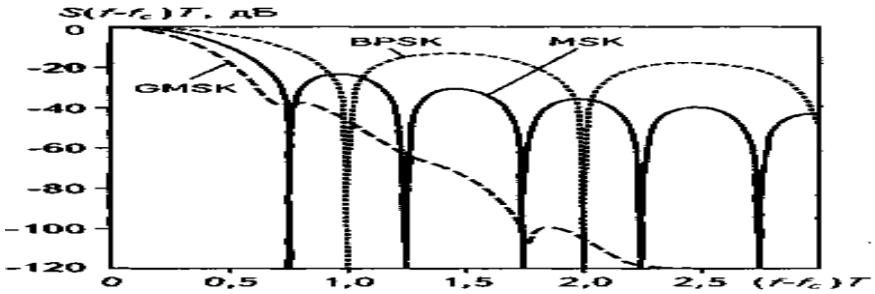


Рис. 1.39. Спектр сигнала с GMSK модуляцией

GMSK-модулятора *(BT* = 0,3), нормализованного по отношению к периоду *T*. Сравнение этого графика со спектральной плотностью мощности *MSK* и *BPSK*, указывает на серьезное преимущество этой модуляции, прежде всего - в части скорости спадания внеполосного излучения, т.е. скорости снижения уровня мощности боковых спектральных лепестков.

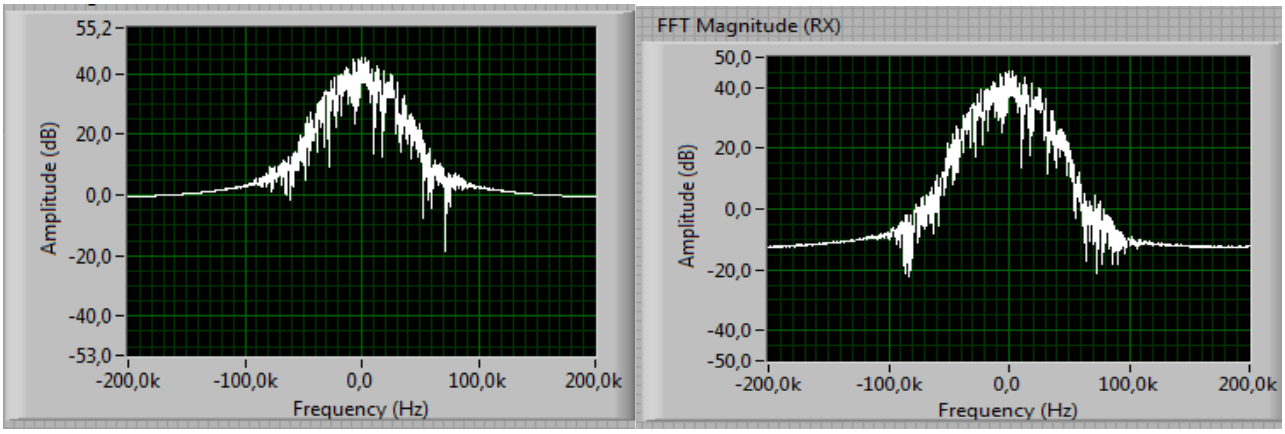


Рис. 1.40. Спектр передаваемого (слева) и принимаемого (справа) сигнала с GMSK модуляцией при BT = 0.2, сигнал/шум = 80

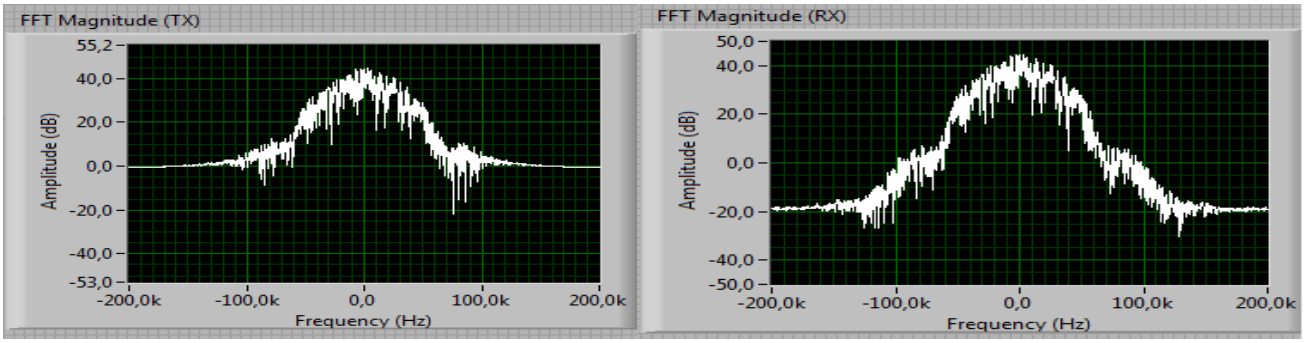


Рис. 1.41. Спектр передаваемого (слева) и принимаемого (справа) сигнала с GMSK модуляцией при BT = 0.3, сигнал/шум = 80

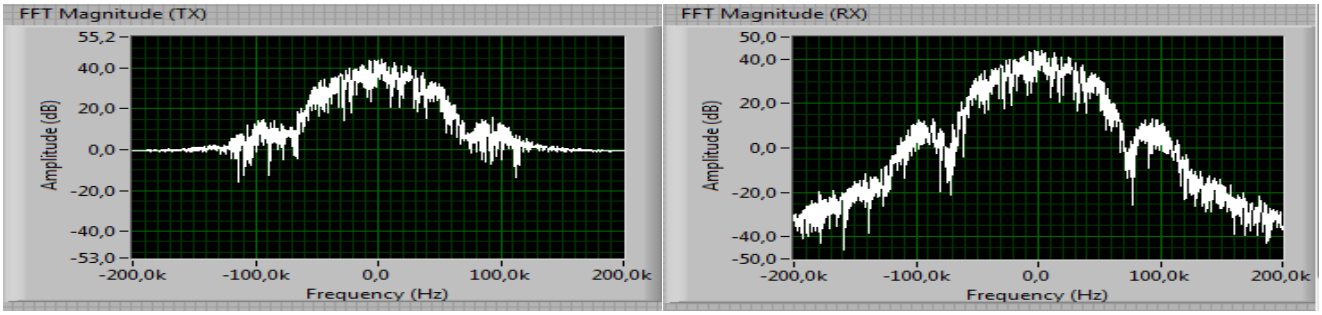


Рис. 1.42. Спектр передаваемого (слева) и принимаемого (справа) сигнала с GMSK модуляцией при BT = 0.5, сигнал/шум = 80

Можно видеть, что в спектре GMSK при уменьшении BT уменьшаются уровни боковых лепестков, кроме того значительно возрастает скорость убывания спектра. Так максимальный уровень бокового лепестка GMSK при BT = 0.3 меньше, чем при MSK модуляции, а скорость убывания линейно зависит от частоты, что обусловлено применением

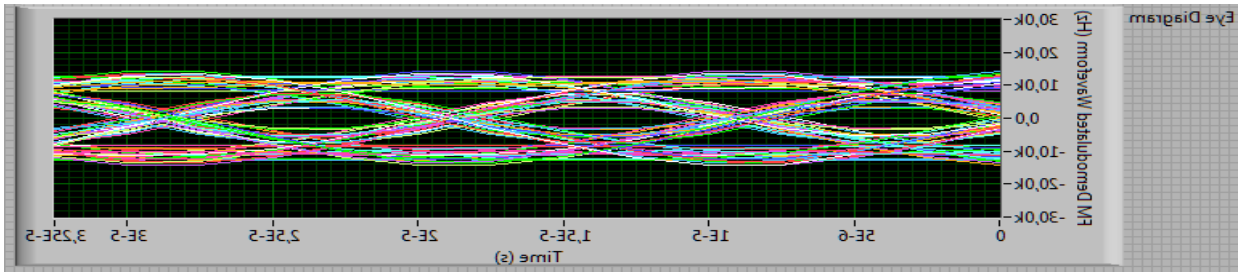


Рис. 1.43. Глазковая диаграмма сигнала с GMSK модуляцией при BT = 0.2, сигнал/шум = 80

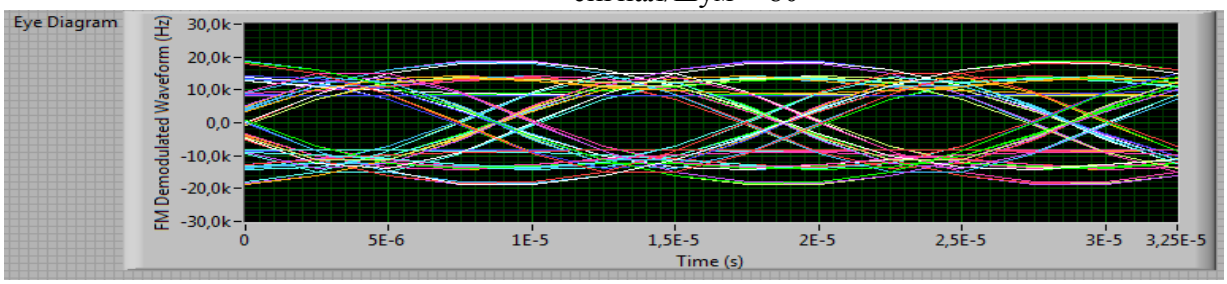


Рис. 1.44. Глазковая диаграмма сигнала с GMSK модуляцией при BT = 0.3, сигнал/шум = 80

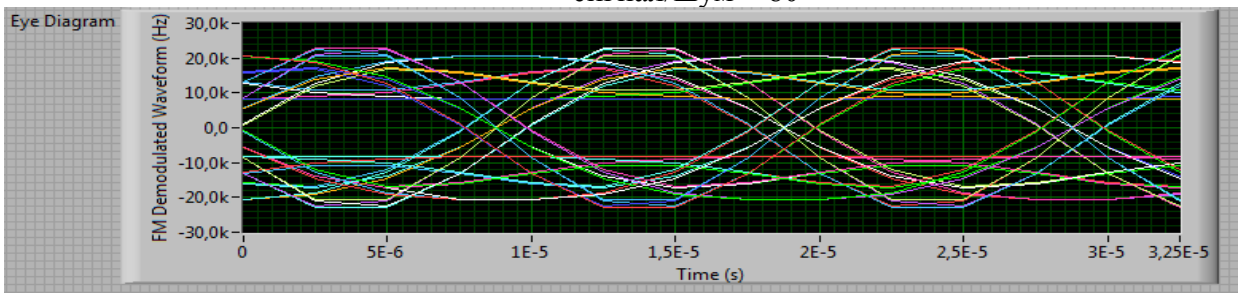


Рис. 1.45. Глазковая диаграмма сигнала с GMSK модуляцией при BT = 0.5, сигнал/шум = 80

Из рисунков отчетливо видно, что с уменьшением межсимвольная интерференция усиливается ввиду расширения фильтра Гаусса. Фильтр Гаусса вносит межсимвольную интерференцию и позволяет снизить уровень боковых лепестков спектра, а также значительно увеличивает скорость убывания спектра GMSK сигнала по сравнению с MSK сигналами. На практике GMSK модуляция нашла применение в сотовой связи формата GSM при , ввиду наиболее эффективного использования радиочастотного ресурса. Дальнейшее уменьшение параметра приводит к существенному усложнению аппаратуры и сильно увеличивает вероятность ошибочного приема символа из-за недопустимо высокой межсимвольной интерференции.

Из полученных графиков можно сделать вывод что для обеспечения той же самой вероятности ошибки требуется различное отношение сигнал\шум, т.е. при использовании фильтра Гаусса требования к качеству принимаемого сигнала возрастают.

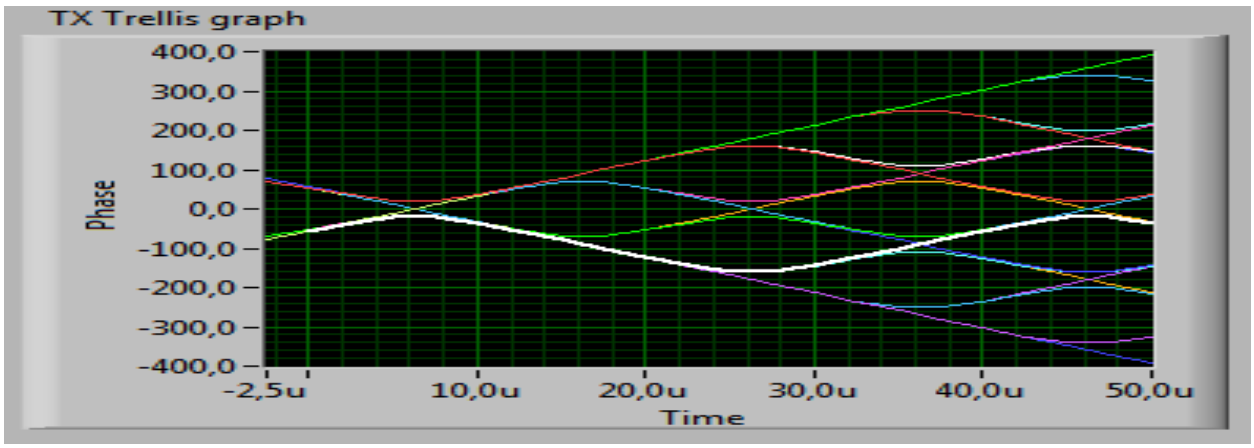


Рис. 1.46. Фазовая диаграмма передаваемого сигнала

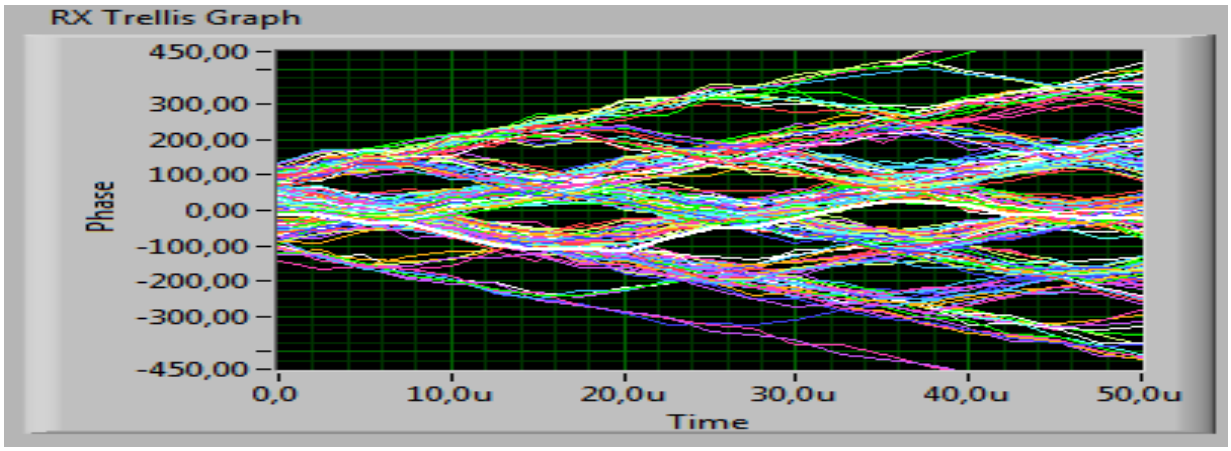


Рис. 1.47. Фазовая диаграмма полученного сигнала при BER=2.04*10^{-3},

сигнал/шум = 14

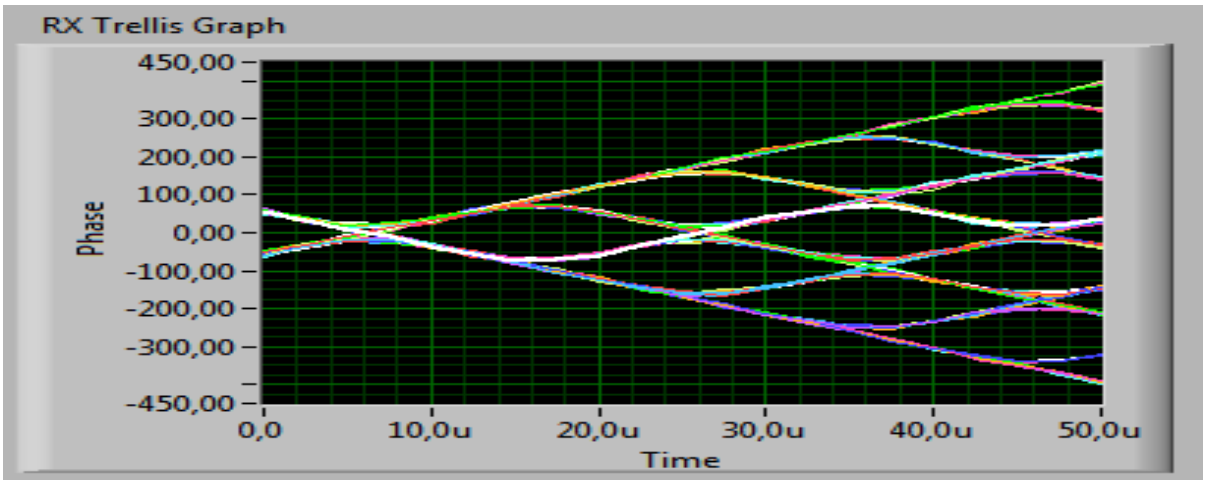


Рис. 1.48. Фазовая диаграмма полученного сигнала при BER= 10^{-4}, сигнал/шум = 28

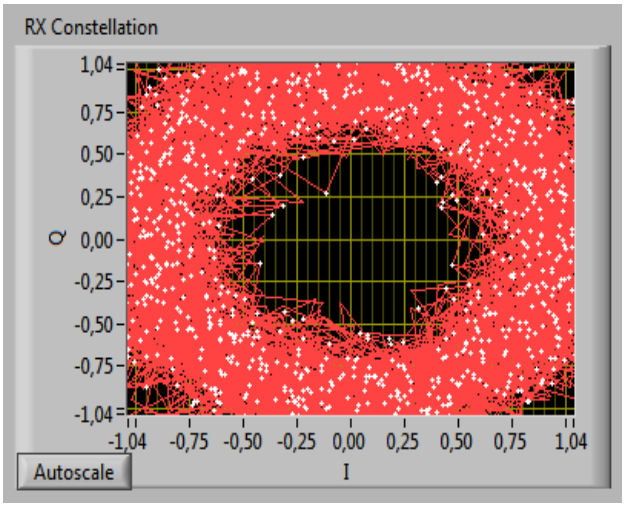


Рис. 1.49. Сигнальное созвездие сигнала при BER=2.04*10^{-3}, сигнал/шум = 14

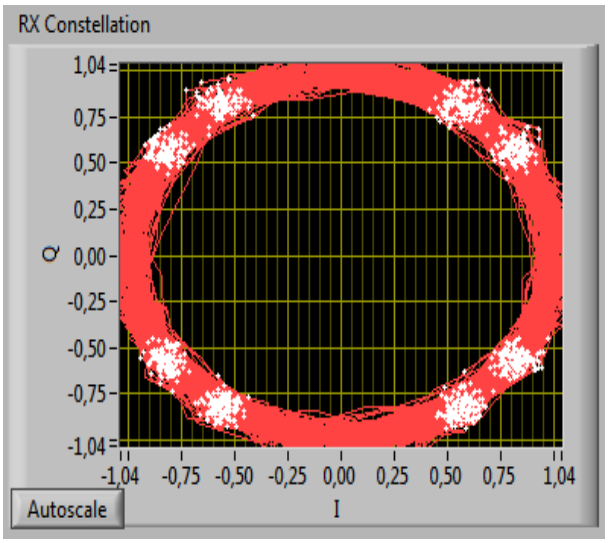


Рис. 1.50. Сигнальное созвездие сигнала при BER= 10^{-4}, сигнал/шум = 28

TX Filter Raised Cosine (синий), Root Raised Cos (красный), Gaussian (зеленый)

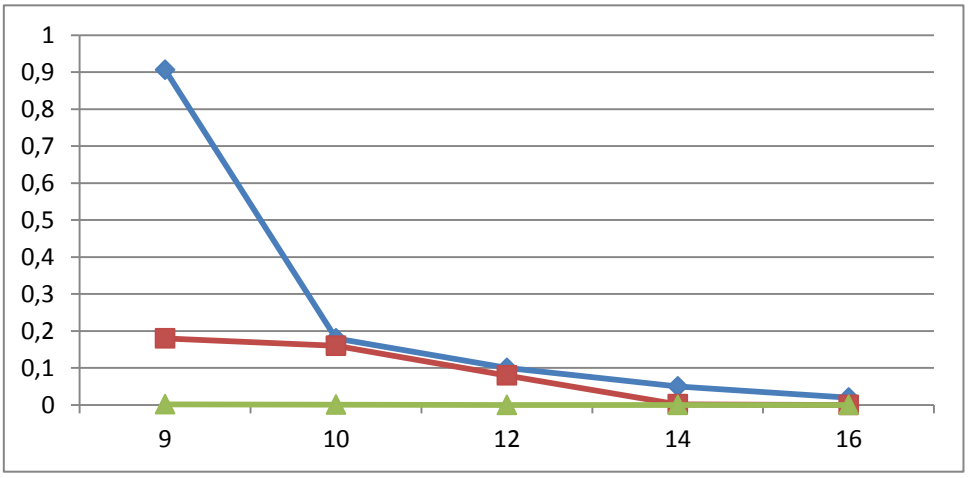


Рис. 1.51. Зависимость вероятности ошибки от отношения сигнал/шум

Таким образом, получаются следующие характеристики этих модуляций. FSK - простота реализации и относительно широкий спектр. GFSK - простота реализации меньший спектр, но и несколько худшая помехоустойчивость по отношению к FSK. MSK - очень высокая помехоустойчивость при относительно небольшом спектре, но сложная схема модулятора-демодулятора. GMSK - ширина спектра практически близка к теоретическому пределу Br (скорость передачи цифровой информации), несколько худшая помехоустойчивость по сравнению с MSK, сложность модулятора-демодулятора такая же как у MSK.

При использовании программного обеспечения LabVIEW, были созданы две виртуальные измерительные лаборатории с различными видами модуляции сигнала FSK(Frequency Shift Keying) и MSK(Minimal Shift Keying) в каждой из которых можно

применить предварительные фильтрации и получить другие подвиды GMSK или GFSK, если использовать фильтр Гаусса.

1.3. Модемы спутниковых систем связи M-QAM, M-PSK и численный анализ вероятности символьной ошибки с использованием ПО LabVIEW

В данном разделе был разработан программный комплекс для изучения многоуровневых методов модуляции применяемых в спутниковых системах разработанная в ПО LabVIEW 2011 SP 1. В разработанном ПО можно исследовать такие виды манипуляции как BPSK, QPSK, 8-PSK, 16-PSK, 32-PSK, 64-PSK и M-QAM на то, как влияет отношение сигнал/шум на вид диаграммы, посмотреть глазковые диаграммы, спектр, оценить BER, посмотреть влияние фильтров на диаграмму.

Наиболее часто используют QPSK манипуляцию. Она имеет преимущество, поскольку может работать при мощности транспондера, близкой к насыщению, то есть эффективной отдачей энергии. В таблице 1 приведены теоретические значения отношения сигнал/шум, требуемые для достижения величины BER, равной 10^{-10}, без кодирования канала связи для различных методов манипуляции.

Таблица 1.2. Сравнение методов цифровой манипуляции для BER, равной 10^{-10}

Метод манипуляции.	Сигнал/шум без применения кодирования, дБ.	Эффективность использования полосы пропускания, бит/с.
BPSK	13.06	1.0
QPSK	13.06	2.0
8-PSK	16.55	3.0
16-PSK	21.09	4.0
4-QAM	13.06	2.0
16-QAM	16.98	4.0
64-QAM	21.40	6.0

Квадратурная фазовая манипуляция (QPSK)

QPSK манипуляция строится на основе кодирования двух бит передаваемой информации одним символом. При этом символьная скорость в два раза ниже скорости передачи информации. На рис. 1.43 представлена векторная диаграмма QPSK.

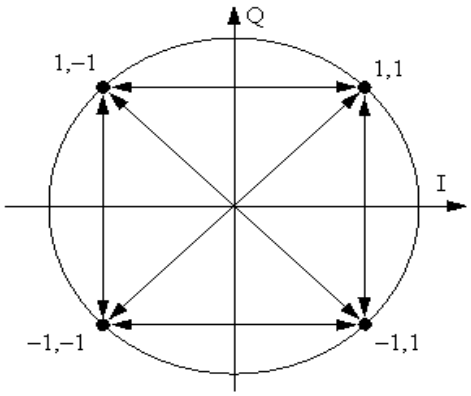


Рис. 1.52. Векторная диаграмма QPSK.

Кодирование осуществляется следующим образом: весь битовый поток разбивается на четные и нечетные биты, тогда I(t) будет кодировать четные биты, а Q(t) – нечетные. Два последовательно идущих друг за другом бита информации кодируются одновременно синфазным I(t) и квадратурным Q(t) сигналами. Это наглядно показано на осциллограммах, приведенных на рисунке 1.53 для информационного потока «1100101101100001».

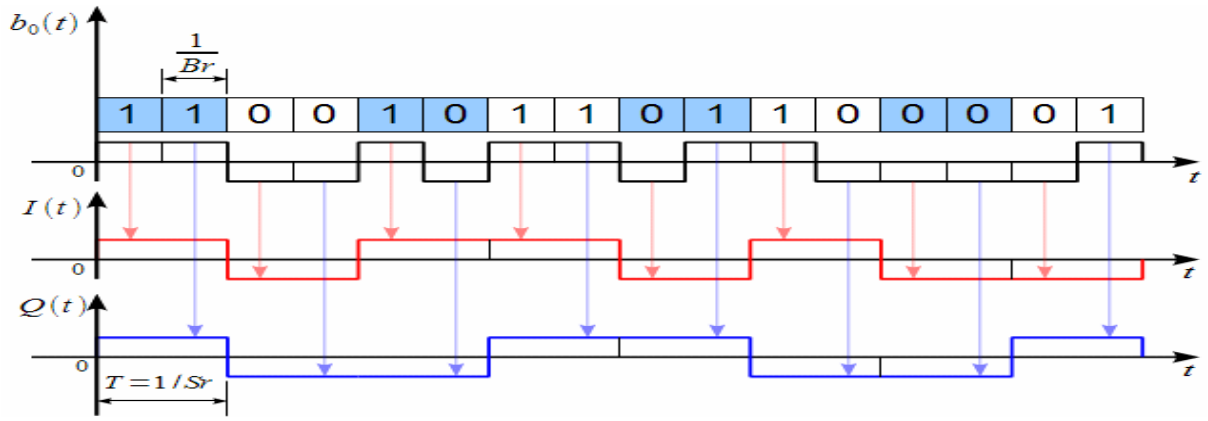


Рис. 1.53. Синфазная и квадратурная составляющие QPSK сигнала

На верхнем графике входной поток разделен на пары бит, соответствующих одной точке созвездия QPSK, показанного на рисунке 1.53. На втором графике показана осциллограмма I(t), соответствующая передаваемой информации I(t)>0. Если четный бит равен 1 (биты нумеруются с нуля, а не с единицы, поэтому первый в очереди бит имеет номер 0, а значит он четный по порядку), и I(t)<0 если четный бит 0 (т.е. b(t)<0). Аналогично строится квадратурный канал Q(t), но только по нечетным битам. Длительность одного символа T=1/Sr в два раза больше длительности одного бита исходной информации. Устройство выполняющее такое кодирование I(t) и Q(t) согласно созвездию, QPSK условно показано на рисунке 1.54.

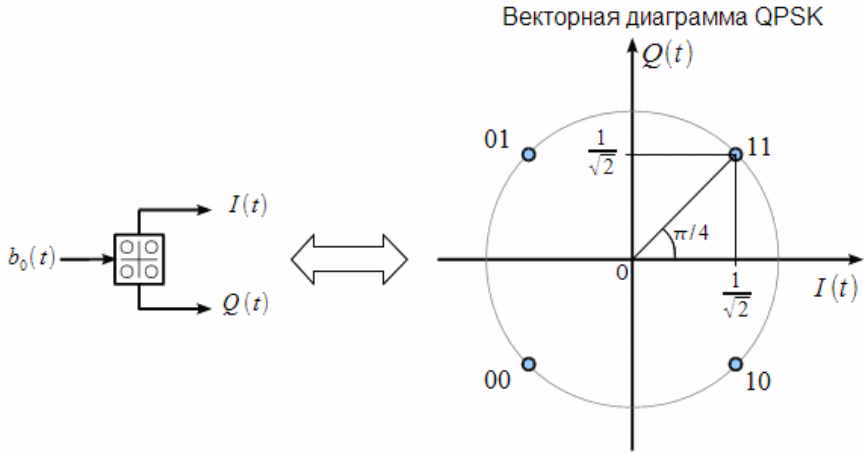


Рис. 1.54. Устройство кодирования синфазной и квадратурной составляющих на основе созвездия QPSK.

В зависимость от пары бит b(t) на входе на выходе получаем постоянные в пределах длительности этой пары бит сигналы I(t) и Q(t), значение которых зависит от передаваемой информации.

Структурная схема QPSK модулятора.

Структурная схема QPSK модулятора на основе универсального квадратурного модулятора показана на рисунке 1.55.

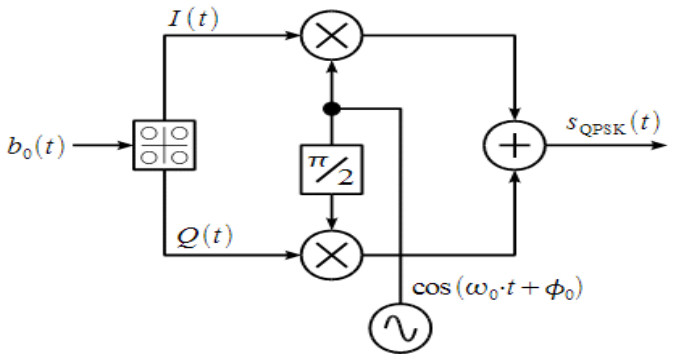


Рис. 1.55. Структурная схема QPSK модулятора.

Сигнал $S_{QPSK}(t)$ имеет вид:

$$S_{QPSK}(t) = I(t) \cdot \cos(\omega_0 t + \varphi_0) - Q(t) \cdot \sin(\omega_0 t + \varphi_0)$$

Синфазная *I(t)* и квадратурная *Q(t)* составляющие это ничто иное, как реальная и мнимая части комплексной огибающей QPSK сигнала $Z(t) = I(t) + j \cdot Q(t)$, которые являются входными сигналами квадратурного модулятора. Тогда можно представить $S_{QPSK}(t)$ через его комплексную огибающую *Z(t)*: $S_{QPSK}(t) = R\left|Z(t) \cdot \exp(j\omega_{0t}\right|$. Из комплексной огибающей можно выделить фазовую огибающую как:

$$\varphi(t) = \arctan\left(\frac{j[Z(t)]}{R[Z(t)]}\right) = \arctan(\frac{Q(t)}{I(t)})$$

Важно отметить, что арктангенс должен вычисляться с учетом четверти комплексной плоскости (функции арктангенс 2). Вид фазовой огибающей $\varphi(t)$ для информационного потока «1100101101100001» показан на рисунке 1.56.

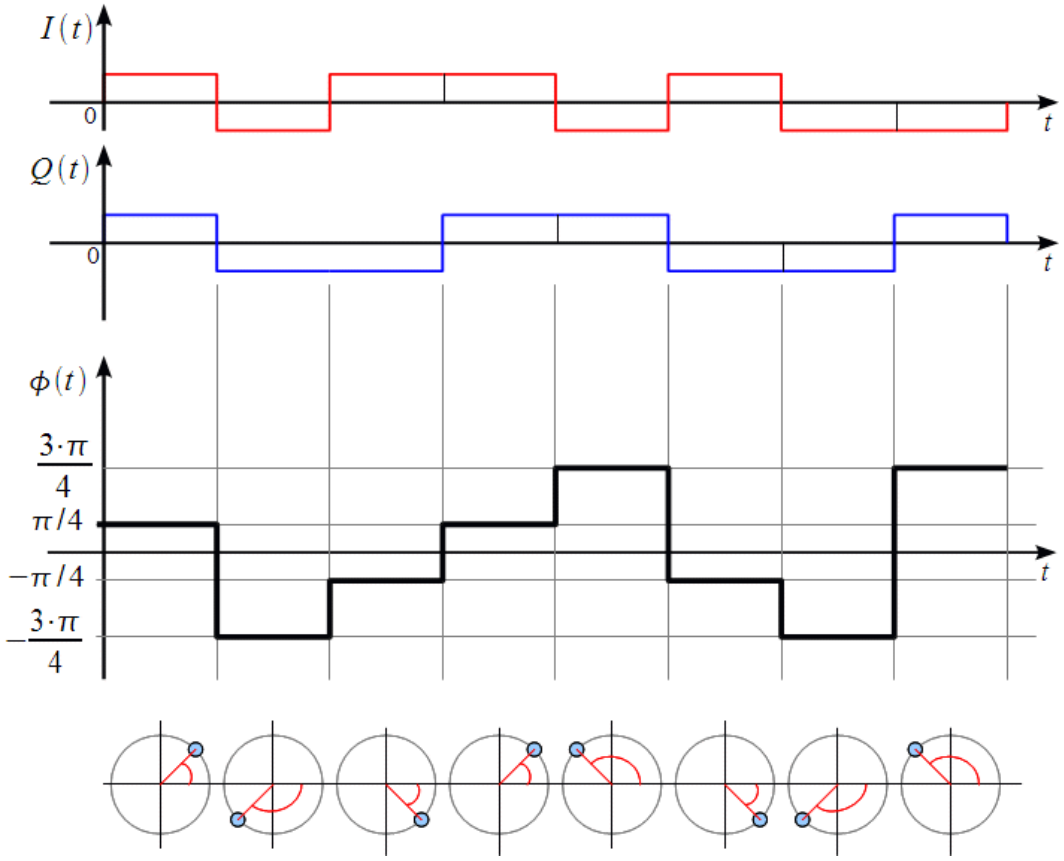


Рис. 1.56. Фазовая огибающая QPSK сигнала.

Фазовая огибающая представляет собой ступенчатую функцию времени, претерпевающую разрывы в моменты смены символа QPSK (один символ QPSK несет два бита информации). При этом в пределах одного символа векторная диаграмма QPSK находится всегда в одной точке созвездия, как это показано внизу, а при смене символа – скачкообразно переходит в точку соответствующую следующему символу. Поскольку у QPSK всего четыре точки в созвездии, то фазовая огибающая может принимать всего четыре значения: $\mp\frac{\pi}{4}$ и $\mp\frac{3\cdot\pi}{4}$.

Амплитудная огибающая QPSK сигнала *a(t)* также может быть получена из комплексной огибающей *Z(t)*:

$$a(t) = \sqrt{I^2(t) + Q^2(t)}$$

Амплитудная огибающая QPSK сигнала равна единице всюду, за исключением моментов смены передаваемых символов, т. е. в моменты перескока фазы и перехода очередной точке созвездия.

Пример осциллограммы QPSK сигнала при входном битовом потоке «1100101101100001» при скорости передачи информации Br=10 кбит/с и несущей частоте 20 кГц показан на рисунке 1.57.

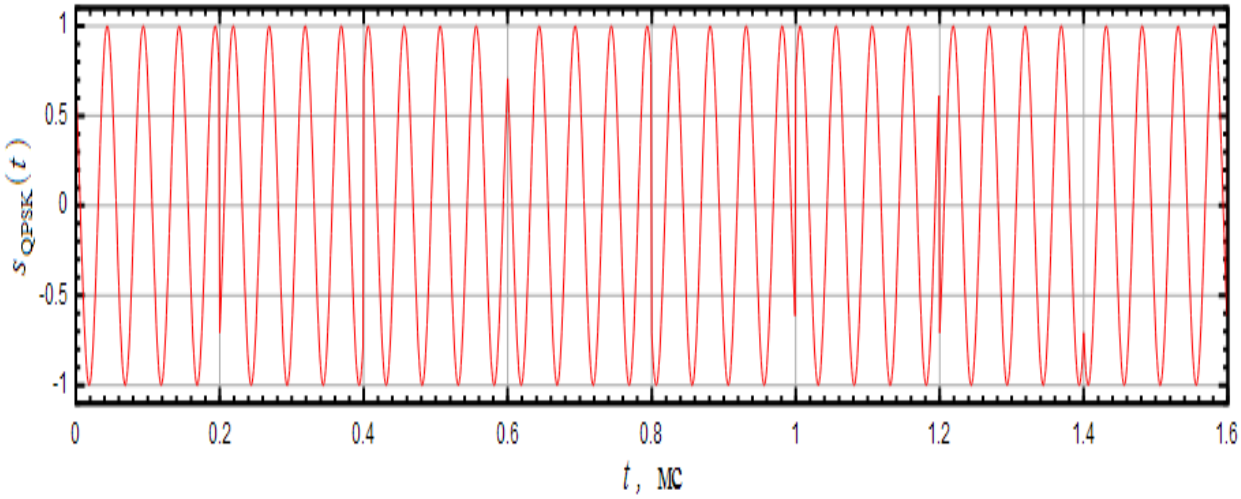


Рис. 1.57. Осциллограмма QPSK сигнала.

Фаза несущего колебания может принимать четыре значения: $\mp\frac{\pi}{4}$ и $\mp\frac{3\cdot\pi}{4}$ радиан. При этом фаза следующего символа относительно предыдущего может не изменится, или измениться на $\mp\frac{\pi}{2}$ или на $\mp\pi$ радиан. Также отметим, что при скорости передачи информации Br=10 кбит/с мы имеем символьную скорост Sr=Br/2=5 кбит/с, и длительность одного символа T=1/Sr=0.2 мс, что отчетливо видно на осциллограмме (скачок фазы происходит через 0.2 мс).

Формирование спектра QPSK сигнала с помощью фильтров Найквиста.

На рисунке 1.58 показан спектр QPSK $\left|S_{QPSK}(f)\right|^2$ сигналов при Br=10 кбит/с и несущей частоте 100 кГц. Можно заметить, что ширина главного лепестка, а также боковых лепестков QPSK сигнала вдвое меньше чем у BPSK сигнала при одной скорости передачи информации. Это обусловлено тем, что символьная скорость Sr QPSK сигнала вдвое меньше скорости передачи информации Br.

Использование формирующих фильтров дает возможность передавать 0.5 символа в секунду на 1 Гц полосы, или 1 бит/с цифровой информации на 1 Гц полосы при использовании фильтра с АЧХ вида приподнятого косинуса.

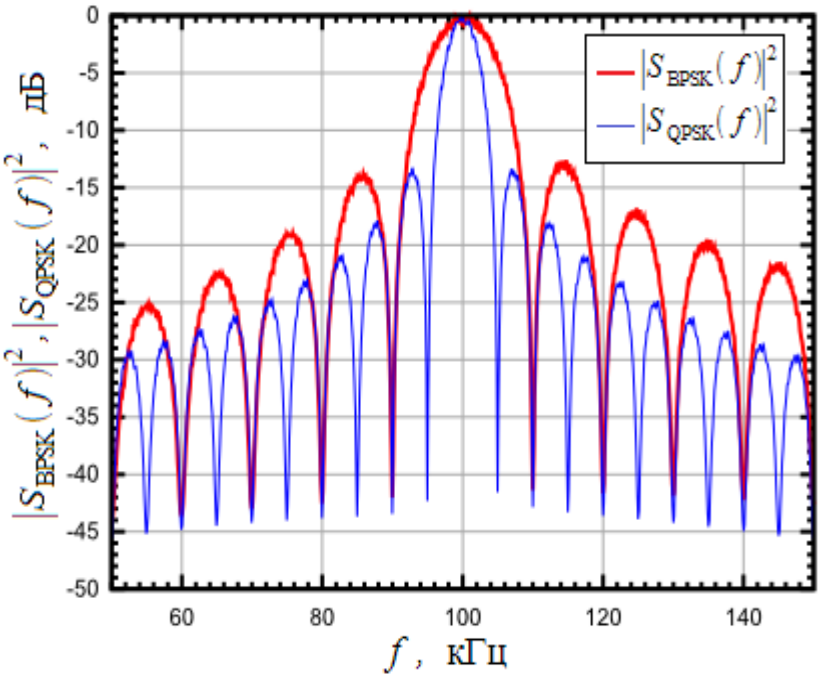


Рис. 1.58. Спектр QPSK сигнала.

На рисунке 1.58 черным показан спектр QPSK сигнала без использования формирующего фильтра. Видно, что применение фильтра Найквиста позволяет полностью подавить боковые лепестки в спектре QPSK сигналов. Структурная схема QPSK модулятора при использовании формирующего фильтра показана на рисунке 1.59.

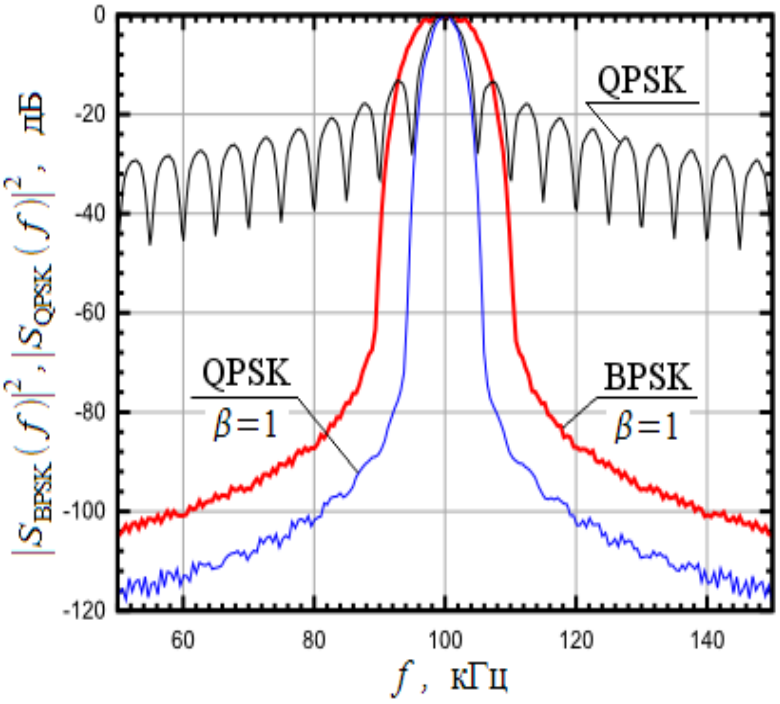


Рис. 1. 59. Спектр QPSK сигнала с формирующим фильтром Найквиста

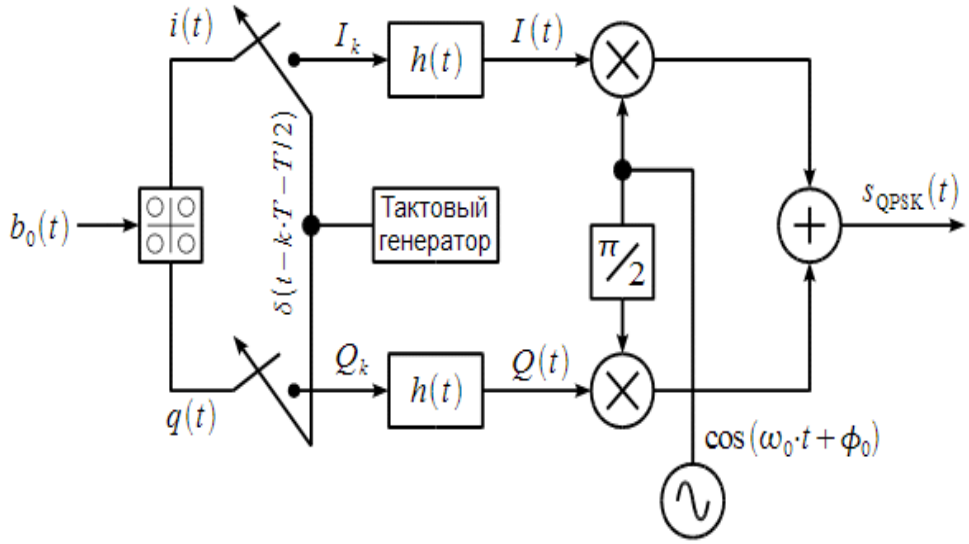


Рис. 1. 60. Структурная схема QPSK модулятора с использованием формирующего фильтра

Графики, поясняющие работу QPSK модулятора показаны на рисунке 1.61.

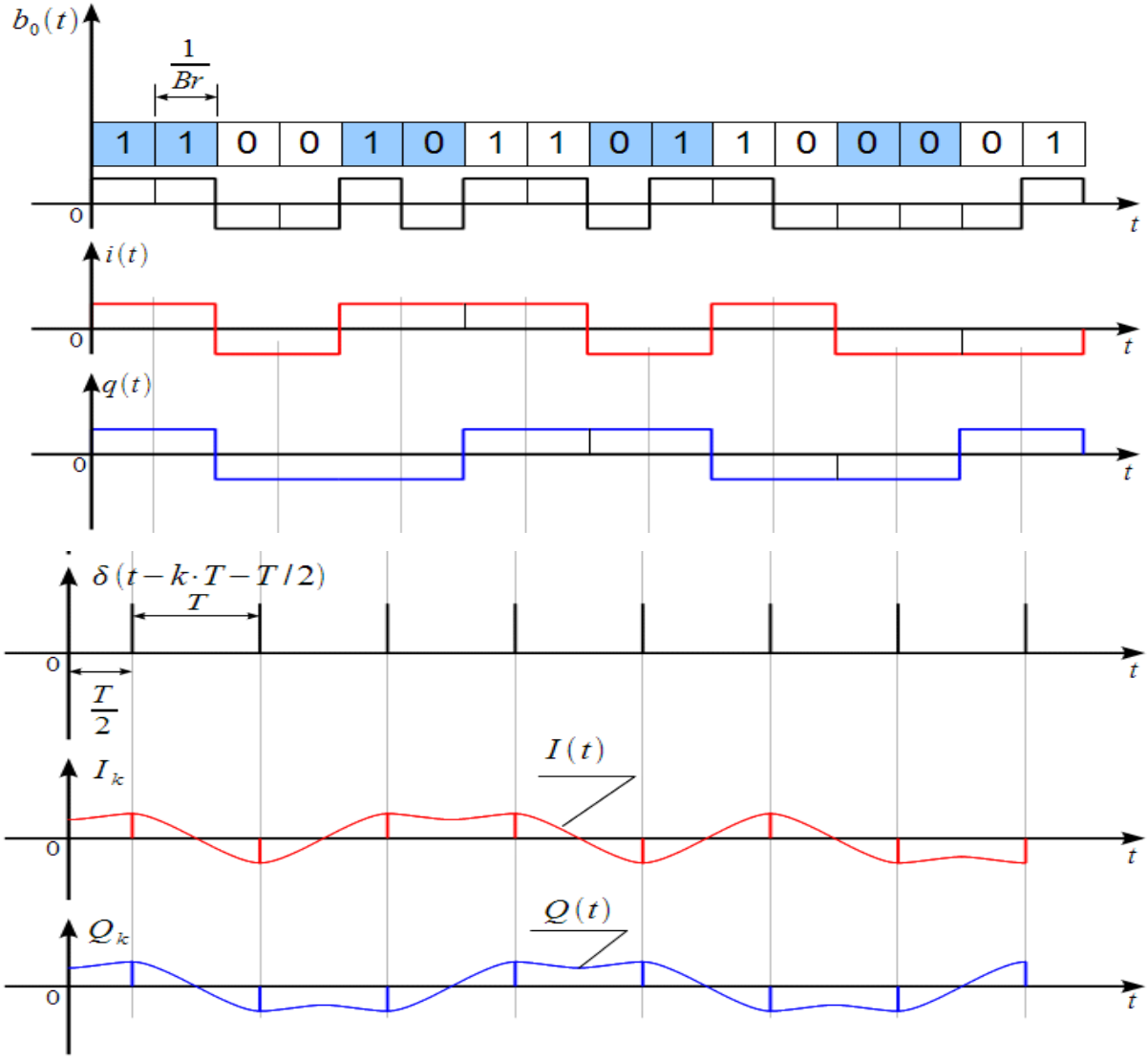


Рис. 1.61. Поясняющие графики.

Цифровая информация поступает со скоростью Br и преобразуется в символы i(t) и q(t) в соответствии с созвездием QPSK, длительность одного передаваемого символа равна T=1/Sr=2/Br. Тактовый генератор выдает последовательность дельта-импульсов с периодом T, но отнесенных к центру импульса i(t) и q(t), как это показано на четвертом графике. Импульсы тактового генератора стробируют i(t) и q(t) при помощи ключей и получаем отсчеты I_k и Q_k, показанные на двух нижних графиках, которые возбуждают формирующий фильтр интерполятор с импульсной характеристикой h(t) и на выходе имеем синфазную I(t) и квадратурную Q(t) составляющие комплексной огибающей, которые подаются на универсальный квадратурный модулятор. На выходе модулятора получаем QPSK сигнал с подавлением боковых лепестков спектра.

Обратим внимание, что синфазная I(t) и квадратурная Q(t) составляющие становятся непрерывными функциями времени, в результате вектор комплексной огибающей QPSK уже не находится в точках созвездия, перескакивая во время смены символа, а непрерывно движется комплексной плоскости как это показано на рисунке 1.62 при использовании фильтра приподнятого косинуса с различными параметрами β.

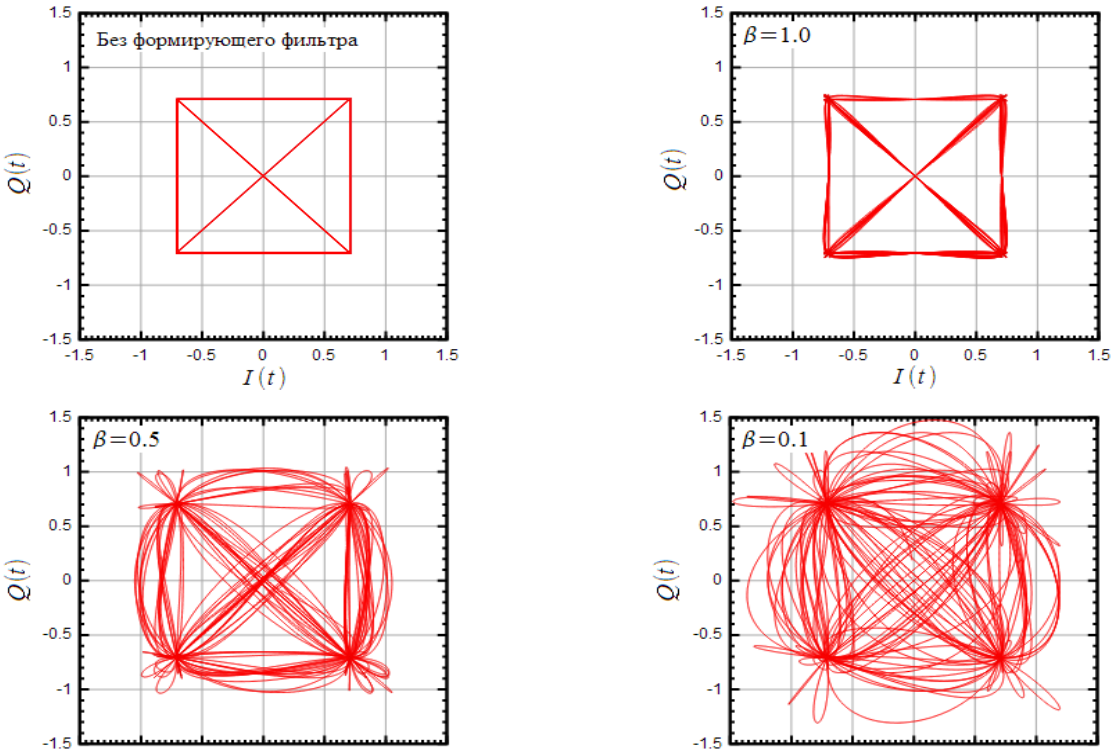


Рис. 1.62. Траектории движения вектора комплексной огибающей QPSK сигнала при различных параметрах формирующего фильтра Найквиста.

На верхнем левом графике показана векторная диаграмма при отсутствии формирующего фильтра при скачкообразном перемещении вектора комплексной огибающей. Если включить фильтр Найквиста, то при уменьшении β векторная диаграмма

превращается в «клубок». При β=1 получаем наилучшее приближение к идеальной диаграмме.

При непрерывном движении вектора комплексной огибающей его амплитуда начинает меняться во времени, значит у QPSK сигнала при использовании формирующего фильтра начинает меняться амплитудная огибающая

$$a(t) = \sqrt{I^2(t) + Q^2(t)},$$

что наглядно демонстрируется осциллограммой QPSK сигнала, показанной на рисунке 1.63.

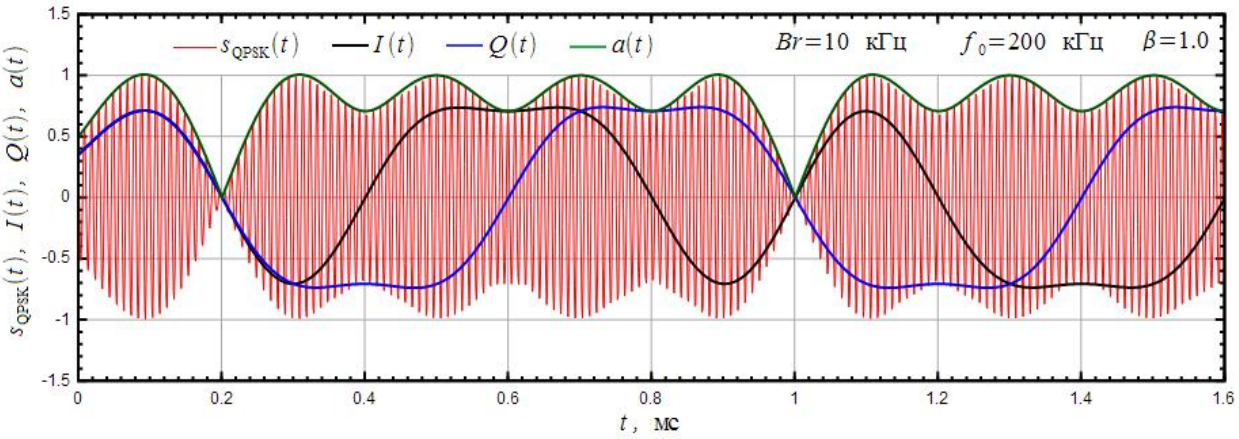


Рис. 1.63. Осциллограмма QPSK сигнала при использовании формирующего фильтра Найквиста

Видно, что фильтр Найквиста приводит к появлению паразитной амплитудной модуляции. При этом в точках когда и синфазная I(t) и квадратурная Q(t) составляющие комплексной огибающей равны нулю, амплитуда a(t) QPSK сигнала также падает до нуля, и фаза поворачивается на π радиан. Глубокая амплитудная модуляция — это негативный эффект, который устраняется офсетной QPSK (OQPSK) модуляции. Важно отметить, что при непрерывных I(t) и Q(t) фазовая огибающая также становится непрерывной функцией времени и перестает меняться скачкообразно, а плавно перетекает от символа к символу, что и приводит к сужению спектра QPSK сигнала при использовании формирующего фильтра.

Практическая часть.

Внешний вид разработанного программного обеспечения представлен на рисунке 1.64.

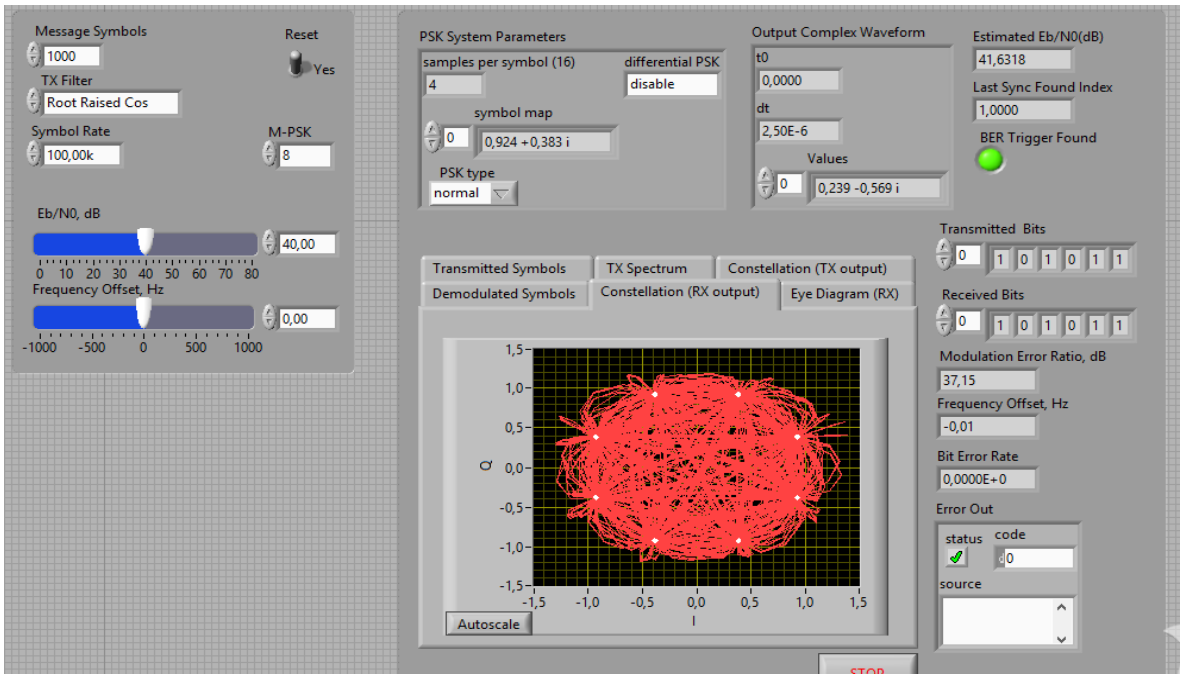


Рис. 1.64. Внешний вид разработанного ПО для исследования QPSK.

Программа имеет два блока:

1. Блок настройки (слева);
2. Блок отображения результата (справа).

В блоке настройки можно задавать количество символов в передаваемом сообщении (поле Message Symbols), чем больше символов в сообщении, тем дольше оно будет передаваться. На рисунке 1.65 показано как меняется диаграмма QPSK снятая с приемника в зависимости от количества символов сообщении.

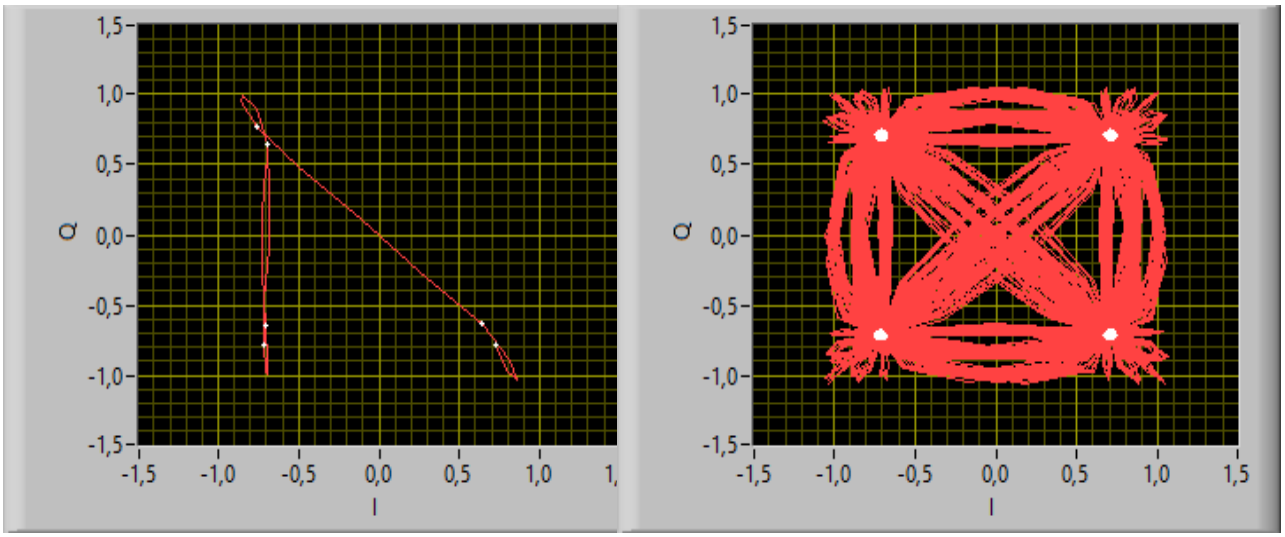


Рис. 1.65. Диаграмма QPSK. 10 символов в сообщении (слева), 1000 символов в сообщении (справа).

Можно задать значение отношения сигнал/шум (поле Eb/N0, dB). На рисунке 16 представлена диаграмма QPSK для разных значений сигнал/шум.

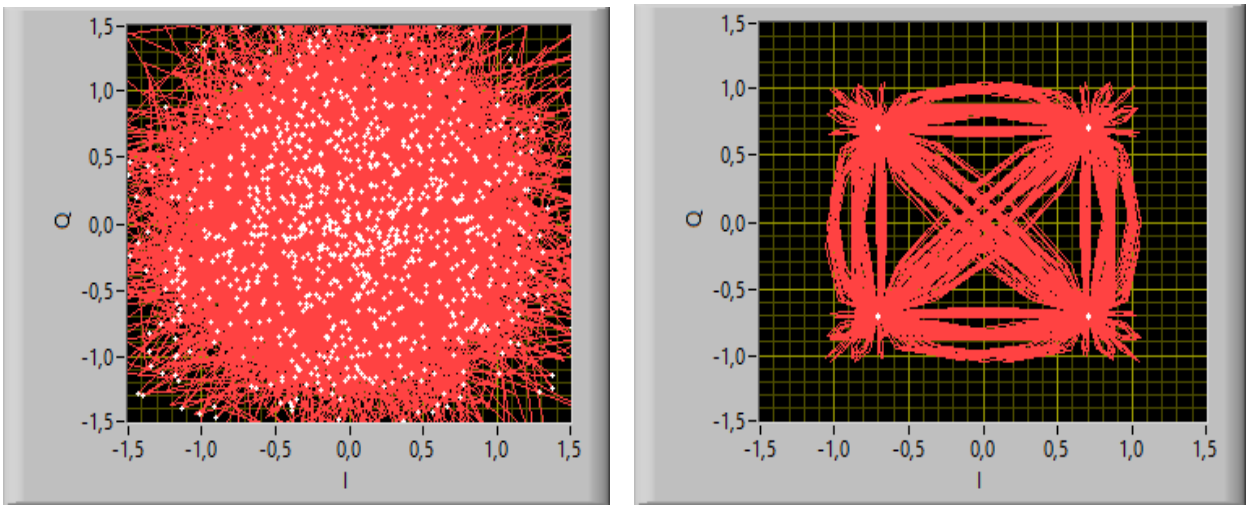


Рис. 1.66. Диаграмма QPSK. Отношение сигнал/шум 0 дБ (слева), Отношение сигнал/шум 60 дБ (справа).

Из рисунка 1.66 видно, что чем больше отношение сигнал/шум, тем четче выглядит диаграмма QPSK.

Так же можно установить:

1. Сдвиг по частоте (поле Frequency Offset, Hz) в диапазоне [-1000,1000] Гц с минимальным шагом 0.01 Гц;
2. Вид фильтра (поле TX Filter). Можно задать следующие фильтры:
   - None (без фильтра);
   - Raised Cosine (Приподнятый косинус);
   - Root Raised Cos();
   - Gaussian (Гаусса).
3. Число точек M в созвездии (поле M-PSK). Можно задать следующие значения:
   - 2 (BPSK);
   - 4 (QPSK);
   - 8 (8-BSK);
   - 16 (16-BSK);
   - 32 (32-BSK);
   - 64 (64-BSK).

4. Последовательность передаваемых бит (поле Transmitted Bits).
5. Последовательность принимаемых бит (поле Received Bits).
6. Скорость передачи (поле Symbol Rate).

На рисунке 1.60 показано как изменяется диаграмма QPSK в зависимости от выбранного значения M.

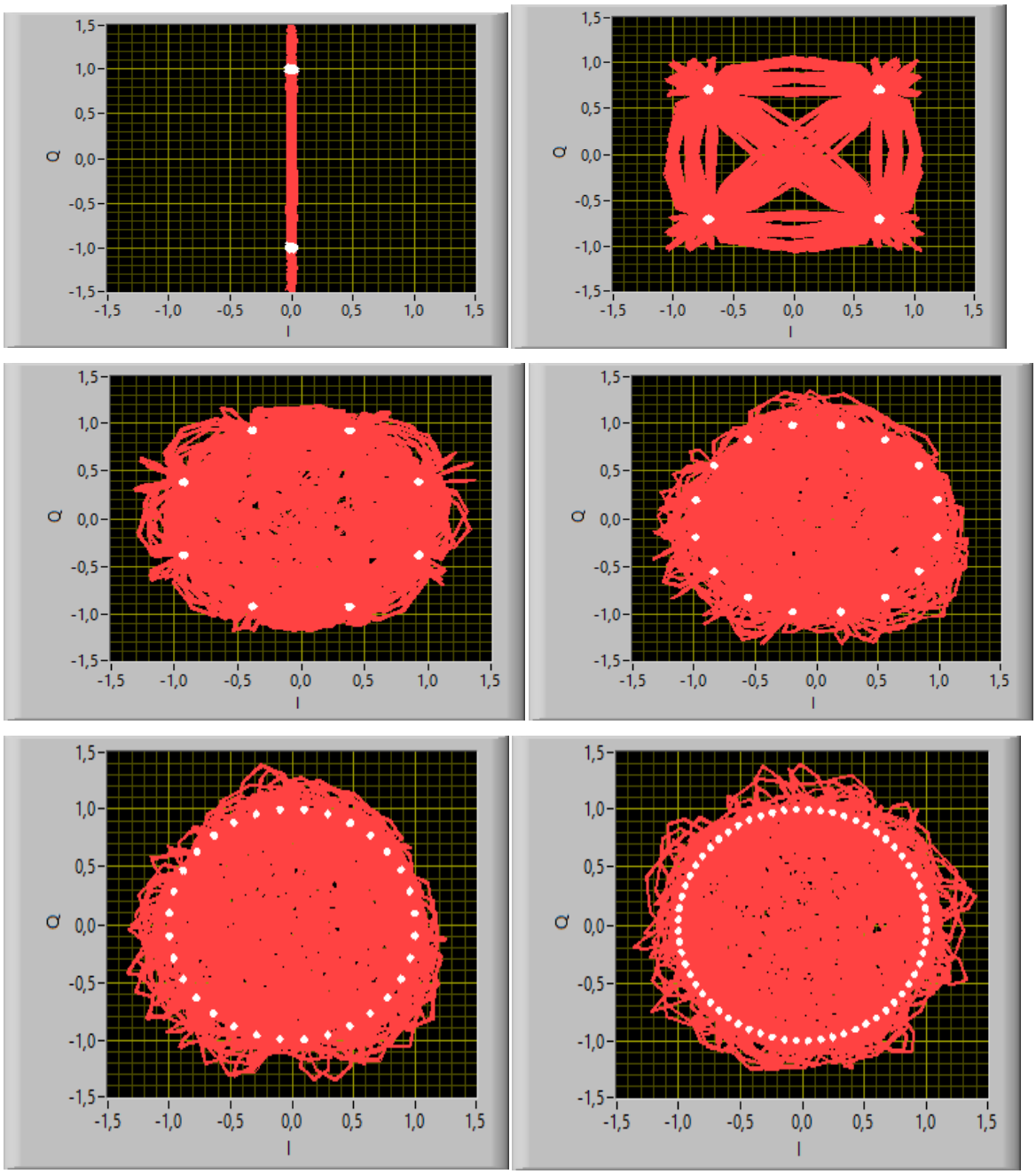


Рис. 1.67. Диаграмма QPSK. Слева на право сверху вниз - 2,4,8,16,32,64.

В разработанном программном обеспечении можно снимать спектр передатчика (вкладка TX Spectrum). На рисунке 1.67 представлены спектрограммы QPSK для разных фильтров.

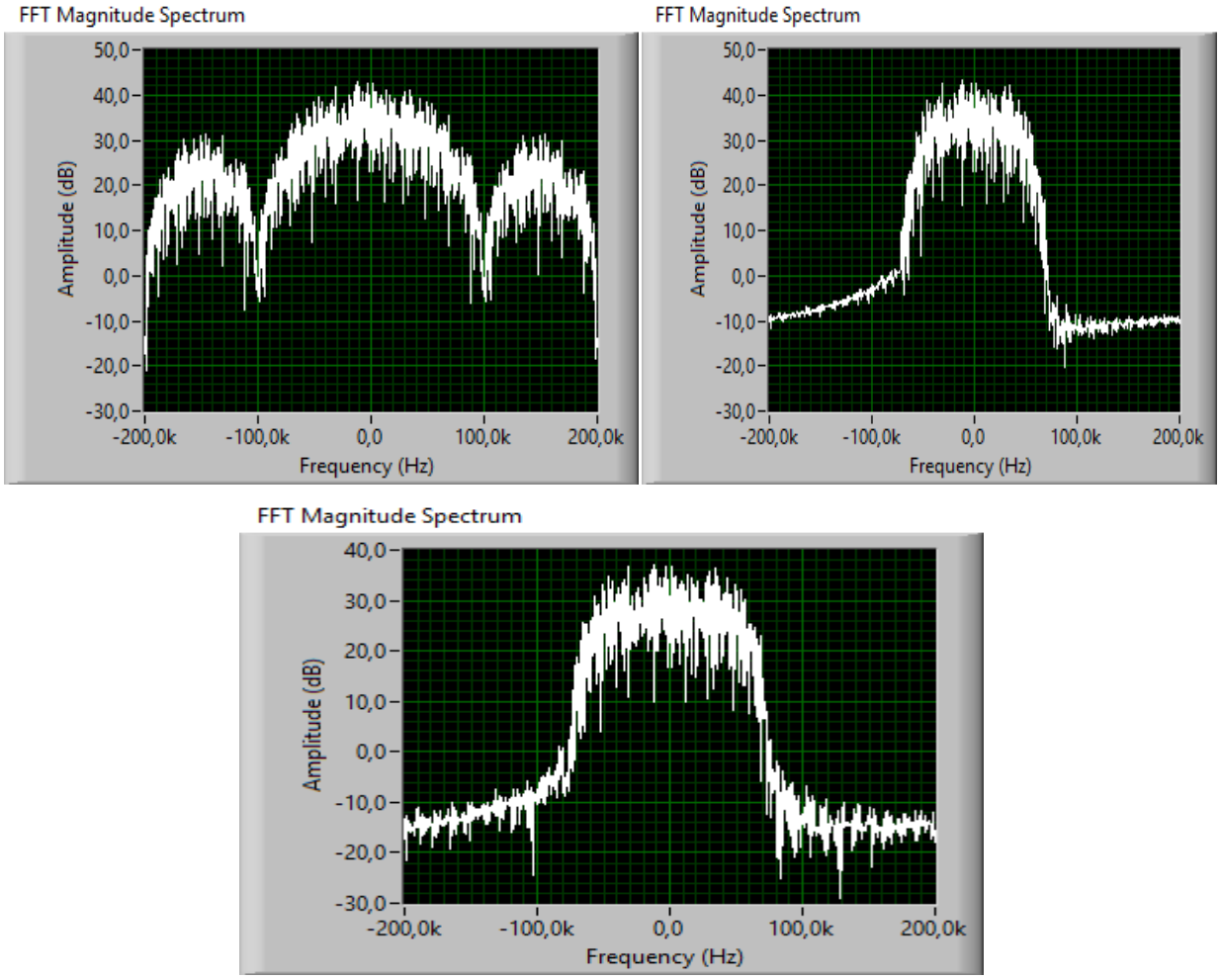


Рис. 1.68. Спектрограммы QPSK с выставленными значениями фильтра (слева на право - сверху вниз): без фильтра, приподнятый косинус, Root Raised Cos

В разработанном программном обеспечении можно снимать глазковую диаграмму с приемника (вкладка Eye Diagramm(RX)). На рисунке 1.68 представлены глазковые диаграммы при разных значения M для оси Q, а на рисунке 1.69 представлены глазковые диаграммы при разных значения M для оси I

В разработанном программном обеспечении можно снимать глазковую диаграмму с приемника (вкладка Eye Diagramm(RX)). На рисунке 17 представлены глазковые диаграммы при разных значения M.

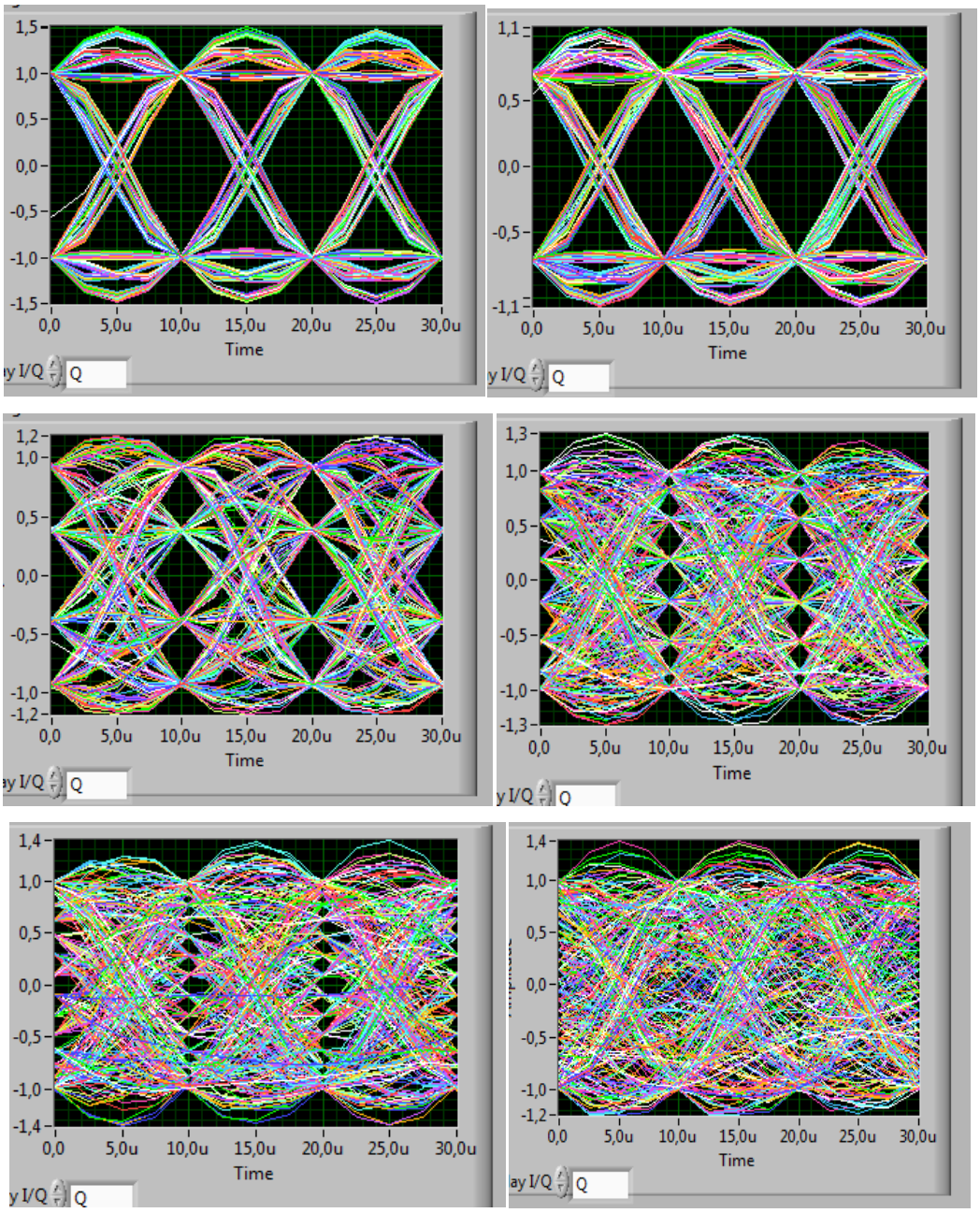


Рис. 1.69. Глазковые диаграммы для разных значений M=PSK: 2,4,8,16,32,64.

Так же программное обеспечение рассчитывает BER (Bit Error Rate – битовая вероятность ошибки). Это значение можно увидеть в поле Bit Error Rate. Быстро определить есть ошибки или нет можно если посмотреть на зеленую кнопку с подписью BER Trigger Found, если она будет гореть значит битовая вероятность ошибки присутствует. В правом нижнем углу есть информационное окно Error Out, в которое будет выводиться код и сообщение ошибки в случае не корректной работы программы.

На вкладках Transmitted Symbols и Demodulated Symbols можно посмотреть переданные и демодулированные символы соответственно.

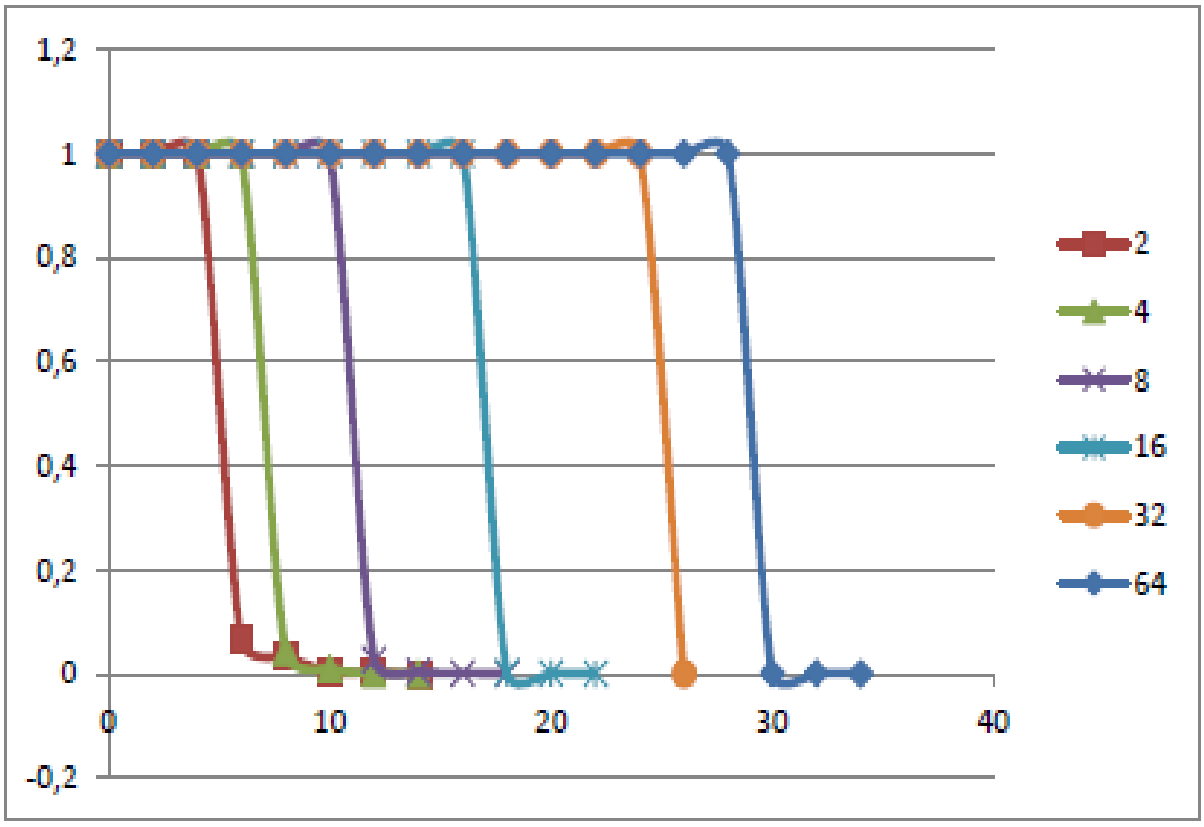


Рис. 1.70. График зависимости вероятности ошибки от отношения сигнал\шум для разных модуляций:
2 - BPSK, 4 - QPSK, 8 - 8-PSK;16 - 16-PSK, 32 - 32-PSK, 64 - 64-PSK

По графику видно, что вероятность ошибки уменьшается с уменьшением отношения сигнал-шум, а также при меньшем значении M.

Ось выбирается на вкладке Eye Diagramm (RX) в поле Display I/Q.

Так же программное обеспечение рассчитывает BER (Bit Error Rate – битовая вероятность ошибки). Это значение можно увидеть в поле Bit Error Rate.

Быстро определить есть ошибки или нет можно если посмотреть на зеленую кнопку с подписью BER Trigger Found, если она будет гореть значит битовая вероятность ошибки присутствует.

В правом нижнем углу есть информационное окно Error Out, в которое будет выводиться код и сообщение ошибки в случае не корректной работы программы.

На вкладках Transmitted Symbols и Demodulated Symbols можно посмотреть переданные и демодулированные символы соответственно.

Квадратурно-амплитудная модуляция - QAM (QAM – Quadrature Amplitude Modulation) может рассматриваться как расширенная многоуровневая ФМ, в которой два исходных сигнала генерируются независимо. Таким образом, здесь имеют место два

полностью независимых квадратурных канала, включающие процессы кодирования и детектирования в основной полосе.

На рисунке показано сигнально - точеное пространство для системы с 16-QAM и четырьмя уровнями в каждом квадратурном канале. Точки представляют составной сигнал, а штрихи на осях отмечают уровни амплитуды в каждом квадратурном канале. Основная схема модулятор – демодулятора 16-QAM представлена на рисунке .

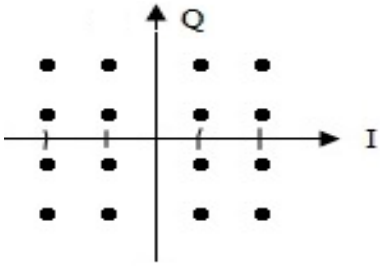


Рис.1.71. Сигнально-точечное пространство модуляции для 16-QAM

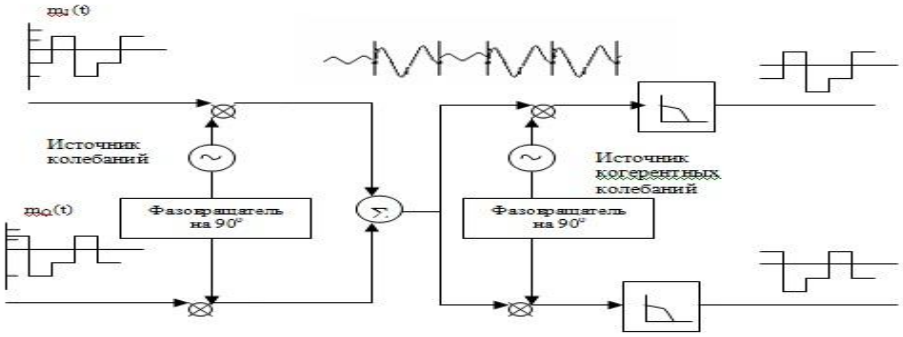


Рис. 1.72. Схема модулятора-демодулятора QAM

В отличие от ФМ сигналов сигналы QAM, показанные на рисунке не содержат постоянной огибающей. Наличие постоянной огибающей в ФМ объясняется поддержанием отношения уровней в квадратурных каналах. В QAM такие ограничения не вводятся ввиду того, что в каждом канале уровни независимы.

Характеристики ошибок систем QAM и ФМ модуляций сильно отличаются. При достаточно большом числе сигнальных точек системы QAM имеют, как правило, лучшие характеристики, чем системы с ФМ. Основная причина состоит в том, что расстояние между сигнальными точками на диаграмме для системы с QAM больше, чем для соответствующей системы с ФМ.

Расстояние d между соседними точками в системе QAM с нормированной к единице пиковой амплитудой и числом уровней L может быть представлено в виде:

$$d = \frac{\sqrt{2}}{L-1} \tag{1.14}$$

На рисунке представлено сравнение систем QAM-16 и ФМ-16 работающих на одинаковой пиковой мощности, по расстоянию между точками.

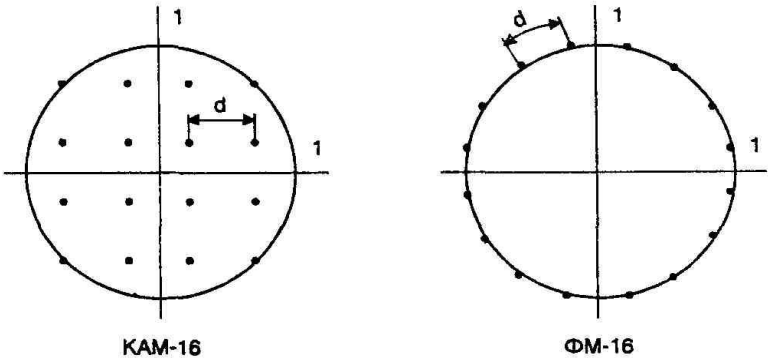


Рис. 1.73. Сравнение систем QAM-16 и ФМ-16 работающих на одинаковой пиковой мощности, по расстоянию между точками

QAM имеет преимущество над системой ФМ при той же пиковой мощности.

В настоящее время для передачи пользуются системами 256-QAM. Надо отметить, что надежное функционирование высокоплотных форматов модуляции, таких как 256- QAM требует строгой линейности усилителей, для возможности обработки широкого диапазона амплитуд сигналов. Соотношения для характеристик ошибок методов 4-,16-,64- и 256 - QAM в зависимости от отношения функции $\frac{E_b}{N_0}$ приведены на рисунке.

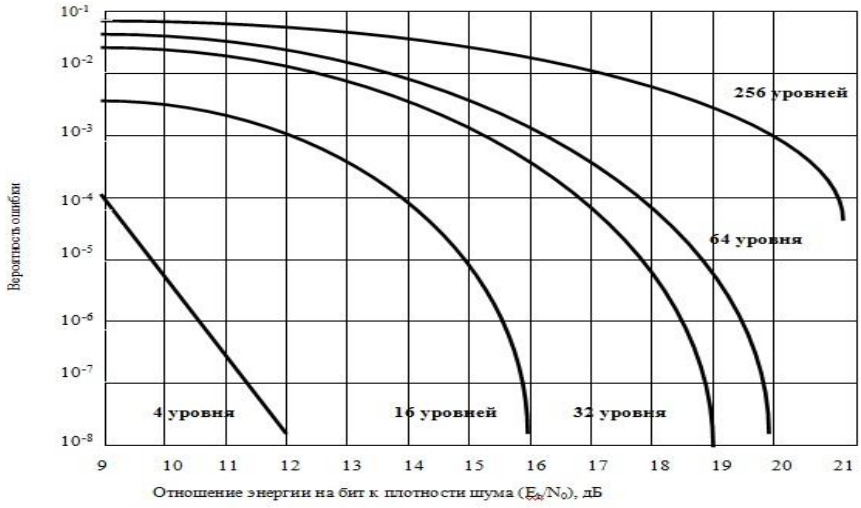


Рис. 1.74. Вероятности ошибок в системах QAM.

Достоинство высоких значений номера QAM – это повышенная скорость передачи данных, поскольку таким образом большее количество битов информации может быть передано в течении одного цикла. Однако, с другой стороны, в этом случае большее число уровней амплитуды сигнала располагаются близко друг к другу, повышая тем самым вероятность неразличимости двух уровней, и как следствие – повышая чувствительность системы к шуму. Таким образом, высокие значения номера QAM более требовательны к параметру SNR (Signal Noise Ratio – Отношение Сигнал/Шум).

Практическая часть QAM

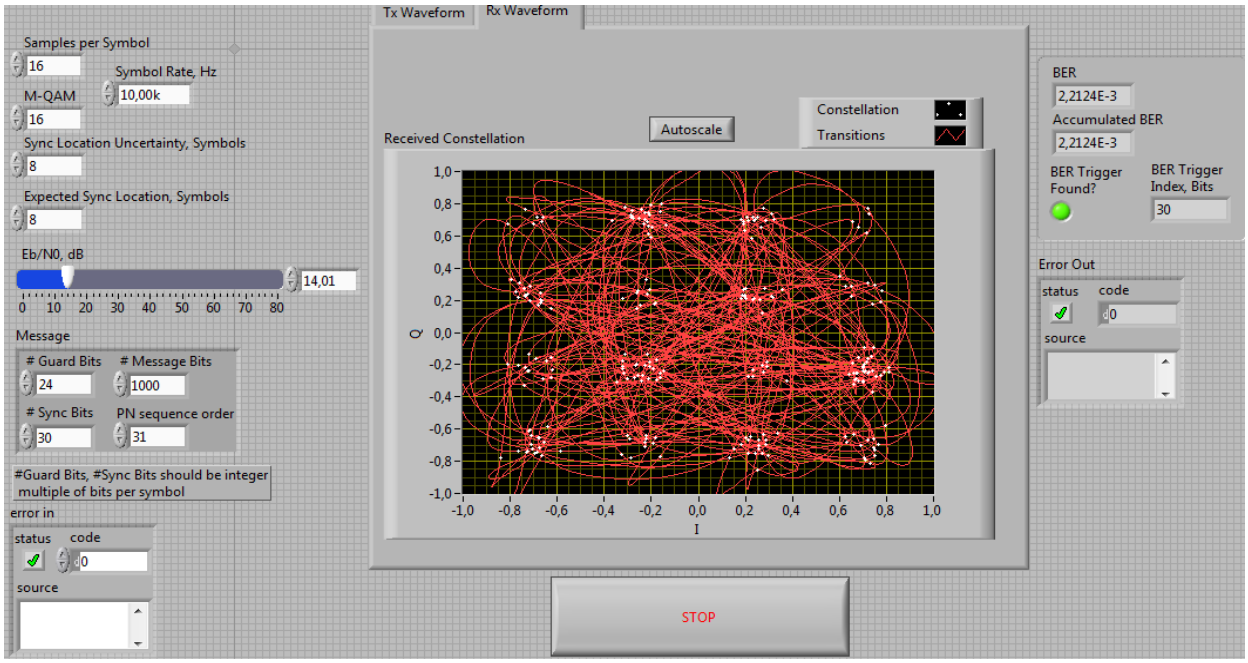


Рис. 1.75. Внешний вид разработанного ПО для исследования QAM

Число посылок 2000 и различные отношения сигнал/шум

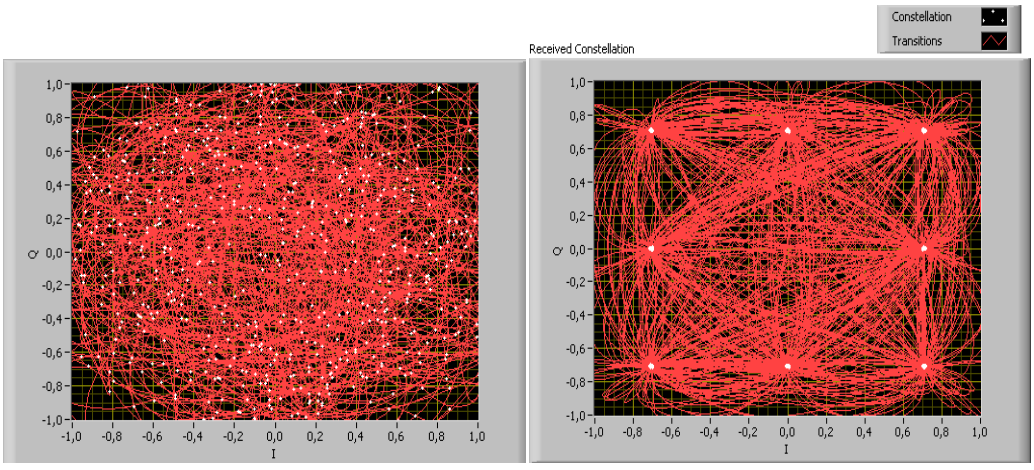


Рис. 1.76. Созвездия для QAM-8 передаваемого сигнала.

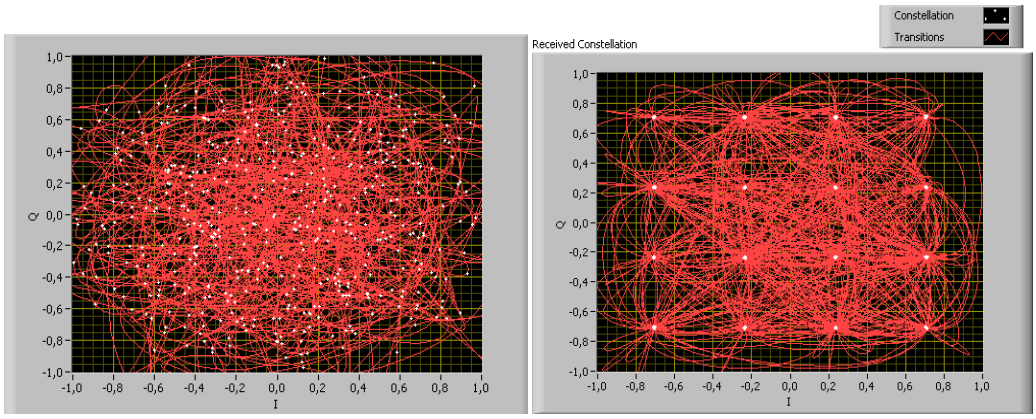


Рис. 1.77. Созвездия для QAM-16 передаваемого сигнала.

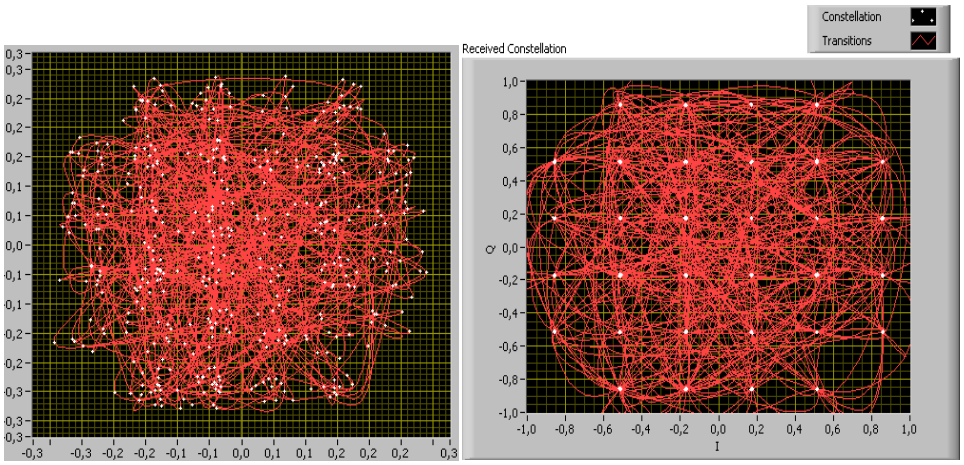


Рис. 1.78 Созвездия для QAM-32 передаваемого сигнала.

На рисунке 1.79. приведены глазковые диаграммы для QAM-8 сигнала.

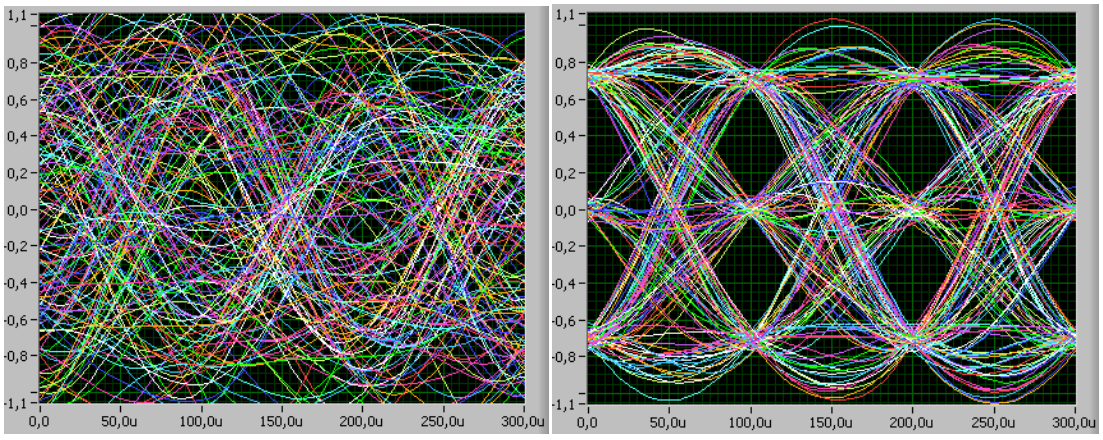


Рис. 1.79. Глазковые диаграммы при малом отношении сигнал/шум и при наилучшем отношении сигнал/шум

На рисунке 1.80. приведены глазковые диаграммы для QAM-16 сигнала.

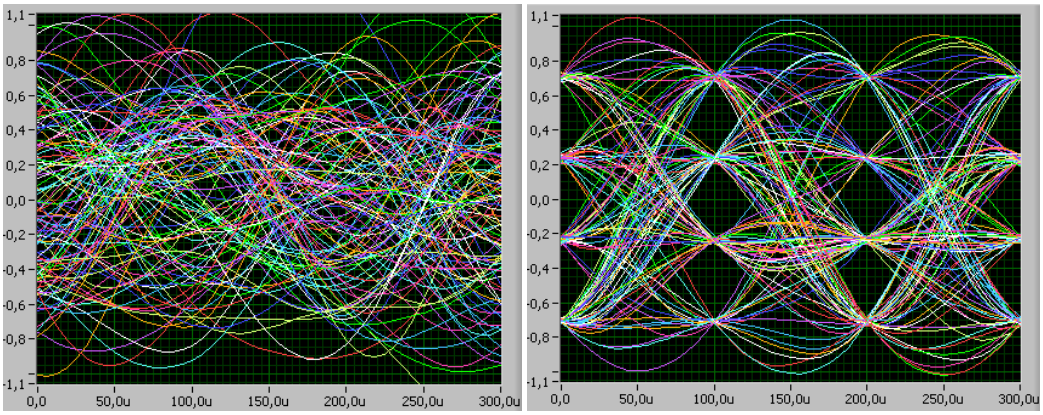


Рис. 1.81. Глазковые диаграммы при малом отношении сигнал/шум и при наилучшем отношении сигнал/шум

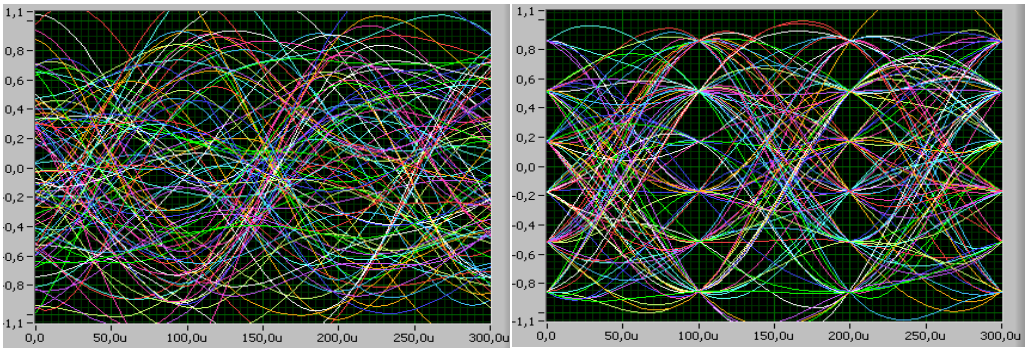


Рис. 1.82. Глазковые диаграммы при малом отношении сигнал/шум и при наилучшем отношении сигнал/шум

На рисунке 1.83 приведены спектрограммы для QAM-8 сигнала.

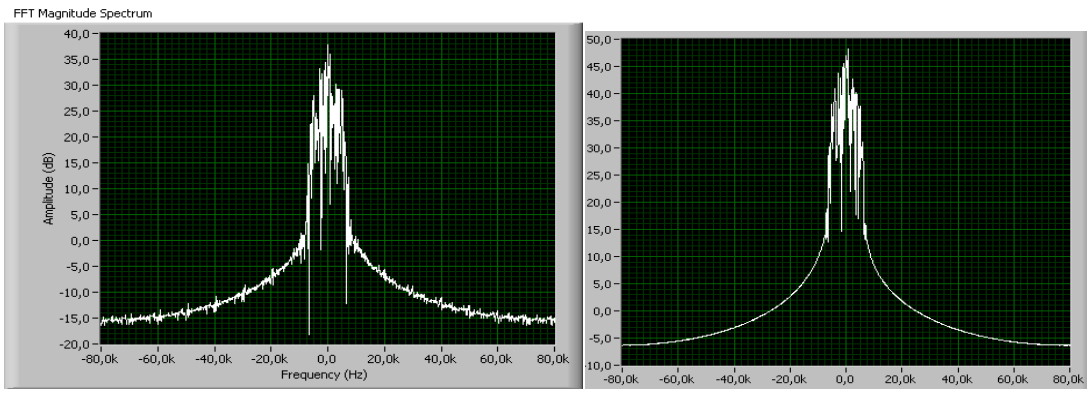


Рис. 1.83. Спектрограммы на входе и на выходе канала

На рисунке 1.84. приведены спектрограммы для QAM-16 сигнала.

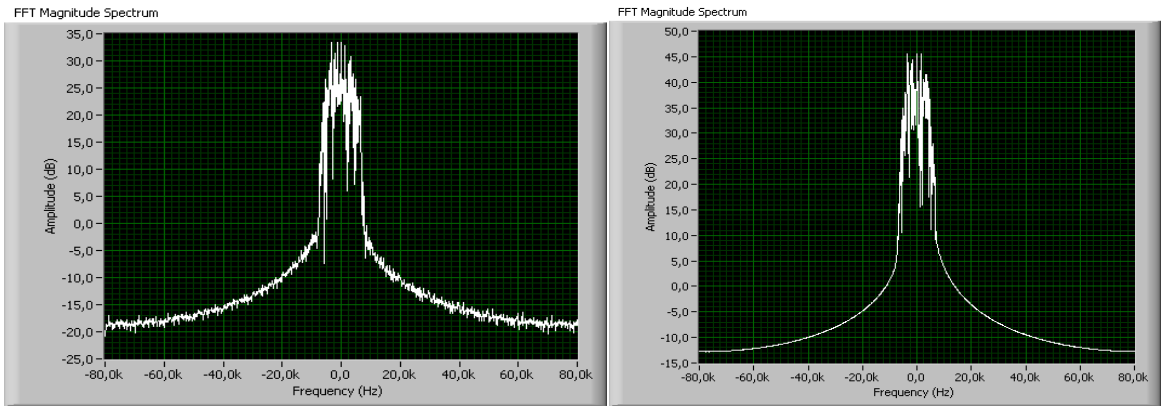


Рис. 1.85. Спектрограммы на входе и на выходе канала

На рисунке 1.86 приведены спектрограммы для QAM-32 сигнала.

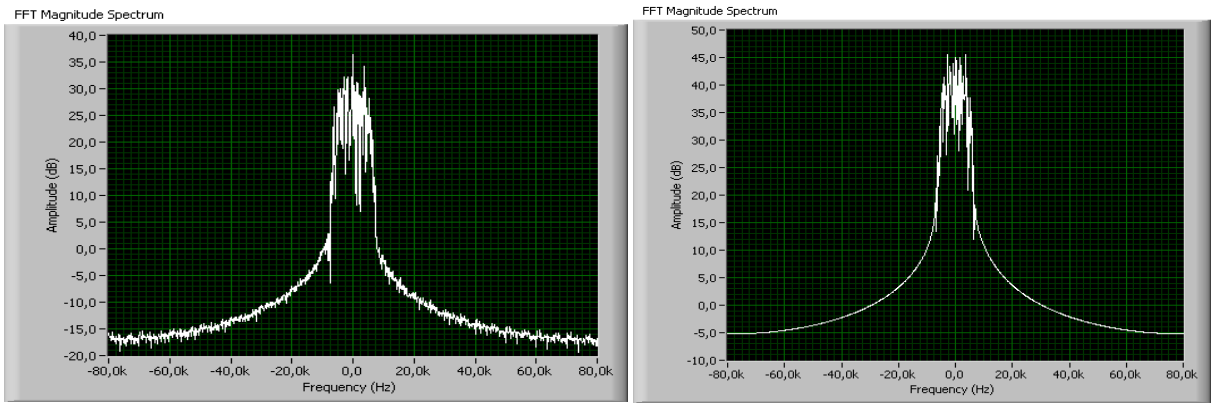


Рис. 1.86. Спектрограммы на входе и на выходе канала

Ползунком Eb/N0 устанавливается уровень отношения сигнал/шума, в поле BER отображается количество обнаруженных ошибок при передаче. Из полученных данных можно построить график зависимости, сравнить показатели у различных видов модуляции и подтвердить/опровергнуть теорию, описанную выше.

В таблице ниже представлены данные для рассмотренных видов модуляций.

BER	1	1	1	1	1	1,00E+00	1,19E-02	3,97E-03	7,94E-03	1,98E-03
BER	1	1	1	1	4,51E-01	1,00E+00	1,72E-02	7,81E-03	2,08E-03	2,08E-03
BER	1	1	1	1	1,00E+00	2,27E-01	2,42E-02	2,10E-03	5,25E-04	5,25E-04
Eb/N0, dB	0	2	4	5	6	8	10	12	14	16

На рисунке представлен график зависимости параметра BER от Eb/N0 для 8-QAM, 16-QAM, 32-QAM

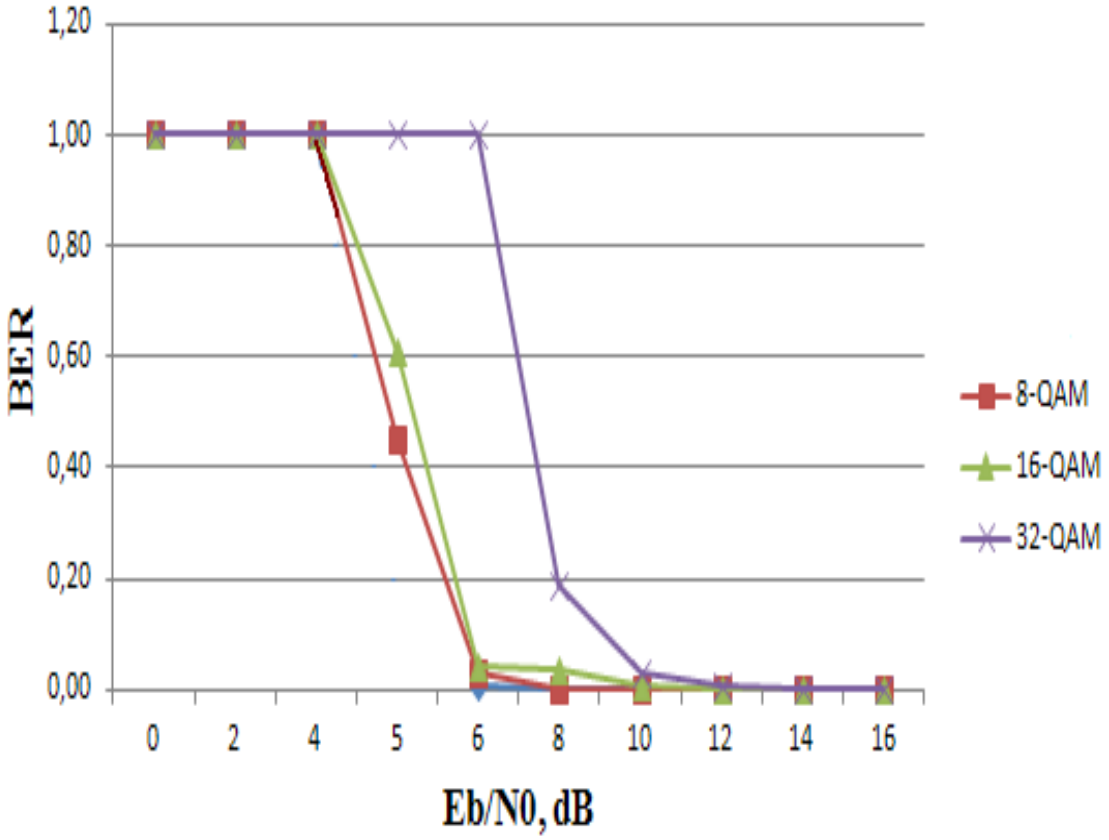


Рис. 1.87. График зависимости параметра BER от Eb/No для 8-QAM, 16-QAM, 32-QAM

ГЛАВА 2. ПРОПУСКНАЯ СПОСОБНОСТЬ КАНАЛА СВЯЗИ. КОДИРОВАНИЕ ИСТОЧНИКА

2.1. Пропускная способность канала связи. Объем сигнала и емкость канала связи, условия их согласования

Рассматривается вопрос согласования дифференциальных характеристик источника дискретной информации (ИДИ) и предоставленного дискретного канала связи (КС) в терминах потока информации, выводимой из ИДИ, и пропускной способности КС. В данном разделе эта задача решается на уровне интегральных характеристик сигнала и канала в виде объема сигнала и емкости канала связи.

Сигнал как модель сообщения (информации) имеет «габаритные размеры», характеризующие объем сигнала, аналитически определяемый выражением

$$V_C = T_C F_C W_C = T_C F_C \log_2(1 + \frac{P_C}{P_n}) \qquad (2.1)$$

где T_c - временная длительность сигнала, F_c - эффективный спектр сигнала, определяемый эффективным спектра элементарного сигнала кода и типом модуляции. В (2.1) компонент $W_C = \log_2(1+\frac{P_C}{P_n})$ именуется логарифмическим превышением сигнала над помехой, в котором P_c - мощность сигнала, P_n - мощность помехи, сопровождающей процесс формирования сигнала.

Аналогичным образом канал связи как транспортная среда характеризуется емкостью канала связи, аналитически задаваемую выражением

$$V_k = T_k F_k W_k = T_k F_k \log_2(1+\frac{P_C}{P_n}), \qquad (2.2)$$

где $T_к$ - длительность интервала времени, на который предоставлен канал связи, $F_к$ - эффективная полоса пропускания канала связи, которая может быть определена аналитически или экспериментально по амплитудной частотной характеристике четырехполюсника, который представляет собой канал связи. В (2.2) компонент $W_k = \log_2(1+\frac{P_C}{P_n})$ именуется логарифмическим превышением сигнала помехой, в котором P_c - мощность сигнала, фиксируемая в канальной среде, P_n - мощность помехи в канальной среде.

Нетрудно понять, что передача сигнала по предоставленному каналу связи возможна только тогда, когда размеры транспортируемого средства (сигнала) не превышают размеров транспортной среды (канала связи).

Таким образом необходимым условием согласования сигнала с предоставленным каналом связи является выполнение неравенства

$$V_C \le V_k. \qquad (2.3)$$

Достаточными условиями согласования сигнала с предоставленным каналом связи является выполнение неравенств

$$T_C \le T_k, \quad F_C \le F_k \quad W_C \le W_k, \qquad (2.4)$$

И наконец, условием эффективного использования предоставленного канала связи является выполнение равенства

$$V_C = V_k. \qquad (2.5)$$

Конструктивным инструментом согласования сигнала с предоставленным каналом связи путем уменьшения объема сигнала V_c за счет уменьшения компонента T_c является использование возможностей эффективного кодирования.

Теорема К.Шеннона об эффективном кодировании символов ИДИ (основная теорема К.Шеннона)

Пусть источник дискретной информации (ИДИ) генерирует алфавит $X = \{x_i : p(x_i); i = \overline{1,n}\}$ составленный из n дискретных статистически независимые

$$H(X) = -\sum_{i-1}^{n} p(x_i) \log_2 p(x_i)$$

символов, характеризующийся энтропией , тогда:

1. существует такой способ кодирования символов, при котором среднее на символ

$$l_{cp}[K(X)] = \sum_{i-1}^{n} p(x_i) l[K(x_i)]$$

число двоичных разрядов кода будет сколь угодно близким к энтропии $H(X)$ ИДИ;

2. не существует такого способа кодирования, при котором среднее на символ число двоичных разрядов кода $l_{cp}[K(X)] = \sum_{i-1}^{n} p(x_i) l[K(x_i)]$ будет меньше энтропии *H(X)* ИДИ.

Наиболее распространенными алгоритмами эффективного кодирования, основанного на использовании основной теоремы К.Шеннона, являются:

- Алгоритм К.Шеннона - Р.Фано,
- Алгоритм Д. Хаффмэна.

Следует сказать, что содержательный момент этих алгоритмов, несмотря на процедурные различия, позволяющий построить эффективный двоичный код, средняя на символ длина которого максимально приближена к энтропии источника, состоит в том, что символам с большей вероятностью появления на выходе ИДИ ставятся в соответствие коды меньшей длины, а символам с меньшей вероятностью - коды большей длины. Таким образом, эффективное кодирование реализуется в классе неравномерных кодов.

Эффективные коды принято характеризовать двумя показателями кода: избыточность и эффективность

Определение 2.1. Избыточностью эффективного кода называется показатель D_k, определяемый выражением

$$D_k = \frac{l_{cp}[K(X)] - l_{\min}[K(X)]}{l_{cp}[K(X)]} = \frac{\sum_{i=1}^{n} p(x_i) l[K(x_i)] - H(X)}{\sum_{i-1}^{n} p(x_i) l[K(x_i)]}. \qquad (2.6)$$

Определение 2.2. Эффективностью эффективного кода называется показатель η, определяемый выражением

$$\eta = \frac{l_{\min}[K(X)]}{l_{cp}[K(X)]} = \frac{H(X)}{\sum_{i-1}^{n} p(x_i) l[K(x_i)]}. \qquad (2.7)$$

Нетрудно видеть, что избыточность и эффективность эффективного кода связаны соотношением

$$D_k = 1 - \eta. \qquad (2.8)$$

2.2. Исследование кодирования источника дискретных сообщений методам Шеннона Фано

Для удобства расположим все имеющиеся п букв в один столбик в порядке убывания вероятностей. Затем все эти буквы следует разбить на две группы – верхнюю и нижнюю – так, чтобы суммарная вероятность первой группы была наиболее близка к суммарной вероятности второй группы. Для букв первой группы в качестве первой цифры кодового обозначения используется цифра 1, а для букв второй группы – цифра 0. Далее, каждую из двух групп подобным образом снова надо разделить на две части и в качестве второй цифры кодового обозначения мы будем использовать цифру 1 или 0 в зависимости от того, принадлежит ли наша группа к первой или ко второй из этих подгрупп. Затем, каждая из содержащих более одной буквы групп снова делится на две части возможно более близкой суммарной вероятности и т.д.; процесс повторяется до тех пор, пока мы не придем к группам, каждая из которых содержит по одной единственной букве.

Например, если наш алфавит содержит всего шесть букв, вероятность которых (в порядке убывания) равны 0,4, 0,2, 0,2, 0,1, 0,05 и 0,05, то на первом этапе деления букв на группы мы отщепим лишь одну первую букву (1-я группа), оставив во второй группе все остальные. Далее, вторая буква составит 1-ю подгруппу 2-й группы; 2-я же подгруппа той же группы, состоящая из оставшихся четырех букв, будет и далее последовательно делиться на части так, что каждый раз 1-я часть будет состоять лишь из одной буквы.

Таблица 2.1.

№ буквы	вероятность	разбиение на подгруппы (римские цифры обозначают номера групп и подгрупп)					кодовое обозначение
1	0,4	} 1					1
2	0,2	} 0	} 1				01

3	0,2			} 1			001
4	0,1		} 0		} 1		0001
5	0,05			} 0		} 1	00001
6	0,05				} 0	} 0	00000

Основной принцип, положенный в основу кодирования по методу Шеннона – Фано, заключается в том, что при выборе каждой цифры кодового обозначения мы стараемся, чтобы содержащееся в ней количество информации было наибольшим, т. е. чтобы независимо от значений всех предыдущих цифр, эта цифра принимала оба возможных для нее значения 0 и 1 по возможности с одинаковой вероятностью.

Процесс декодирования

Теперь рассмотрим алгоритм декодирование кодов Шеннона-Фано. Процесс усложняется тем, что невозможно, как в случае кодирования, заменять каждые 8 бит входного потока, кодом переменной длины. При восстановлении исходной последовательности необходимо провести обратные операции - заменить код переменной длины символом длиной 8 бит. В данном случае, лучше всего будет использовать бинарное дерево, ячейками которого будут являться символы.

Пример кодирования сообщений

Для наглядной иллюстрации алгоритма, сделаем сжатие предложения "Карлсон, который живет на крыше.". Вычислим, сколько раз встречается каждый символ, и занесем данные в таблицу 2.2.

Таблица 2.2.

Символ	Количество появлений
Пробел	4
р	3
о	3
к	2
а	2
е	2
т	2
ы	2
н	2

К	1
й	1
л	1
в	1
и	1
ш	1
ж	1
с	1
.	1
,	

Далее, разделив таблицу на две части, таким образом, чтобы число появлений символов в верхней половине таблицы примерно равнялось общему числу в нижней половине. В предложении 32 символа, а значит, верхняя часть таблицы должна содержать около 16 появлений. Мы можем получить этот результат, поместив разделительную линию (далее р.л.) между строками "е" и "т". В итоге верхняя часть будет содержать появление 16 символов, а нижняя, соответственно, оставшиеся 16.

Далее проделаем то же с каждой из частей таблицы: размесим линии так, чтобы разделить 2 полученные части на приблизительно равные по количеству появлений. Продолжаем процесс разделения до того момента, пока каждый символ не будет отделен от другого.

Результирующее дерево.

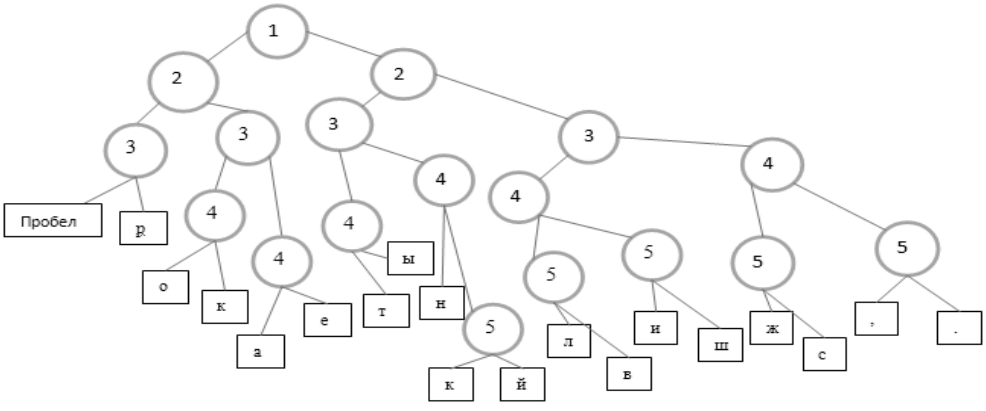


Рис.2.1. Результирующее дерево

Если принять, что перемещение влево эквивалентно нулевому биту, а вправо - единичному, можно создать таблицу кодирования.

Таблица 2.3

Си	Кодир

мвол	овка
Пр	000
обел	001
р	0100
о	0101
к	0110
а	0111
е	1000
т	1001
ы	1010
н	10110
К	10111
й	11000
л	11001
в	11010
и	11011
ш	11100
ж	11101
с	11110
.	11111
,	

Таблица содержит 137 бит, а оригинальная фраза "Карлсон, который живет на крыше." занимает 256 бит, следовательно, у нас получается коэффициент сжатия 53%.

Практическая часть

1. Запустить программный комплекс [6].

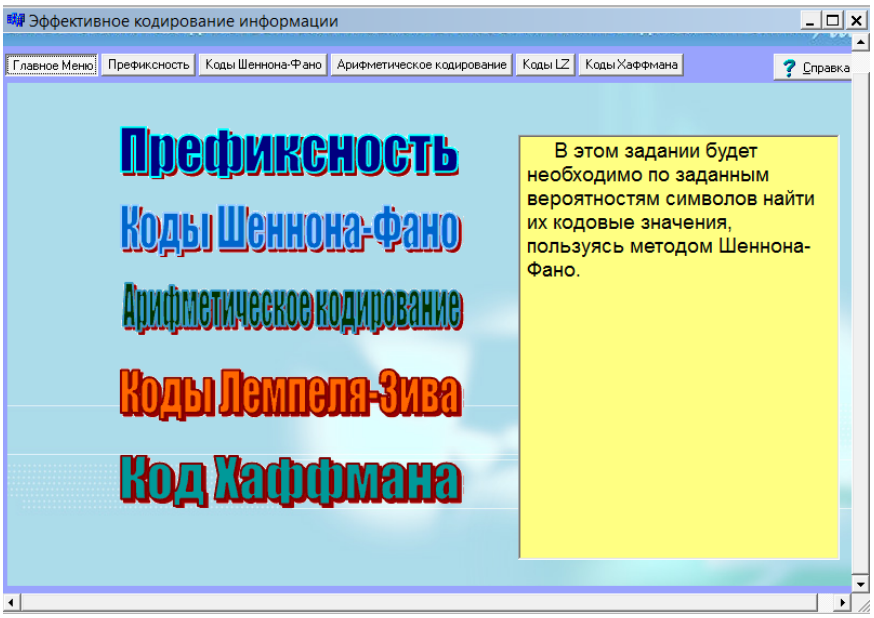
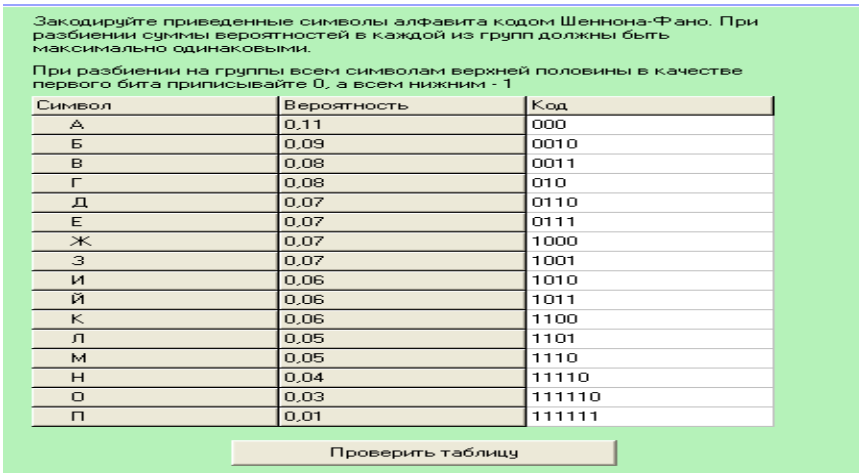


Рис. 2.2. Панель программного комплекса

2. Выбрали в меню пункт задания «Коды Шеннона-Фано». Изучили предложенную таблицу 16-символьного алфавита. Символы отсортированы в порядке убывания вероятностей их появления.

3. Пользуясь теоретическим материалом, найти двоичные коды для каждого из предложенных символов, записали их в графу Код кодовой таблицы. Следуя правилу приписывания очередного символа, указанному в окне программы над кодовой таблицей.



Символ	Вероятность	Код
А	0,11	000
Б	0,09	0010
В	0,08	0011
Г	0,08	010
Д	0,07	0110
Е	0,07	0111
Ж	0,07	1000
З	0,07	1001
И	0,06	1010
Й	0,06	1011
К	0,06	1100
Л	0,05	1101
М	0,05	1110
Н	0,04	11110
О	0,03	111110
П	0,01	111111

Рис. 2.3. Кодовая таблица для данного сообщения

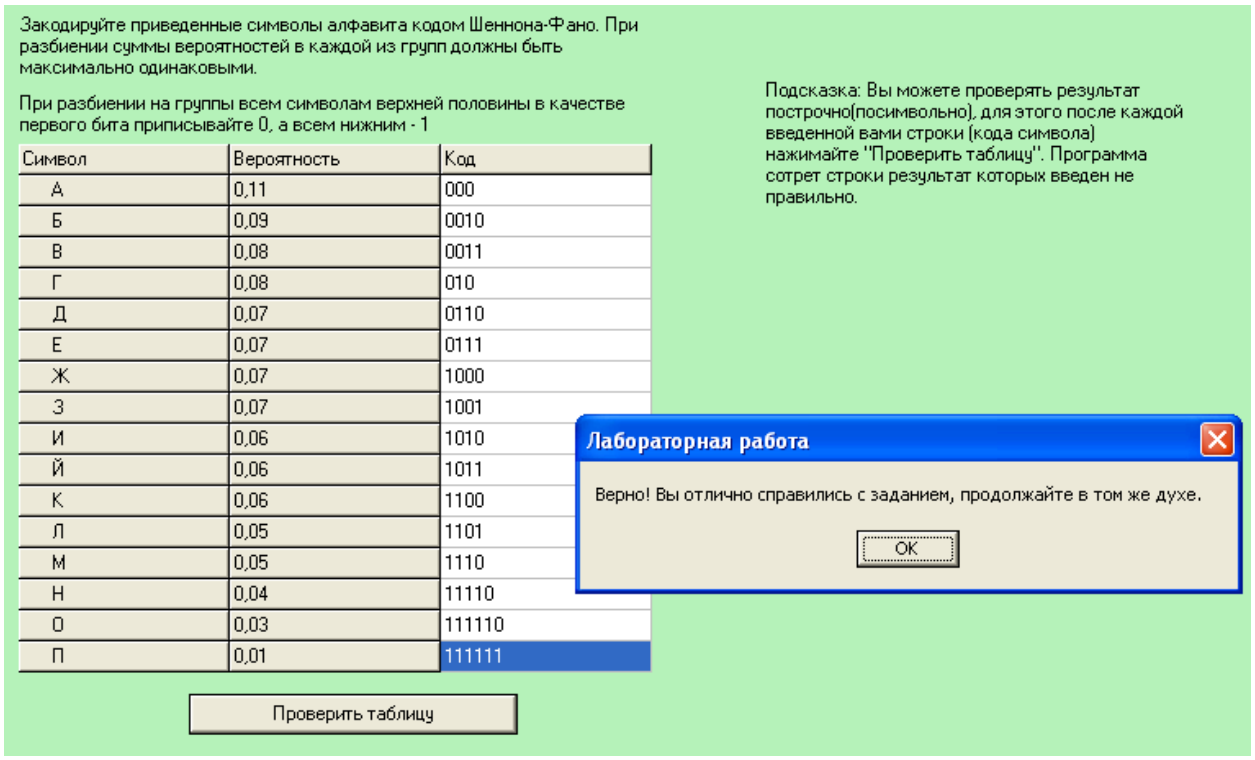


Рис. 2.4. Процесс проверки кодовой таблицы

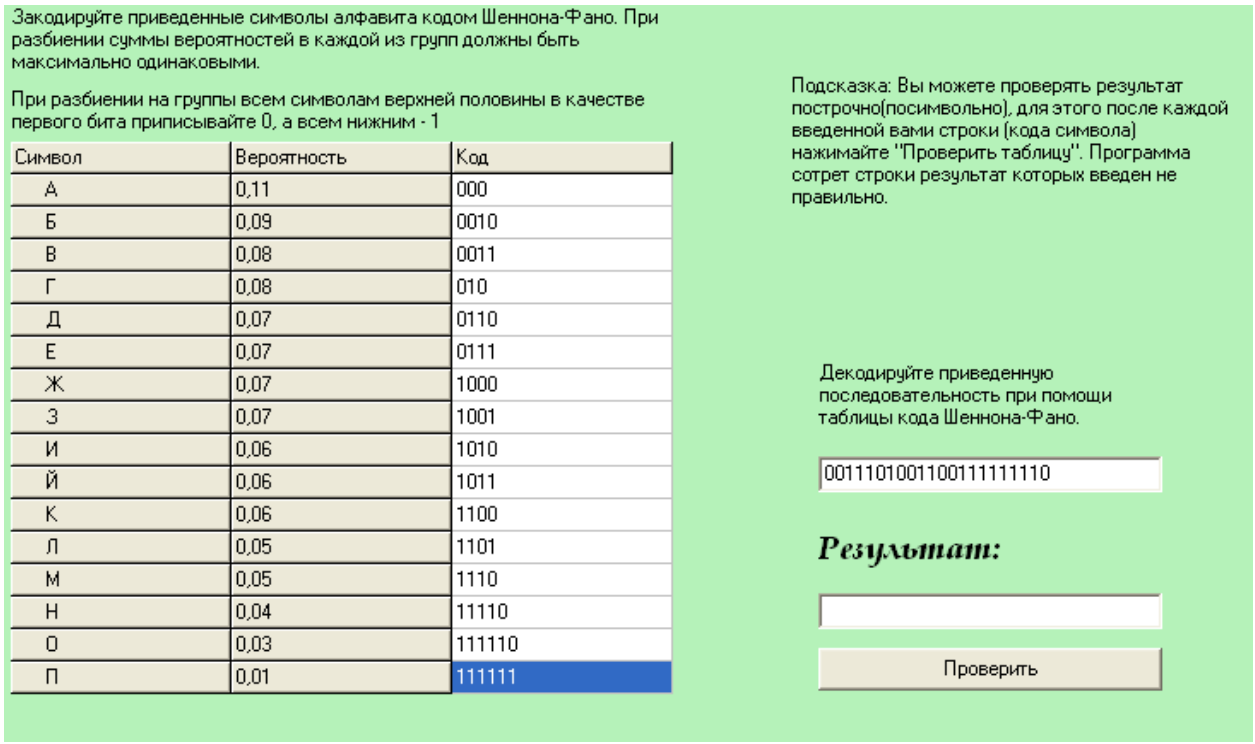


Рис. 2.5. Закодированная последовательность

4. Декодировали при помощи составленной таблицы предложенной последовательности символов. Ответ записали в поле результат. По завершении проверили.

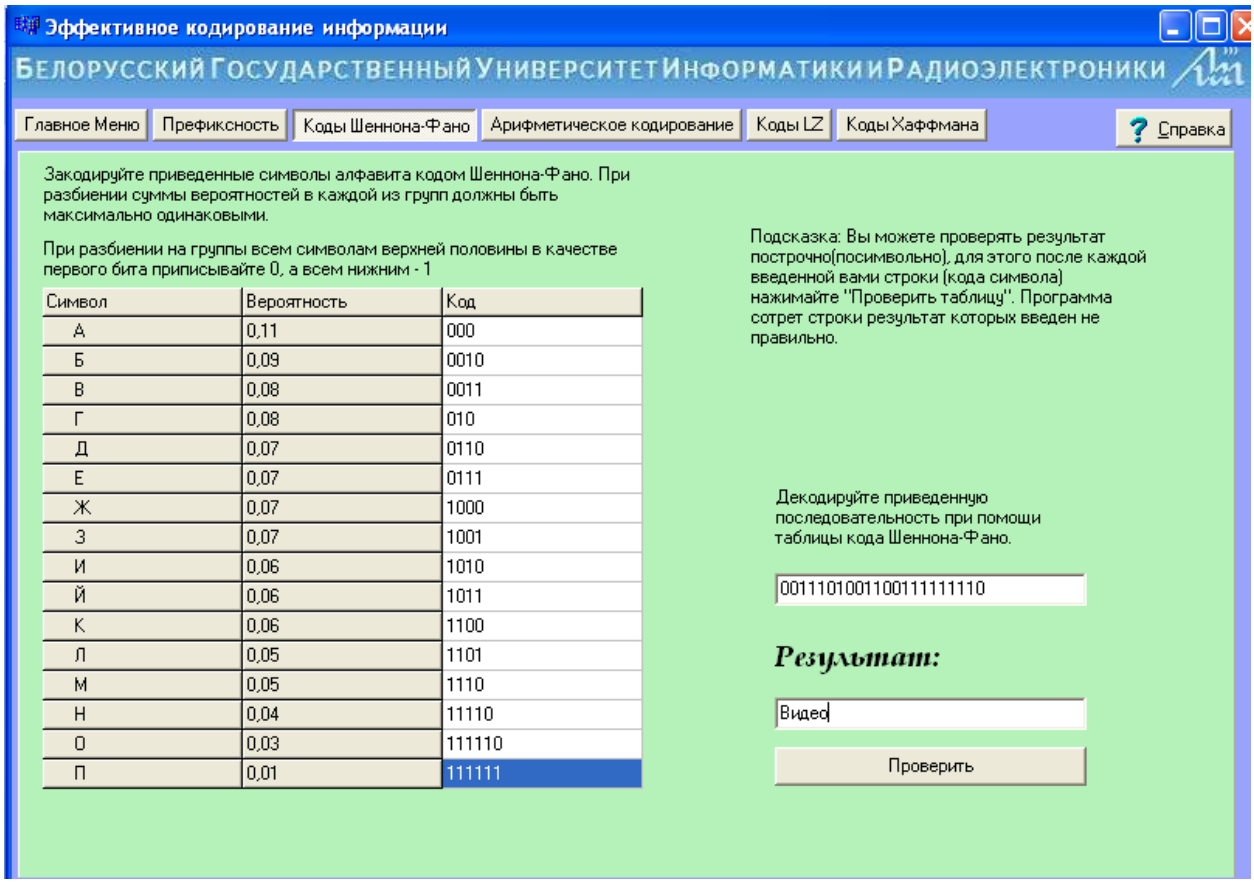


Рис. 2.6. Декодирование закодированной последовательности

Полученную последовательность в соответствии с таблицей символов декодировали самостоятельно и в результате получили слово: «Видео».

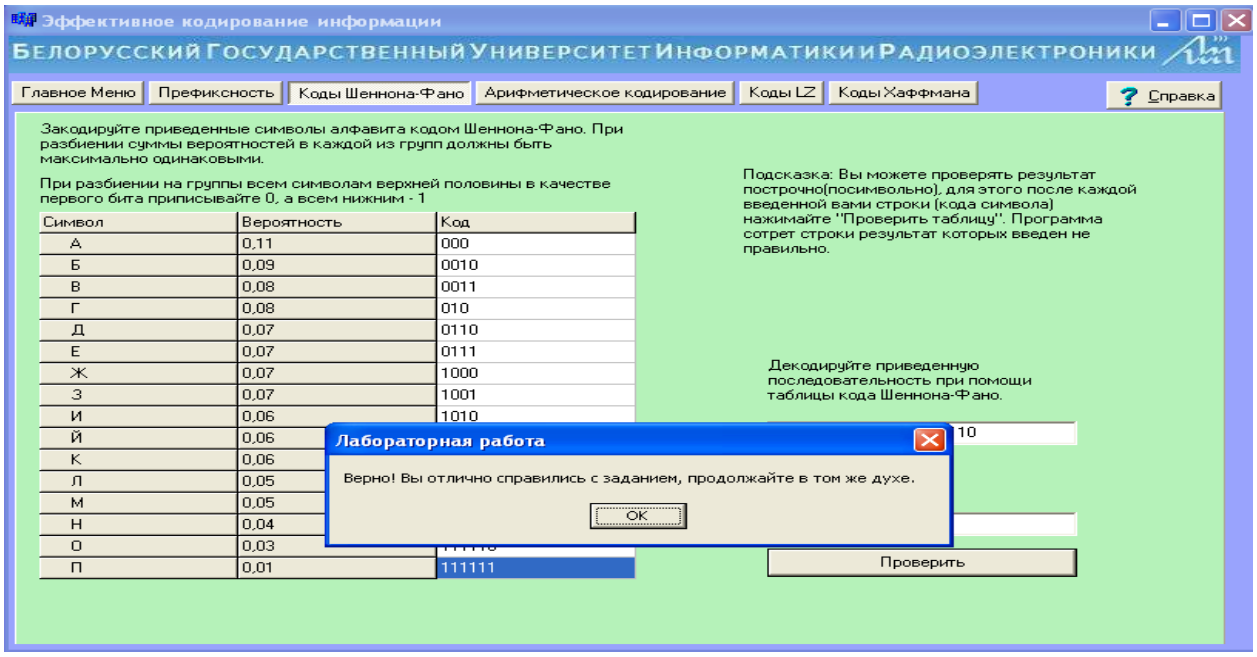


Рис. 2.7. Результат проверки

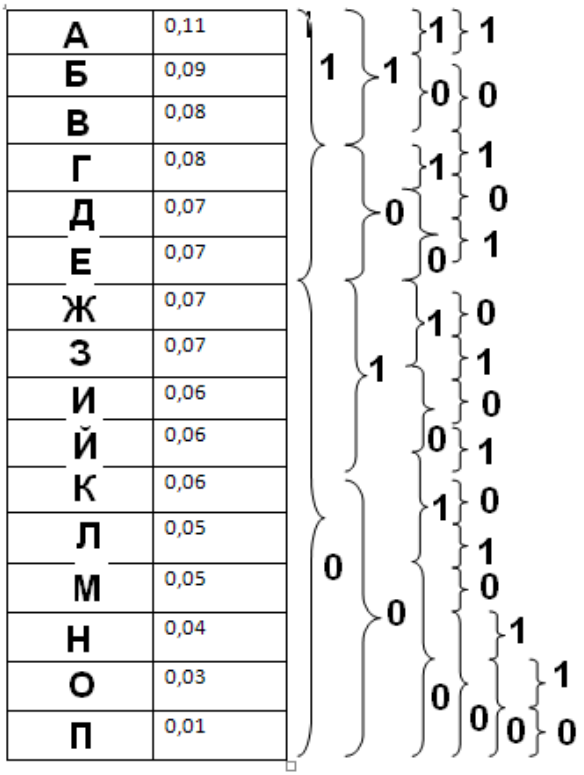


Рис. 2.8. Кодовое дерево

Преимуществом данного метода является его простота и, как следствие этого, высокая скорость кодирования и декодирования. Данный метод является простым в понимании, в реализации, а как следствие довольно эффективным. В настоящее время не используется, так как был усовершенствован Хаффманом, алгоритм которого в данный момент широко используется в связках с другими алгоритмами при кодировании/декодировании информации.

2.3. Исследование алгоритма Лемпеля - Зива

Основная идея алгоритма Лемпеля-Зива состоит в замене появления фрагмента в данных (группы байт) ссылкой на предыдущее появление этого фрагмента. Существуют два основных класса методов, названных в честь Якоба Зива (Jakob Ziv) и Абрахама Лемпеля (Abraham Lempel), впервые предложивших их в 1977 (алгоритм LZ77) и 1978 годах (алгоритм LZ78).

Алгоритм LZ77

LZ77 использует уже просмотренную часть сообщения как словарь. Чтобы добиться сжатия, он пытается заменить очередной фрагмент сообщения на указатель в содержимое словаря.

В качестве модели данных LZ77 использует “скользящее” по сообщению окно, разделенное на две неравные части. Первая, большая по размеру, включает уже

просмотренную часть сообщения. Вторая, намного меньшая, является буфером, содержащим еще не закодированные символы входного потока. Обычно размер окна составляет несколько килобайтов. Буфер намного меньше, обычно не более ста байтов. Алгоритм пытается найти в словаре фрагмент, совпадающий с содержимым буфера.

Алгоритм LZ77 выдает коды, состоящие из трех элементов:

- смещение в словаре относительно его начала подстроки, совпадающей

с содержимым буфера;

- длина подстроки;

- первый символ в буфере, следующий за подстрокой.

Пример. Размер окна – 20 символов, словаря – 12 символов, а буфера -8. Кодируется сообщение «ПРОГРАММНЫЕ ПРОДУКТЫ ФИРМЫ MICROSOFT». Пусть словарь уже заполнен. Тогда он содержит строку «ПРОГРАММНЫЕ_», а буфер строку «ПРОДУКТЫ». Просматривая словарь, алгоритм обнаружит, что совпадающей подстрокой будет «ПРО», в словаре она расположена со смещением 0 и имеет длину 3 символа, следующим символом в буфере является «Д». Таким образом, выходным кодом будет тройка (0, 3, «Д»). После этого алгоритм сдвигает влево все содержимое окна на длину совпадаю-щей подстроки +1 и одновременно считывает столько же символов из входного потока в буфер. Получаем в словаре строку «РАММНЫЕ ПРОД», в буфере – «УКТЫ ФИР». В данной ситуации совпадающей подстроки обнаружить не удастся и алгоритм выдаст код (0, 0, «У»), после чего сдвинет окно на один символ. Затем словарь будет содержать «АММНЫЕ ПРОДУ», а буфер – «КТЫ ФИРМ». и т.д.

Проблемы LZ77

Очевидно, что быстродействие кодера LZ77 сильно зависит от того, каким образом будет осуществляться поиск совпадающей подстроки в словаре. Если искать совпадение полным перебором всех возможных вариантов, то очевидно, что сжатие будет очень медленным. Причем при увеличении размеров окна для повышения степени сжатия скорость работы будет пропорционально уменьшаться. Для декодера это неважно, так как при декодировании не осуществляется поиск совпадения.

Быстродействие и кодера, и декодера зависит от того, как реализовано “скольжение” окна по содержимому сообщения. Рационально было бы для работы с окном пользоваться кольцевым буфером, организованным на физически сплошном участке памяти, а не реальным сдвигом содержимого окна. Хотя для поддержания кольцевого буфера необходимы дополнительные затраты на сохранение целостности индексов в нем, в целом это дает очень существенный выигрыш потому, что отсутствует постоянное сдвигание большого блока памяти. Если размер кольцевого буфера равен степени двойки, то

стандартная для кольцевого буфера операция “смещение по модулю размер”, может быть заменена побитовой логической операцией, что еще больше повышает эффективность.

Помимо проблем с быстродействием, у алгоритма LZ77 возникают проблемы с самим сжатием. Они появляются, когда кодер не может найти совпадающую подстроку в словаре и выдает стандартный 3-компонентный код, пытаясь закодировать один символ. Если словарь имеет длину 4К, а буфер — 16 байтов, то код <0, 0, символ> будет занимать 3 байта. Кодирование одного байта в три имеет мало общего со сжатием и существенно понижает производительность алгоритма.

Упрощенный алгоритм LZ77

пока (буфер_предпросмотра не пуст) найти наиболее длинное соответствие (позиция_начала, длина) в буфере_предыстории и в буфере_предпросмотра; если (длина>минимальной_длины), то вывести в кодированные данные пару (позиция_начала, длина); иначе вывести в кодированные данные первый символ буфера_предпросмотра; изменить указатель на начало буфера_предпросмотра и продолжить.

Алгоритм LZ78

Этот алгоритм генерирует на основе входных данных словарь фрагментов, внося туда фрагменты данных (последовательности байт) по определенным правилам и присваивает фрагментам коды (номера). При сжатии данных (поступлении на вход программы очередной порции) программа на основе LZ78 пытается найти в словаре фрагмент максимальной длины, совпадающий с данными, заменяет найденную в словаре порцию данных кодом

фрагмента и дополняет словарь новым фрагментом. При заполнении всего словаря (размер словаря ограничен по определению) программа очищает словарь и начинает процесс заполнения словаря снова. Реализации этого метода различаются конструкцией словаря, алгоритмами его заполнения и очистки при переполнении.

Обычно, при инициализации словарь заполняется исходными (элементарными) фрагментами всеми возможными значениями байта от 0 до 255.

Это гарантирует, что при поступлении на вход очередной порции данных будет

найден в словаре хотя бы однобайтовый фрагмент.

Алгоритм LZ78 резервирует специальный код, назовем его «Reset», который вставляется в упакованные данные, отмечая момент сброса словаря. Значение кода Reset обычно принимают равным 256.

Таким образом при начале кодирования минимальная длина кода составляет 9 бит. Максимальная длина кода зависит от объема словаря – количества различных фрагментов, которые туда можно поместить. Если объем словаря измерять в байтах (символах), то очевидно, что максимальное количество фрагментов равно числу символов, а,

следовательно, максимальная длина кода равна log2 (объем словаря в байтах).

Проведение эксперимента и обработка результатов

Кодирование фразы алгоритмом LZ77 со следующими размерами СЛОВАРЬ/БУФЕР

Таблица 2.3. Варианты расчетных заданий

ФРАЗА	СЛОВАРЬ/БУФЕР			
«IDI TYDA, NE ZNAU KUDA»	8/5	12/5	12/8	16/8

Используем ввод фразы с клавиатуры в программе LZ77.EXE для разных размеров словаря/буфера:

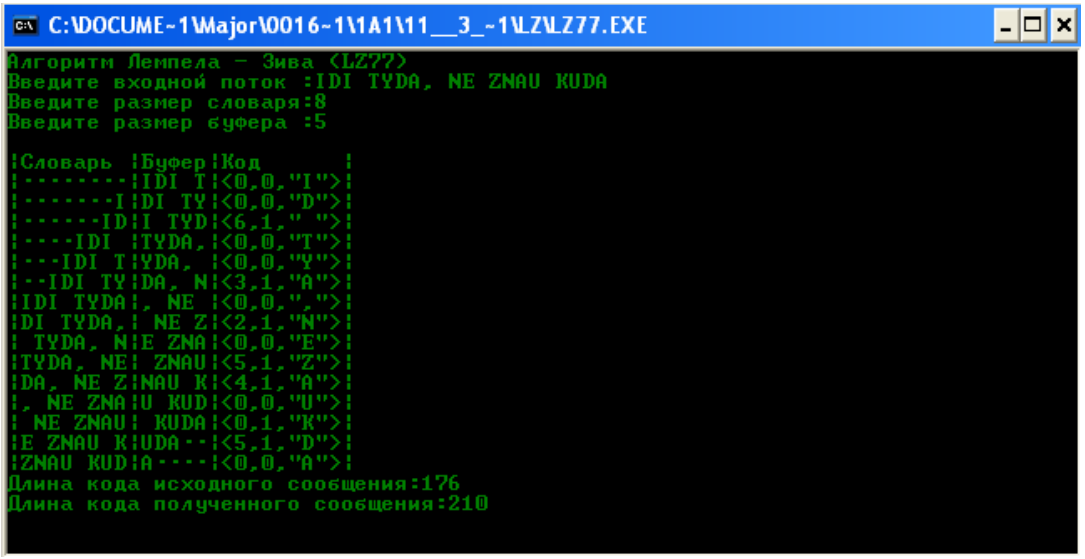


Рис. 2.9. Ввод фразы с клавиатуры для размеров 8/5

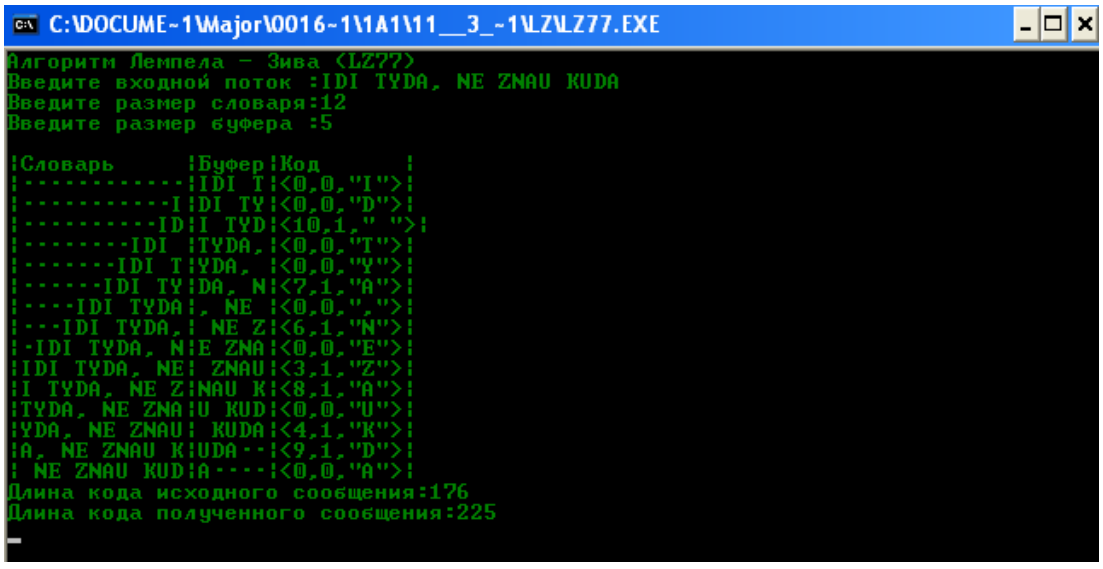


Рис. 2.10. Ввод фразы с клавиатуры для размеров 12/5

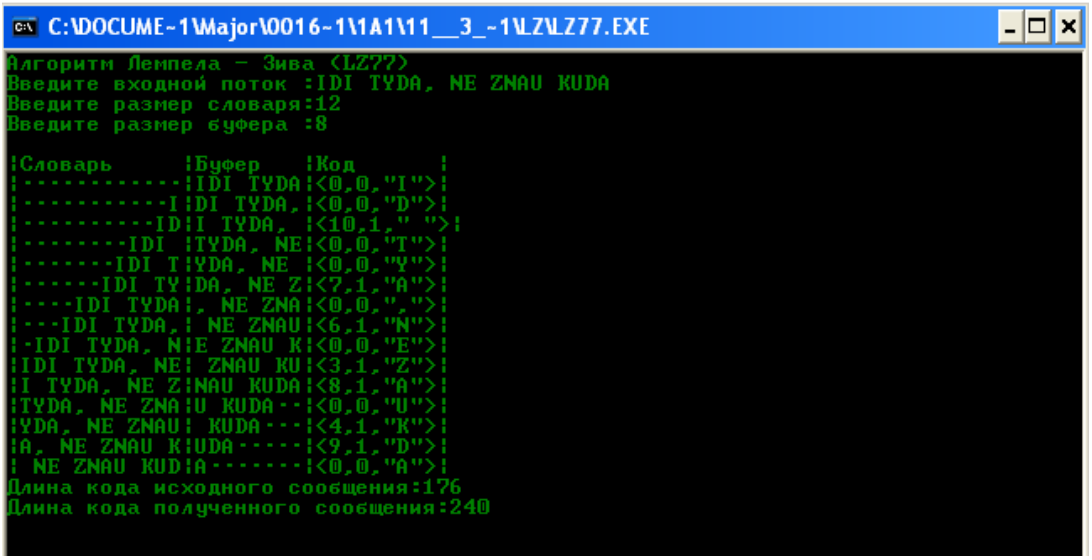


Рис. 2.11. Ввод фразы с клавиатуры для размеров 12/8

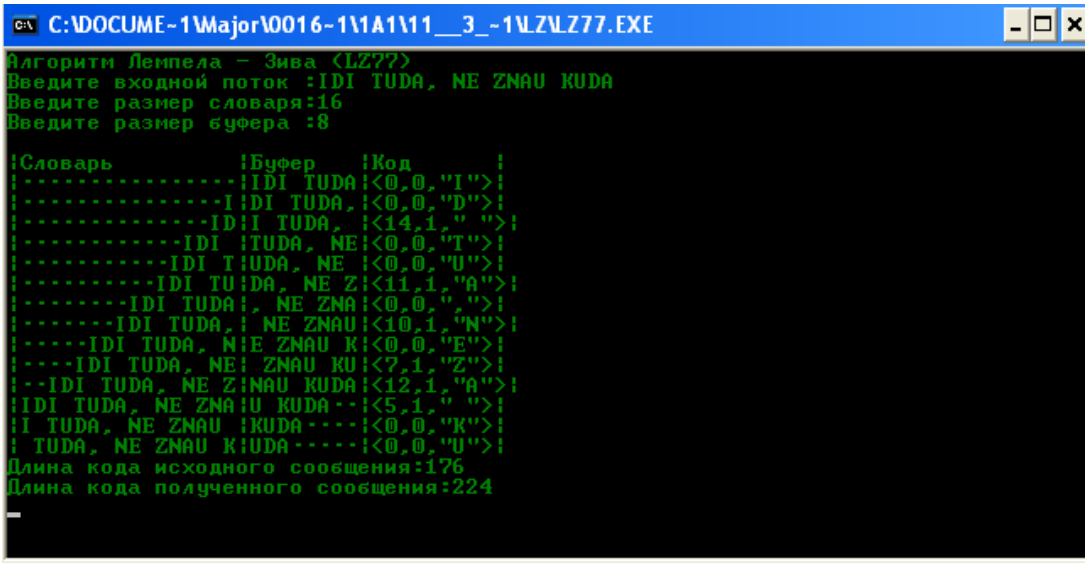


Рис. 2.12. Ввод фразы с клавиатуры для размеров 16/8

Используем ввод текста из файла для разных размеров словаря/буфера:

Текст: «Полуадаптированное моделирование решает эту проблему, используя для каждого текста свою модель, которая строится еще до самого сжатия на основании результатов предварительного просмотра текста (или его образца). Перед тем, как окончено формирование сжатого текста, модель должна быть передана раскодировщику. Несмотря на дополнительные затраты по передаче модели, эта стратегия в общем случае окупается благодаря лучшему соответствию модели тексту.

Адаптированное (или динамическое) моделирование уходит от связанных с этой передачей расходов. Первоначально и кодировщик, и раскодировщик присваивают себе некоторую пустую модель, как если бы символы все были равновероятными. Кодировщик

использует эту модель для сжатия очередного символа, а раскодировщик - для его разворачивания. Затем они оба изменяют свои модели одинаковым образом (например, наращивая вероятность рассматриваемого символа). Следующий символ кодируется и достается на основании новой модели, а затем снова изменяет модель. Кодирование продолжается аналогичным раскодированию образом, которое поддерживает идентичную модель за счет применения такого же алгоритма ее изменения, обеспеченным отсутствием ошибок во время кодирования. Используемая модель, которую к тому же не нужно передавать явно, будет хорошо соответствовать сжатому тексту.

Адаптированные модели представляют собой элегантную и эффективную технику, и обеспечивают сжатие по крайней мере не худшее производимого неадаптированными схемами. Оно может быть значительно лучше, чем у плохо соответствующих текстам статичных моделей. Адаптированные модели, в отличии от полуадаптированных, не производят их предварительного просмотра, являясь поэтому более привлекательными и лучшесжимающими. Т.о. алгоритмы моделей, описываемые в подразделах, при кодировании и декодировании будут выполнятся одинаково. Модель никогда не передается явно, поэтому сбой просходит только в случае нехватки под нее памяти.»

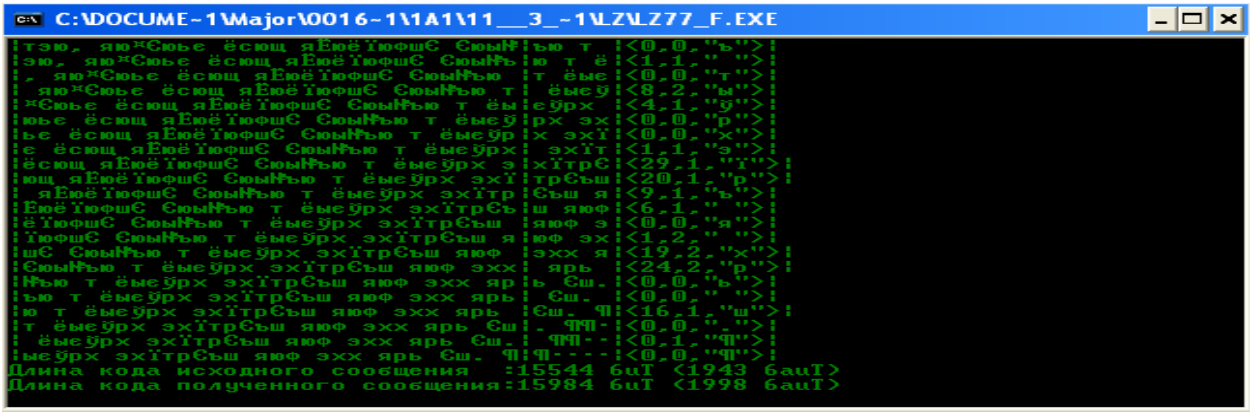


Рис. 2.13. Ввод текста из файла для размеров 32/5

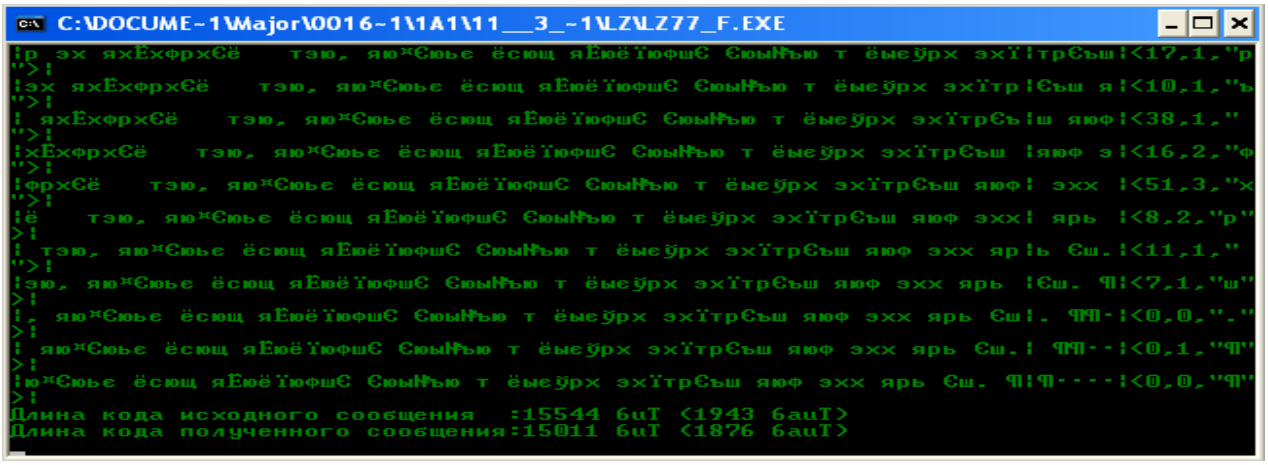


Рис. 2.14. Ввод текста из файла для размеров 64/5

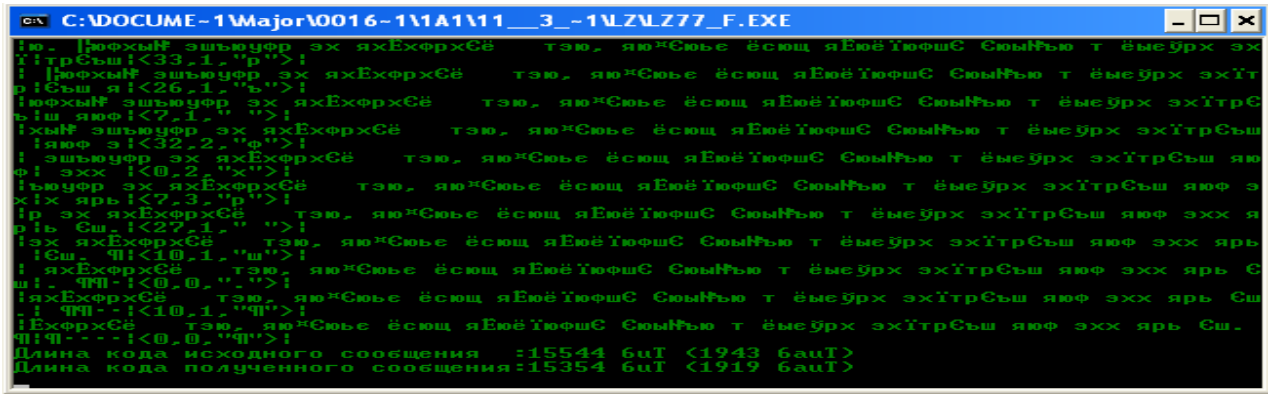


Рис. 2.15. Ввод текста из файла для размеров 80/5

Рис. 2.16. Ввод текста из файла для размеров 128/5

Таблица 2.4. Результаты измерений

$D_{словарь}$	32	64	80	128
$L_{вх}/L_{вых}$	0,972	1,036	1,012	1,141

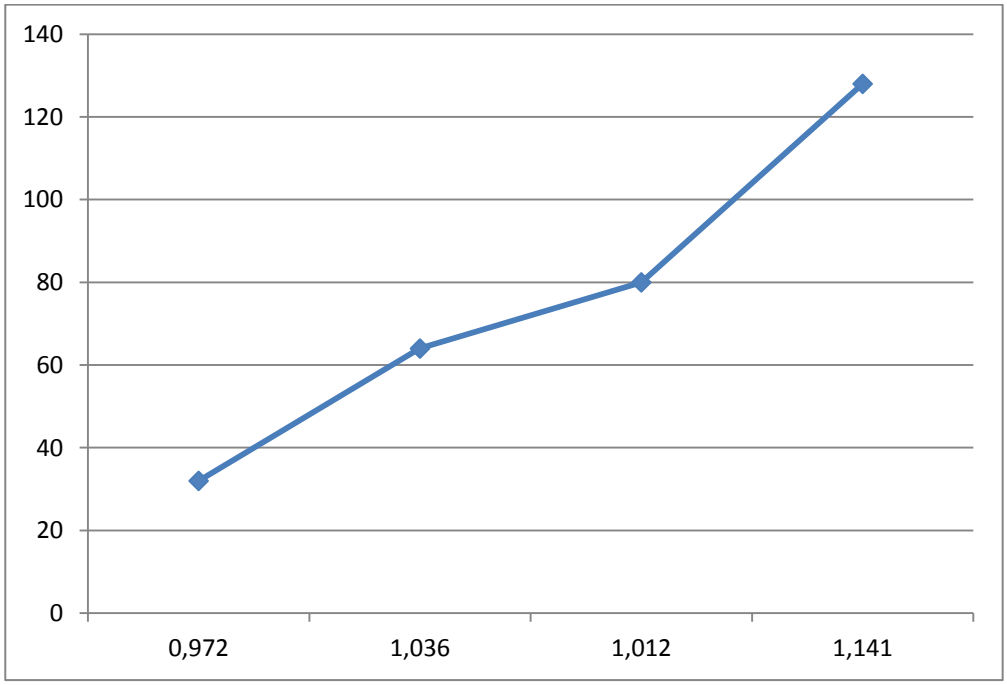


Рис. 2.17. График зависимости коэффициента сжатия от размера словаря

Кодирование для разной длины текста и одинакового размера словаря 64/5:

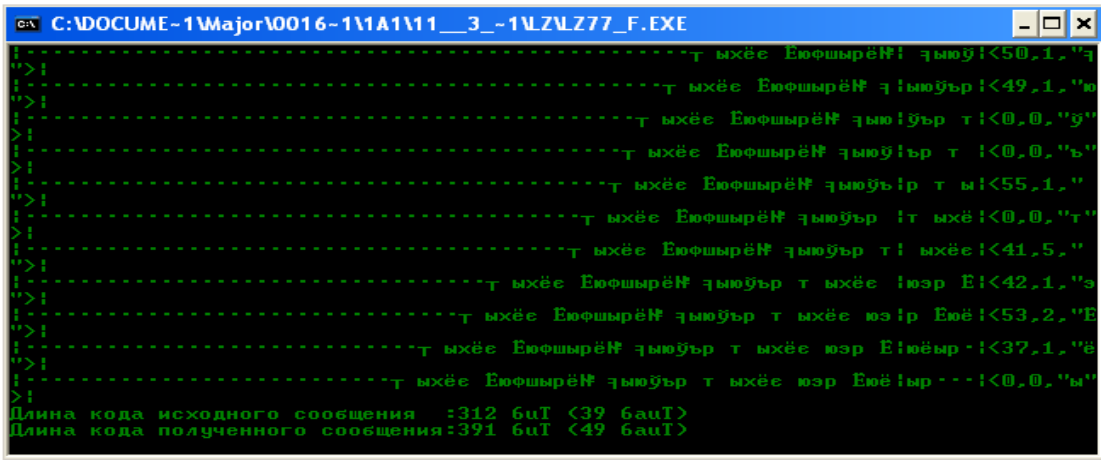


Рис. 2.18. Ввод текста длиной 39 байт

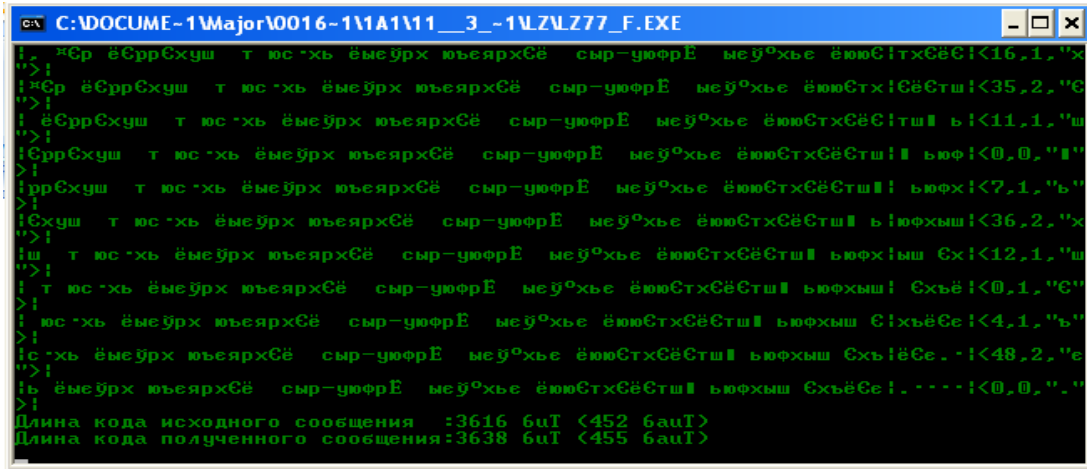


Рис. 2.19. Ввод текста длиной 452 байта

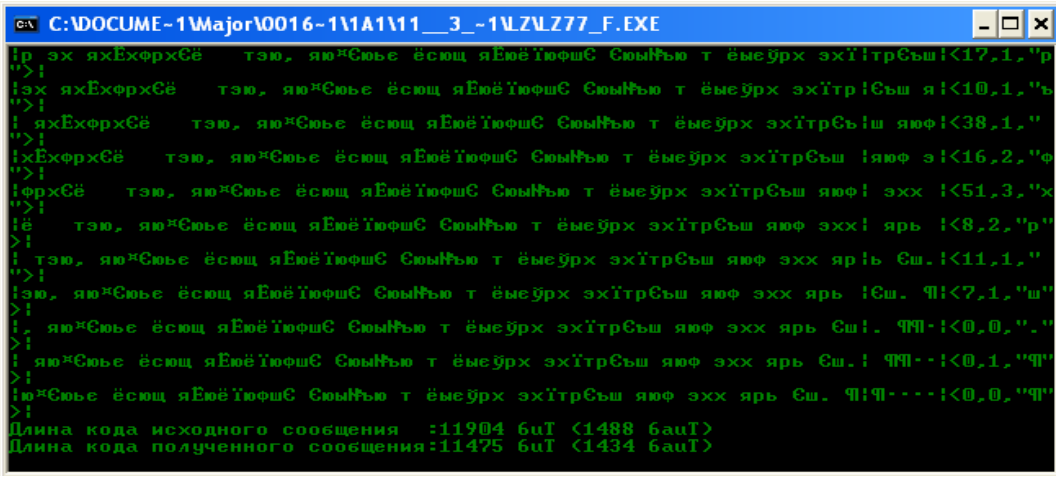


Рис. 2.20. Ввод текста длиной 1488 байт

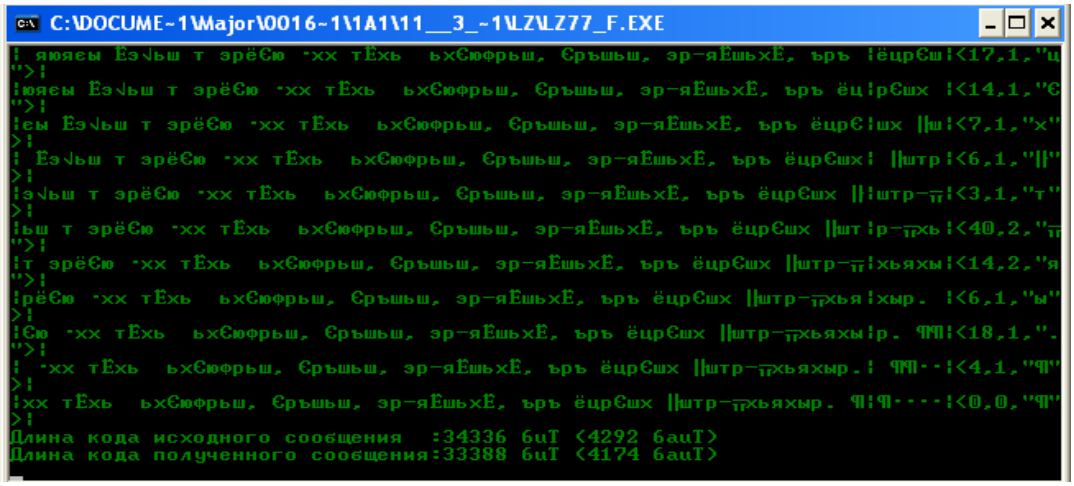


Рис. 2.21. Ввод текста длиной 4292 байта

Таблица 2.5. Результаты измерений

$L_{вх}$	39	452	1488	4292
$L_{вх}/L_{вых}$	0,795	0,993	1,037	1,028

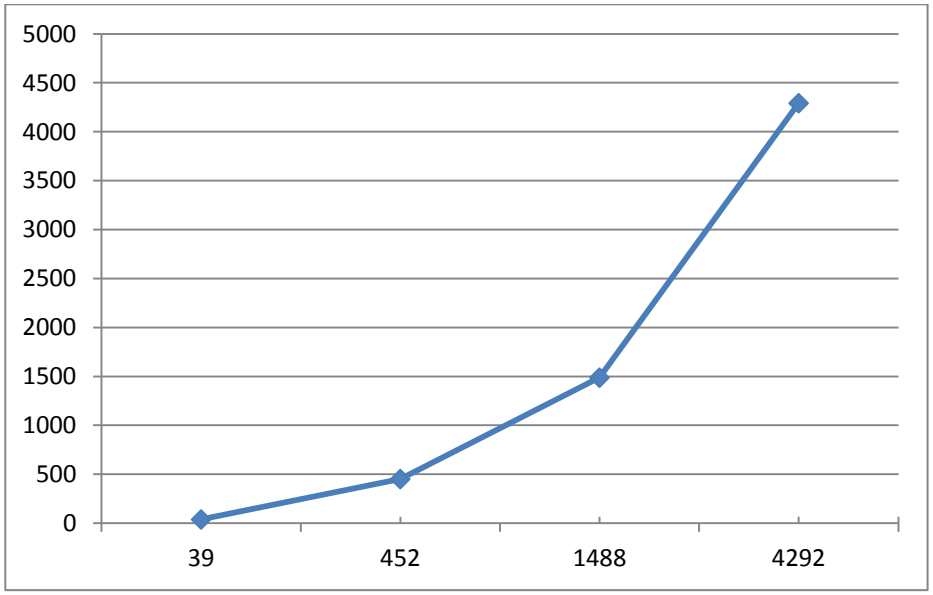


Рис. 2.22. График зависимости коэффициента сжатия от размера словаря

При кодировании коротких текстов коэффициент сжатия оказывается меньше 1, т.е. избыточность не удаляется, а вводится еще больше. При увеличении размера словаря коэффициент сжатия увеличивается. Большой размер буфера также ухудшает коэффициент сжатия. Если размер словаря кратен степеням двойки, то сжатие лучше. Однако, при

большом размере словаря снижается скорость кодирования. При увеличении размера текста коэффициент сжатия увеличивается.

2.4. Фрактальные методы кодирования изображений

Современные компьютеры весьма интенсивно применяют графику. Операционные системы с интерфейсом оконного типа используют картинки, например, для отображения директорий или папок. Некоторые совершаемые системой действия, например загрузку и пересылку файлов, также отображаются графически. Многие программы и приложения предлагают пользователю графический интерфейс (GUI), который значительно упрощает работу пользователя и позволяет легко интерпретировать полученные результаты. Компьютерная графика используется во многих областях повседневной деятельности при переводе сложных массивов данных в графическое представление.

Фрактал (лат. fractus — дроблёный, сломанный, разбитый) — геометрическая фигура, обладающая свойством самоподобия, то есть составленная из нескольких частей, каждая из которых подобна всей фигуре целиком. В математике под фракталами понимают множества точек в евклидовом пространстве, имеющие дробную метрическую размерность (в смысле Минковского или Хаусдорфа), либо метрическую размерность, отличную от топологической [14]. Слово «фрактал» может употребляться не только как математический термин. Фракталом в прессе и научно-популярной литературе могут называть фигуры, обладающие какими-либо из перечисленных ниже свойств: Обладает нетривиальной структурой на всех масштабах. В этом отличие от регулярных фигур (таких, как окружность, эллипс, график гладкой функции): если мы рассмотрим небольшой фрагмент регулярной фигуры в очень крупном масштабе, он будет похож на фрагмент прямой. Для фрактала увеличение масштаба не ведёт к упрощению структуры, на всех шкалах мы увидим одинаково сложную картину.

Является самоподобной или приближённо самоподобной. Обладает дробной метрической размерностью или метрической размерностью, превосходящей топологическую. Многие объекты в природе обладают фрактальными свойствами, например, побережья, облака, кроны деревьев, снежинки, кровеносная система и система альвеол человека или животных.

Фракталы, особенно на плоскости, популярны благодаря сочетанию красоты с простотой построения при помощи компьютера. Первые примеры самоподобных множеств с необычными свойствами появились в XIX веке (например, множество Кантора). Термин «фрактал» был введён Бенуа Мандельбротом в 1975 году и получил широкую популярность с выходом в 1977 году его книги «Фрактальная геометрия природы».

При фрактальном сжатии изображения для области меньшего размера подыскивается похожая на нее область большего размера того же изображения (рисунок 2.23).

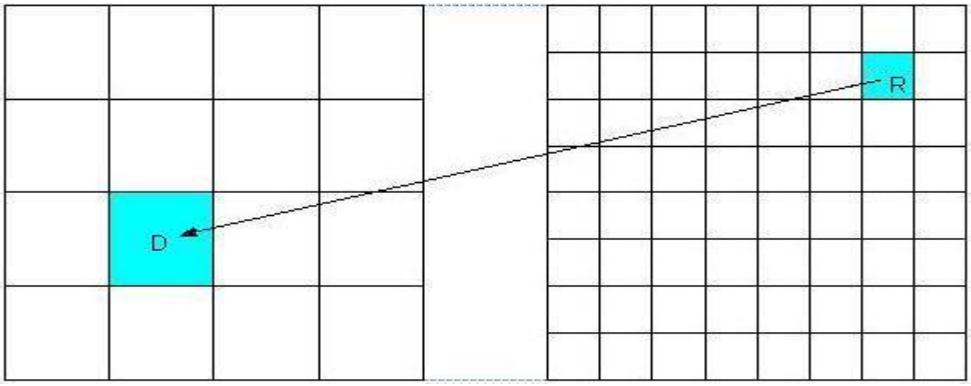


Рис. 2.23. Доменный и ранговый блоки

Область большого размера называется доменным блоком, а меньшего – ранговым блоком.Перевод доменного блока в ранговый осуществляется посредствомаффинных преобразований, которые в случае изображения в оттенках серого цвета можно представить следующей системой уравнений (2.9):

$$\begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & u \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ v \end{bmatrix}, \quad (2.9)$$

где x, y, z – соответственно координаты и яркость пикселя доменного блока;
x*, y*, z* – соответственно координаты и яркость пикселя рангового блока;
a, b, c, d, e, f – коэффициенты преобразований координат на плоскости;
u – коэффициент сжатия яркости;
v – сдвиг по яркости.

В рассматриваемых алгоритмах фрактального сжатия и восстановления изображения введенные упрощения позволяют заменить матричные преобразования операциями изменения ориентации доменного блока и операциями расчета яркости пикселей. Изменения ориентации доменного блока осуществляется за счет его поворота на угол кратный 90° и зеркального отражения и поворота зеркального отражения.

Из всех возможных доменных блоков выбирается блок, ближайший (в выбранной метрике) к рассматриваемому ранговому блоку.
Когда доменный блок найден, то запоминаются его номер и параметры преобразования в текущий ранговый блок, из которых и состоит решение задачи фрактального сжатия.

Ниже, на рисунке 2.32. ,представлен алгоритм фрактального сжатия.

Рис. 2.24. Алгоритм фрактального сжатия

Программное обеспечение в DELPHI 7

Приложение выполняет фрактальное сжатие / распаковку изображений с помощью классического алгоритма.

В нее можно загружать любые изображения, но размер должен быть не более 512x512 пикселей. Программа будет автоматически убирать цвет изображений.

Эти ограничения введены для того, чтобы существенно сократить время сжатия изображений на базе фрактального алгоритма.

Основные характеристики:

- Сжатие и декодирование изображений формата .bmp;
- Просмотр полученного результата;
- Просмотр размер полученного изображения.

На рисунке 2.35. представлено главное и единственное меню программы.

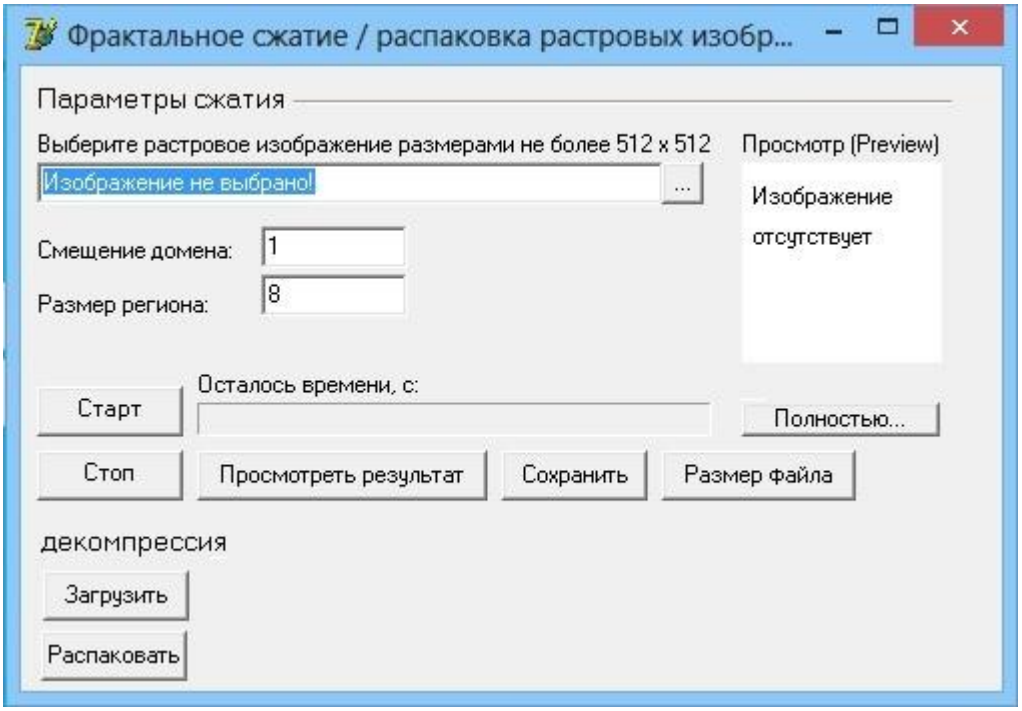


Рис. 2.25. Главное меню программы.

В программе доступны для изменения 2 следующих пункта: смещение домена и размер региона.

Смещение домена: Определяет шаг поиска участка в доменном изображении. Минимальный шаг равен 1. Чем больше шаг, тем быстрее выполняется поиск, но при этом могут быть пропущены важные детали изображения.

Размер региона: определяет размер области, на которое разбивается исходное изображение. При компрессии для каждой области осуществляется поиск подходящего домена с учетом трансформации (аффинных преобразований). Чем больше размер региона, тем хуже качество и при этом уменьшается размер IFS-изображения.

Методика работы в программе и проведение исследования основных технических характеристик системы

Для обеспечения фрактального сжатия была выбрана программа «Фрактальное сжатие / распаковка растровых изображений».

Для того чтобы начать работать с программой необходимо:

1. Загрузить изображение, которое удовлетворяет требование программы;
2. Выставить значение параметров «Смещение домена» и «Размер региона»;
3. Нажать кнопку старт;

4. Дождаться окончания сжатия изображения;

5. Просмотреть полученный результат и, если он удовлетворил нас, сохранить его.

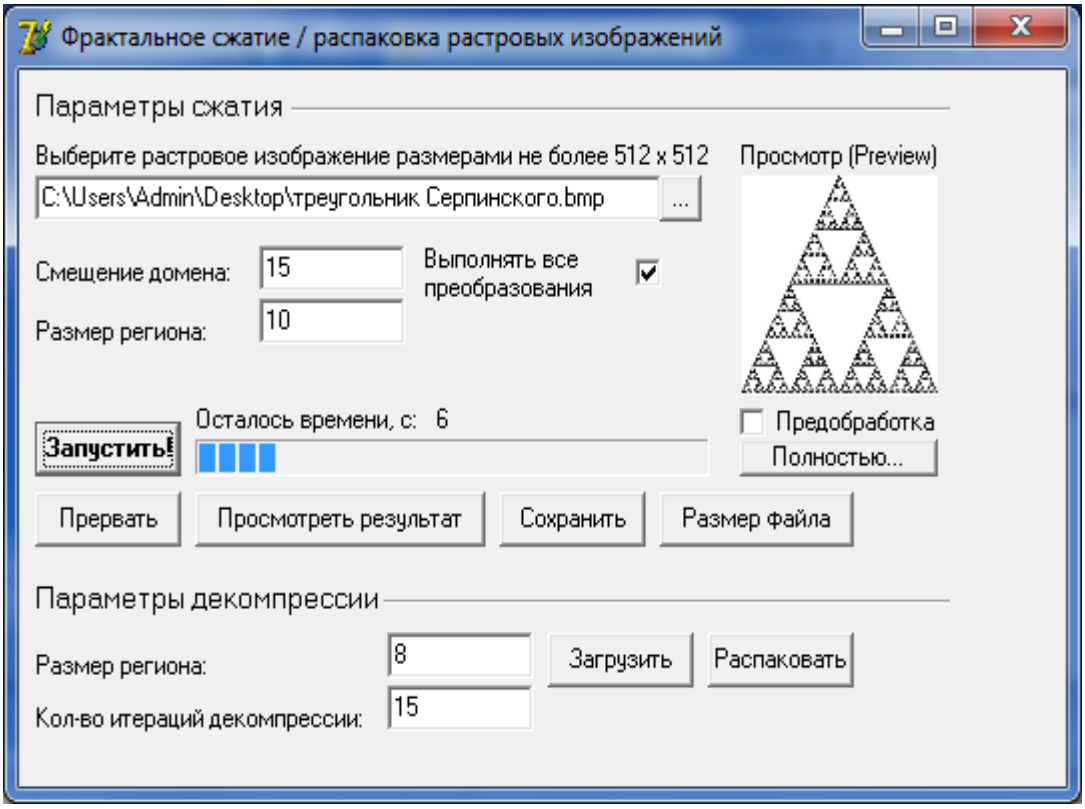


Рис. 2.26. Сжатие изображения в программе «Фрактальное сжатие / распаковка растровых изображений»

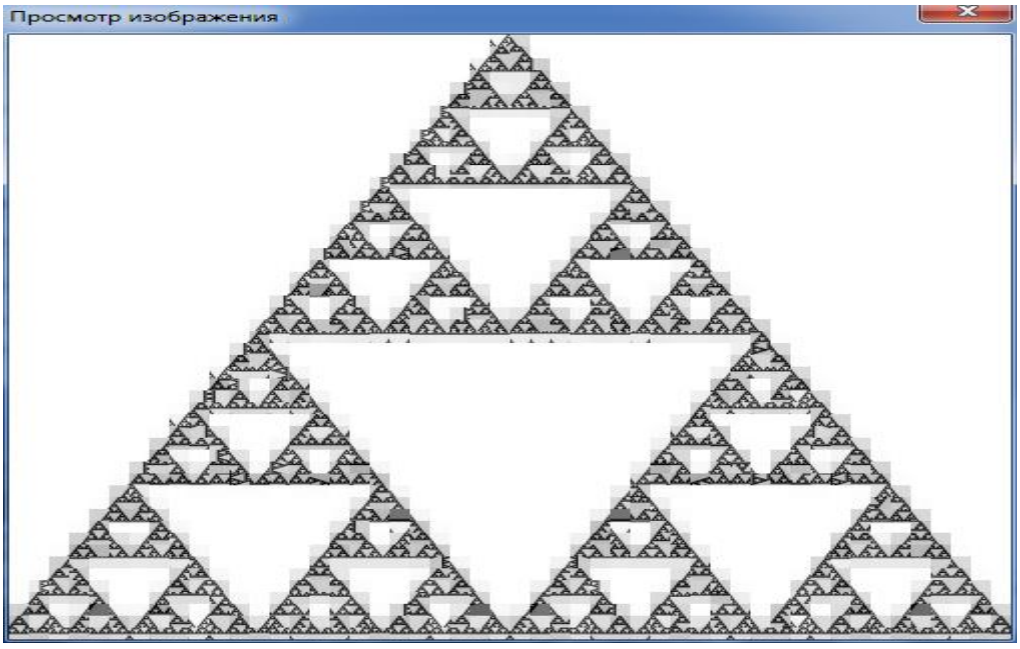


Рис. 2.27. Результат сжатия.

После сжатия изображение будет иметь формат .IFS.

Стандартные кодеки Windows не способны декодировать полученное изображение, поэтому в программе предусмотрена функция просмотра изображений формата .IFS.

Чтобы просмотреть нужное нам изображение необходимо:

1. В меню «декомпрессия» нажать на кнопку «Загрузить»;
2. Выбрать изображение формата .IFS;
3. В меня «декомпрессия» нажать на кнопку «Распаковать»

Далее рассмотрим сжатие изображений со спутника X-SAR Европейского космического агентства. На рисунке 2.36 представлено первоначальное изображение со спутника размером 435 Кб и разрешением 473x314 пикселей.

Рис. 2.28. Первоначальное изображение.

Далее, на рисунке 2.29, представлено это же изображение после обработки, при параметре смещение домена=1 и размер региона=8, время потраченное на сжатие t=703 с, размер файла =11,4Кб.

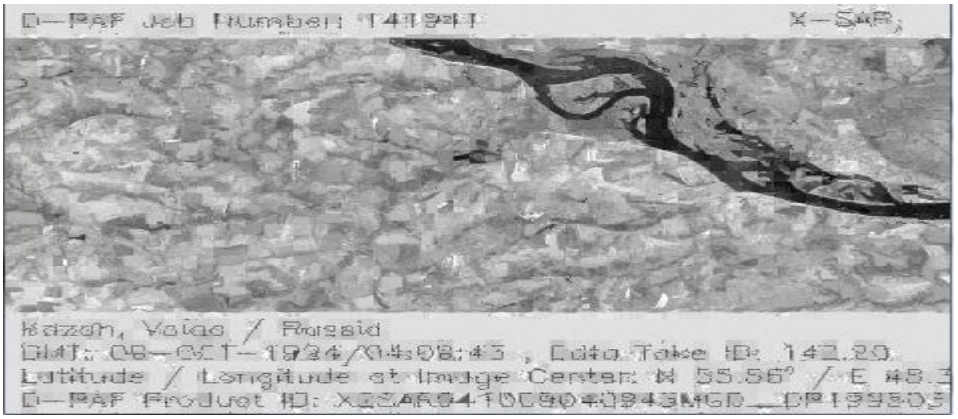


Рис. 2.29. Обработанное изображение.

Далее, на рисунке 2.30, представлено это же изображение после обработки, при параметре смещение домена=10 и размер региона=8, время потраченное на сжатие t=6 с, размер файла =11Кб.

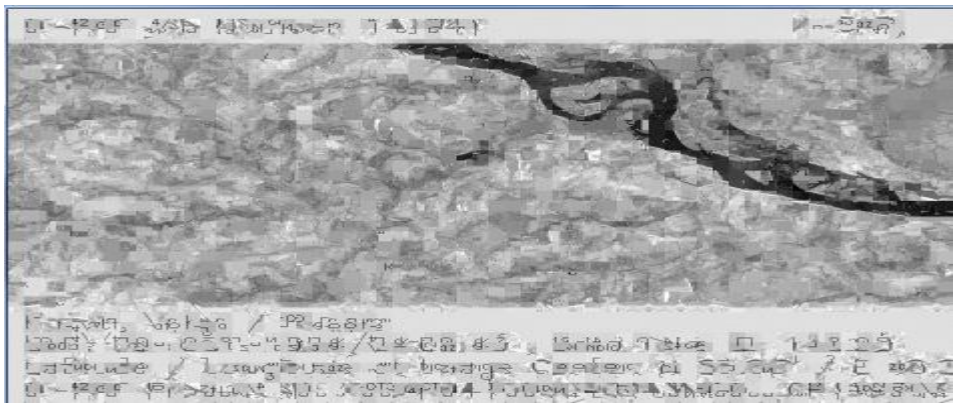


Рис. 2.30. Обработанное изображение.

Далее, на рисунке 2.31, представлено это же изображение после обработки, при параметре смещение домена=5 и размер региона=10, время потраченное на сжатие t=27 с, размер файла =7,12Кб.

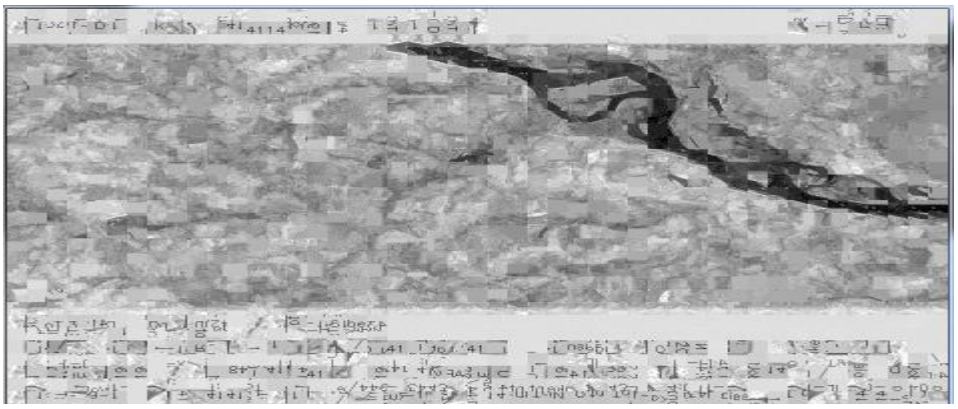


Рис. 2.31. Обработанное изображение.

Далее, на рисунке 2.32, представлено это же изображение после обработки, при параметре смещение домена=10 и размер региона=12, время потраченное на сжатие t=5 с, размер файла =4,95Кб.

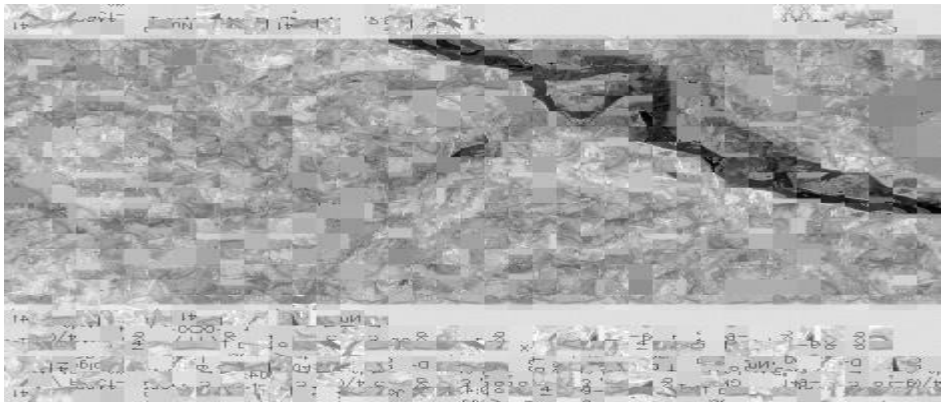


Рис. 2.32. Обработанное изображение.

Далее, на рисунке 2.33, представлено это же изображение после обработки, при параметре смещение домена=1 и размер региона=15, время потраченное на сжатие t=566 с, размер файла =3,04Кб.

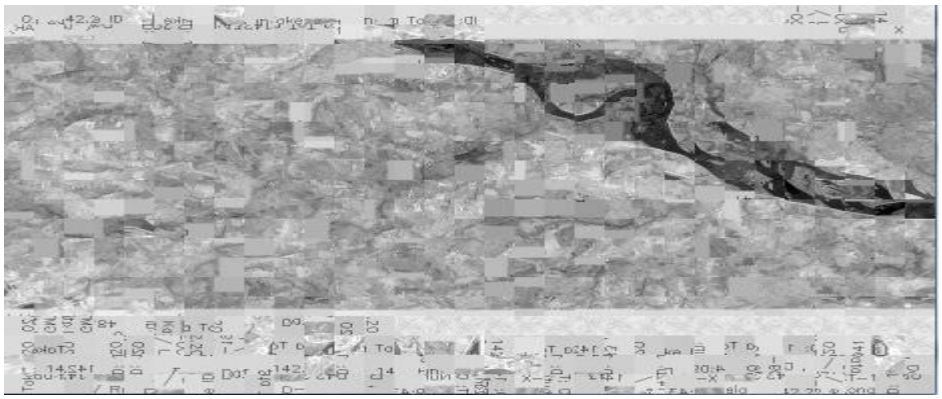


Рис. 2.33. Обработанное изображение.

Далее, на рисунке 2.34, представлено это же изображение после обработки, при параметре смещение домена=10 и размер региона=14, время потраченное на сжатие t=4 с, размер файла =3,55Кб.

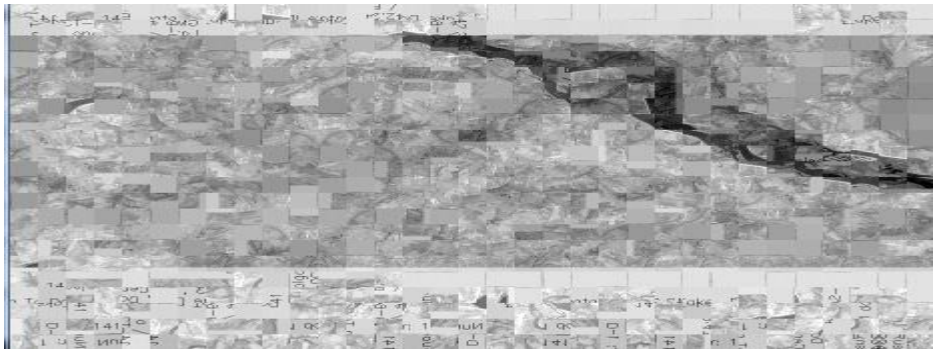


Рис. 2.34. Обработанное изображение.

Далее, на рисунке 2.35, представлено это же изображение после обработки, при параметре смещение домена=20 и размер региона=10, время потраченное на сжатие t=1 с, размер файла =4,12Кб.

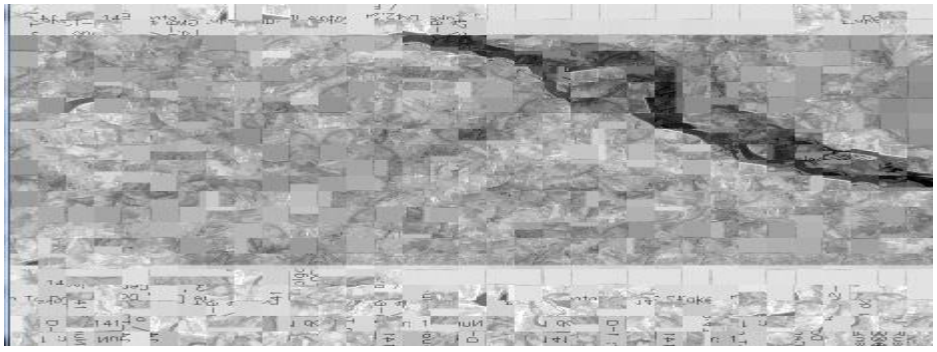


Рис. 2.35. Обработанное изображение.

Ниже, на рисунке 2.36 представлен график зависимости времени сжатия от размера изображения.

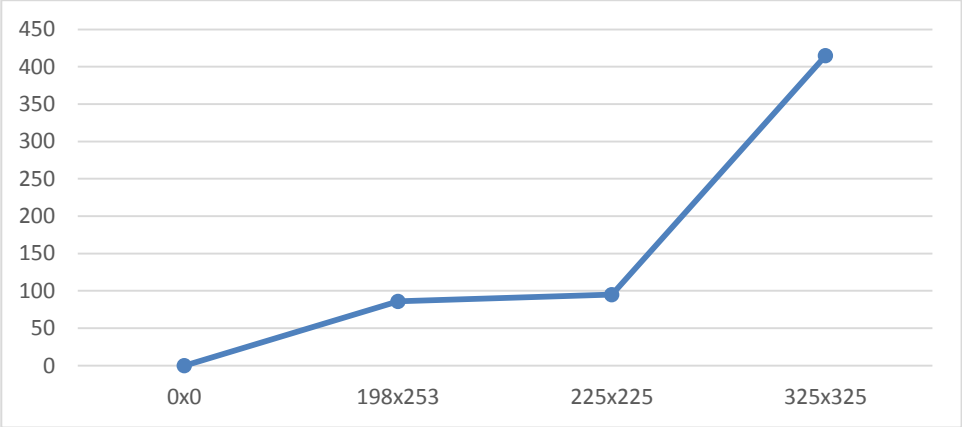


Рис. 2.36. График зависимости времени сжатия от размера изображения.

На рисунке 2.37 представлен график зависимости времени сжатия от параметра «смещение домена»

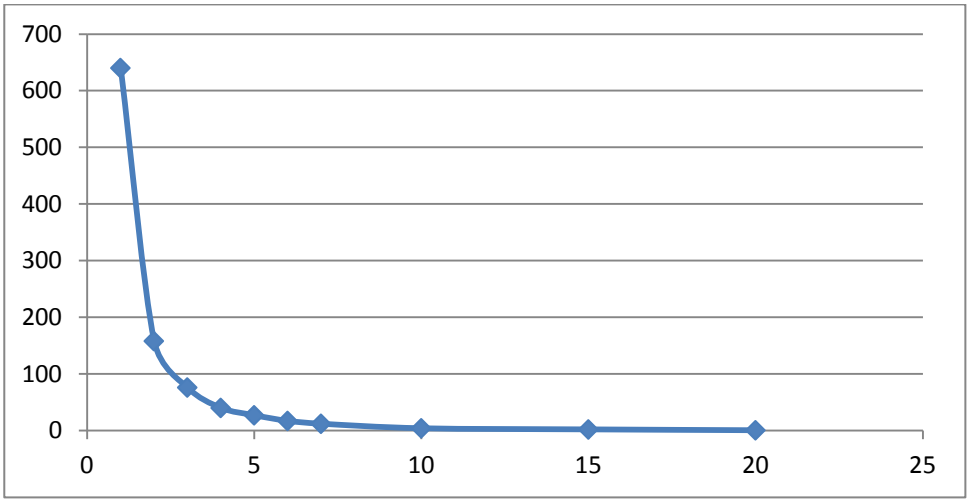


Рис. 2.37. График зависимости времени сжатия от параметра «смещение домена».

На рисунке 2.38 представлен график зависимости времени сжатия от параметра «размер региона».

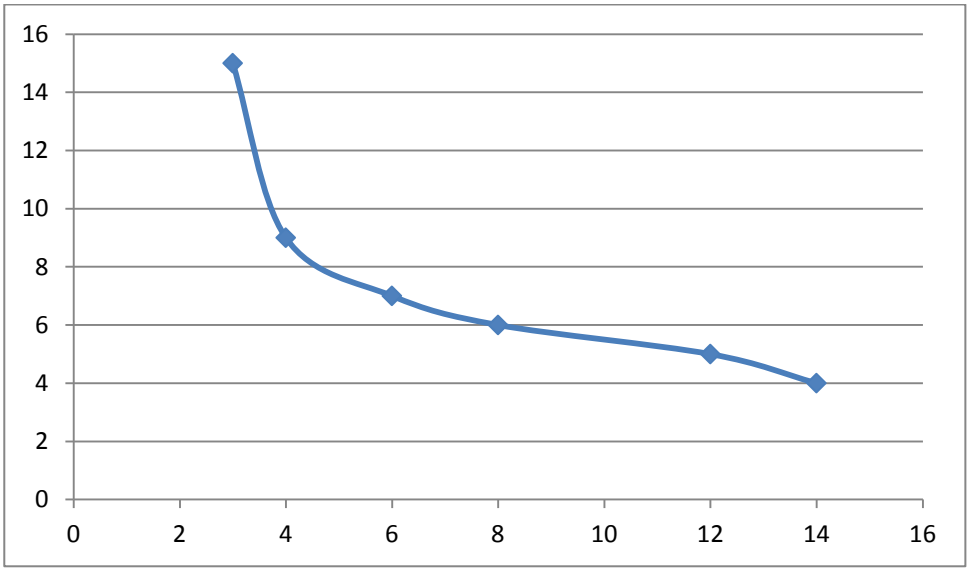


Рис. 2.38. График зависимости времени сжатия от параметра «размер региона»

На рисунке 2.39 представлен график зависимости размера изображения (в КБайт) от параметра «размер региона»

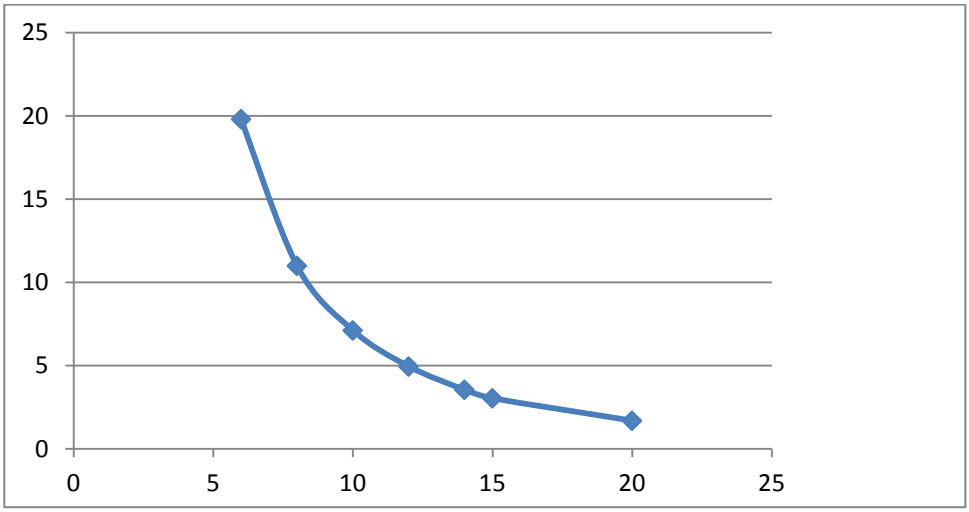


Рис. 2.39. График зависимости размера изображения (в КБайт) от параметра «размер региона»

Ниже, в таблице 2.6 представлена зависимость размера от коэффициента сжатия.

Таблица 2.6. Зависимость размера изображения от коэффициента сжатия.

Коэ-т сжатия	143	122	105	88	61	40
Размер изображения, Кб	3,04	3,55	4,12	4,95	7,12	11

Как видно из таблицы, при максимально достигнутом коэффициенте сжатия равному 143, размер изображения уменьшился с 435Кб до 3,04Кб.

Из представленных выше графиков можно сделать вывод, что:

1. Чем больше изображение, тем больше время сжатия;
2. Чем больше параметр «смещение домена», тем меньше время сжатия;
3. Чем меньше параметр «размер региона», тем больше время сжатия;
4. Чем меньше параметр «размер региона», тем больше размер изображения.

Рассмотрим таблицу 2.4 , в которой сводятся воедино параметры различных алгоритмов сжатия изображений [13].

Таблица 2.7. Алгоритмы сжатия

Алгоритм	**К-ты сжатия**	**На что ориентирован**	**Потери**
RLE	32, 2, 0.5	3,4-х битные	Нет
LZW	1000, 4, 5/7	1-8 битные	Нет
Хаффмана	8, 1.5, 1	8 битные	Нет
CCITT-3	213(3), 5, 0.25	1-битные	Нет
JBIG	2-30 раз	1-битные	Нет
Lossless JPEG	2 раза	24-битные, серые	Нет
JPEG	2-20 раз	24-битные, серые	Да
Рекурсивное сжатие	2-200 раз	24-битные, серые	Да
Фрактальный	2-2000 раз	24-битные, серые	Да

Использование сжатия с потерями предоставляет возможность за счет потерь регулировать качество изображений. Коэффициенты сжатия у фрактальных алгоритмов варьируются в пределах 2-2000 раз. Причем большие коэффициенты достигаются на реальных изображениях, что нетипично для предшествующих алгоритмов. Ниже представлен график зависимости размера изображения от коэффициента сжатия

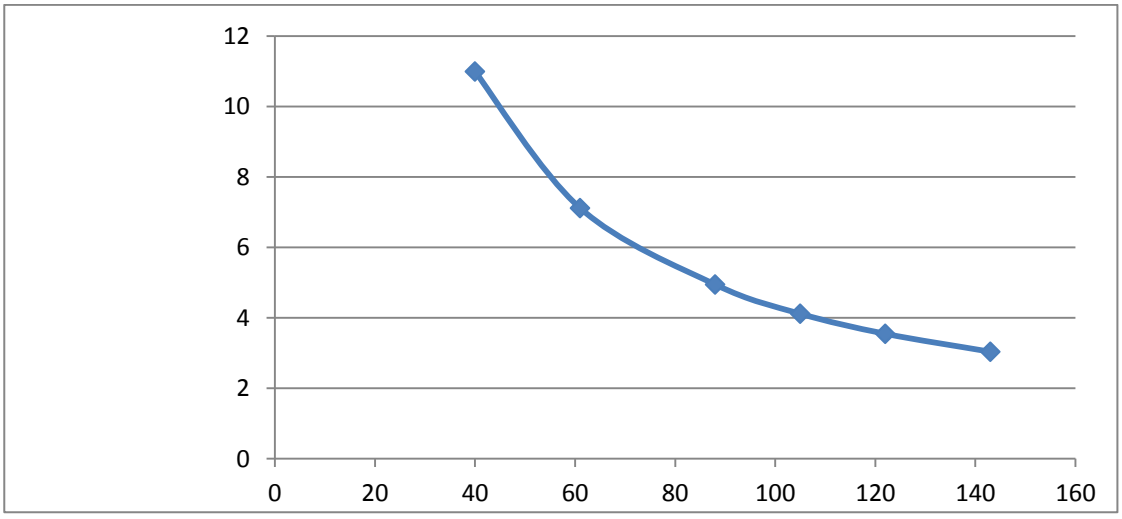


Рис. 2.40. Зависимость размера изображения по оси ординат, от коэффициента сжатия по оси абсцисс

Ниже, для наглядности, приведены 2 рисунка, первый – исходное изображение; второй – изображение с максимально достигнутом коэффициентом сжатия равным 143.

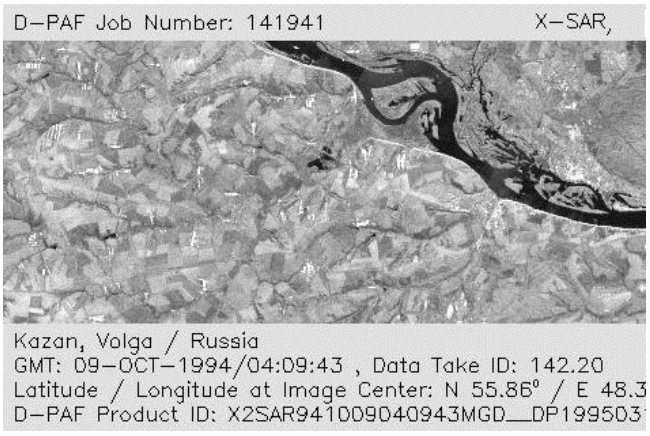


Рис. 2.41. Исходное изображение

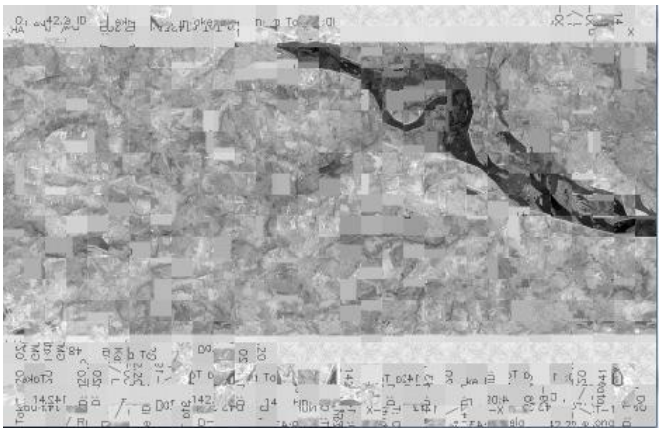


Рис. 2.42. Изображение с коэффициентом сжатия 143

Как видно из рисунков, при сжатии текстовой информации она становится нечитабельная, хотя как всё изображение в целом остаётся узнаваемым.

Недостатком этого алгоритма является потребность в больших вычислительных мощностях при архивации. Фактически это первый существенно несимметричный алгоритм. Причем, если у всех предшествующих алгоритмов коэффициент симметричности (отношение времени архивации ко времени разархивации) не превышает 3, то у фрактального алгоритма он колеблется от 1000 до 10000.

2.5. Вейвлет преобразования сигналов и изображений [15]

В настоящее время вейвлет-анализ является одним из наиболее мощных и при этом гибких средств исследования данных: помимо возможностей сжатия и фильтрации данных, анализ в базисе вейвлет-функций позволяет решать задачи идентификации, моделирования, аппроксимации стационарных и нестационарных процессов, исследовать вопросы наличия разрывов в производных, осуществлять поиск точек склеивания данных, удалять в данных тренд, отыскивать признаки фрактальности информации. Стоит отметить, что в основе подобных возможностей, обеспечивающих вейвлет-анализу весьма перспективное будущее, лежит природа его многомасштабности. Иначе говоря, гармонический анализ не способен конкурировать с вейвлет-анализом.

Вейвлет-преобразование широко используется для анализа сигналов. Помимо этого, оно находит большое применение в области сжатия данных. В дискретном вейвлет-преобразовании наиболее значимая информация в сигнале содержится при высоких амплитудах, а менее полезная — при низких. Сжатие данных может быть получено за счет отбрасывания низких амплитуд. Вейвлет-преобразование позволяет получить высокое

соотношение сжатия в сочетании с хорошим качеством восстановленного сигнала. Вейвлет-преобразование было выбрано для стандартов сжатия изображений JPEG2000 и ICER. Однако, при малых сжатиях вейвлет-преобразование уступает по качеству в сравнении с оконным Фурье-преобразованием, которое лежит в основе стандарта JPEG.

Выбор конкретного вида и типа вейвлетов во многом зависит от анализируемых сигналов и задач анализа. Для получения оптимальных алгоритмов преобразования разработаны определенные критерии, но их еще нельзя считать окончательными, так как они являются внутренними по отношению к самим алгоритмам преобразования и, как правило, не учитывают внешних критериев, связанных с сигналами и целями их преобразований. Отсюда следует, что при практическом использовании вейвлетов необходимо уделять достаточное внимание проверке их работоспособности и эффективности для поставленных целей по сравнению с известными методами обработки и анализа.

Изобретение вейвлетов было напрямую связано с необходимостью более глубокого анализа сигналов, чем анализ сигнала с помощью преобразования Фурье. Вейвлеты используют в тех случаях, когда результат анализа некоторого сигнала должен содержать не только простое перечисление его характерных частот (масштабов), но и сведения об определенных локальных координатах, при которых эти частоты проявляют себя. Анализ и обработка нестационарных (во времени) или неоднородных (в пространстве) сигналов разных типов представляют собой основное поле применения вейвлет-анализа.

Вейвлеты – новые системы базисных функций, используемых для представления, фильтрации, сжатия, хранения и т.д. любого из «сигналов»

$$f : R^n \to C. \tag{2.10}$$

Если n =1, переменная t представляет собой время, и мы работаем с временными сигналами. $f : R \to C$. Случай n=2 относится к обработке изображений. То есть, вейвлет представляет собой функцию от времени, которая используется для анализа одномерного или двумерного сигнала.

В своих работах Малл определяет вейвлет как функцию ϕ с нулевым средним значением:

$$\int_{-\infty}^{\infty} \phi(t)dt = 0. \tag{2.11}$$

Вейвлетом называется волновое колебание с начальным значением амплитуды, равным нулю, затем увеличивающимся значением и затем снова уменьшающейся до нулевого значения. Это выражается в небольшом колебании исследуемого сигнала. Такое поведение этих функций позволяет записать и исследовать сейсмические волны или колебания сердца.

В общем случае, вейвлеты представляют собой функции, имеющие специфические свойства, позволяющие эффективно обрабатывать сигналы. Вейвлеты могут комбинироваться (с применением операций сдвига, умножения и суммирования), с выборками изучаемого сигнала для получения соответствующей информации.

Одна из основополагающих идей использования вейвлетов для представления сигналов заключается в разбивке приближения к сигналу на две составляющие: грубую (аппроксимирующую) и утонченную (детализирующую), с последующим их уточнением итерационным методом. Каждый шаг такого уточнения соответствует определенному уровню декомпозиции и восстановления сигнала. Это возможно как во временной, так и в частотной областях представления сигнала вейвлетами. Вейвлеты применяются во многих областях науки; главные области применения – обработка и анализ сигналов и сжатие изображений.

Для обработки сигнала с помощью вейвлетов сначала выбирается анализирующий (материнский) вейвлет. Обозначим его $x \mapsto \phi(x)$. Как правило, они определены на компактном носителе (на промежутке [0,L]). Под носителем функции понимается ее область определения.

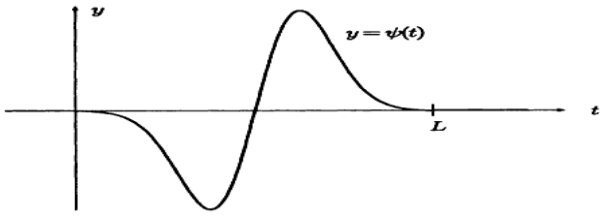


Рис. 2.43. Материнский вейвлет

Растянутые и сдвинутые копии вейвлета ϕ называются вейвлетными функциями. Для них принято следующее обозначение:

$$\varphi_{a,b} : R \to C, t \mapsto \frac{1}{|a|^{1/2}} \psi\left(\frac{t-b}{a}\right). \qquad (2.12)$$

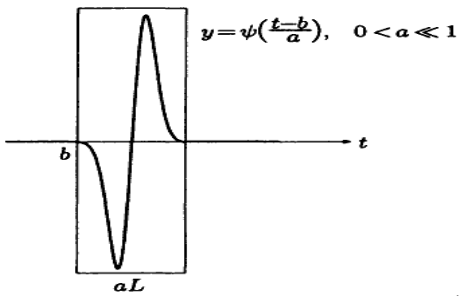


Рис. 2.44. Общий вид совокупности волновых функций (вейвлет-пакетов)

Параметр *a* называется масштабирующим параметром, *a* параметр *b* – параметром сдвига.

Результат выполнения вейвлет-преобразования (массив данных, получающийся в результате) полностью зависит от материнского вейвлета.

На практике используются следующие типы вейвлетов:

- Вейвлет Хаара.
- Мексиканская шляпа.
- Модулированная гауссова кривая.
- Производная гауссовой кривой.
- Вейвлеты Добеши, Грассмана, Мейера.
- Вейвлеты Бэттла-Лемарье.
- Симлеты.
- Койфлеты.

Математический аппарат вейвлета Хаара

Вейвлет Хаара представляет собой следующую функцию:

$$\psi(\mathrm{x}) = \begin{cases} 1, (0 \le \mathrm{x} \prec \frac{1}{2}) \\ -1, (\frac{1}{2} \le \mathrm{x} \prec 1) \end{cases} \quad (2.13)$$

Используя в качестве материнской функции вейвлет Хаара, можно определить вейвлетные функции.

$$\psi_{r,k} = 2^{-r/2} \psi_{haar} \left(\frac{t - k \cdot 2^r}{2^r} \right), r, k \in Z \quad (2.14)$$

В качестве носителя данная функция имеет интервал длины 2^r:

$$I_{r,k} = \lfloor k \cdot 2^r, (k+1) \cdot 2^r \rfloor \quad (2.15)$$

Большим значениям r соответствуют большие интервалы L, следовательно, вейвлетные функции имитируют большие волны. Одним из примеров, иллюстрирующих применение вейвлетов, является разложение сигнала с помощью выбранного преобразования [5].

Масштабирующая функция имеет вид:

$$\phi(x) = \begin{cases} 1, 0 \le x < 1 \\ 0, x \notin [0,1) \end{cases} \quad (2.16)$$

Ее интеграл: $\int_{-\infty}^{\infty} \phi(x)dx = 1$.

Масштабирующая функция определяет аппроксимацию сигнала (позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов).

Построение масштабирующей функции и функции вейвлета Хаара представлено на рисунок 2.45.

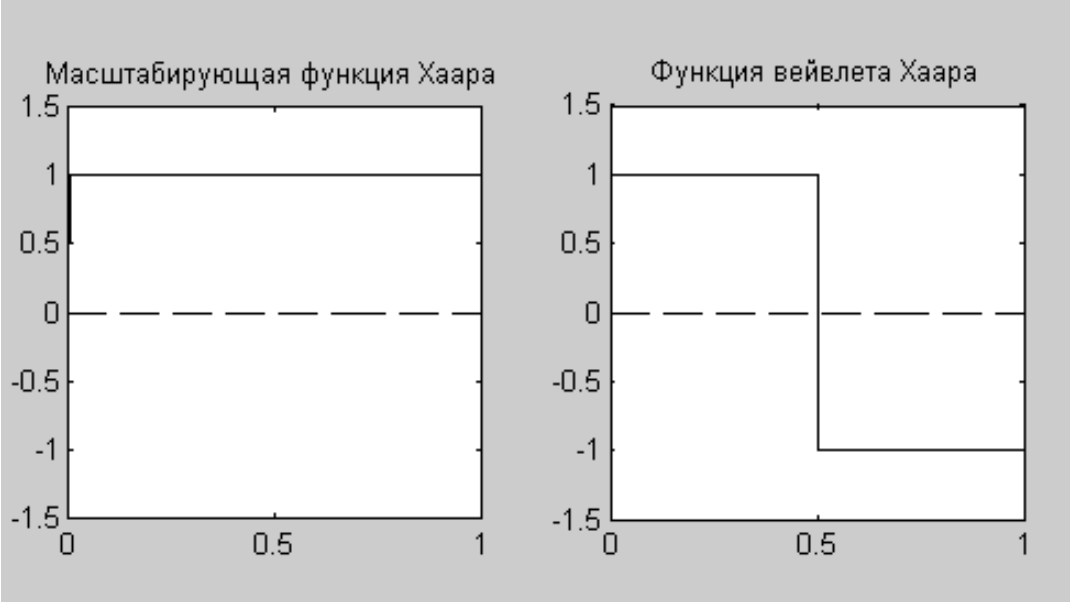


Рис. 2.45. Масштабирующая функция и функция вейвлета Хаара

Преобразование Хаара использует комбинацию этих функций. Чтобы понять, как выполняется преобразование, выполним сначала преобразование одномерного массива.

Пример. Выполним преобразование Хаара следующего массива:

A = (6, 7, 8, 9, 10, 6, 7, 8).

Вычислим сначала средние и полуразности элементов:

1 шаг:

Средние: (6+7)/2 = 13/2 = 6.5; (8+9)/2 = 8.5; (10+6)/2 = 8, (7+8)/2 = 7.5.

Полуразности: (6-7)/2 = -0.5; (8-9)/2 = -0.5; (10-6)/2 = 2; (7-8)/2 = -0.5

Таким образом, получим массив из восьми элементов:

A_1 = (6.5, 8.5, 8, 7.5, -0.5, -0.5, 2, 0.5)

Обратим внимание на последние четыре элемента – их значения гораздо меньше, чем до выполнения преобразования. Поэтому их оставим без изменений. Средние значения называются аппроксимирующими коэффициентами, в которых накапливается максимальная информация, а значения полуразностей позволяют осуществлять полное восстановление

исходного сигнала. На следующем шаге мы будем работать с первыми четырьмя элементами, а вторую половину массива оставим без изменений.

2 шаг:

Средние: (6.5+8.5)/2 = 7.5; (8+7.5)/2 = 7,75.

Полуразности: (6.5-8.5)/2 = -1; (8-7.5)/2 = 0.25.

Новый массив: A_2 = (7.5, 7.75, -1, 0.25, -0.5, -0.5, 2, -0.5)

3 шаг:

Средние: (7.5+7.75)/2 = 7.625; (-1+0.25) = 0.625.

Полуразности: (7.5-7.75)/2 = -0.125; (-1-0.25)/2 = -0.625.

Новый массив A_3 = (7.625, 0.625,-0.125, -0.625, -0.5, -0.5, 2, -0.5)

Из за постоянного вычисления полуразностей происходит постоянное уменьшение значений исходных элементов (пикселей изображения), поэтому после выполнения преобразования, можно к получившемуся массиву дополнительно применить один из алгоритмов сжатия (например, Хаффман или арифметическое кодирование). Если необходимо использовать сжатие с потерями, то здесь используется операция квантования или удаления наименьших значений полуразностей: например, может быть записано: A_3 = (7.625, 0.6250, 0, 0, 0, 2, 0). Тогда исходные данные восстановятся с искажениями.

Рис. 2.46. Результат применения вейвлета Хаaара

Математический аппарат вейвлета Добеши

В настоящее время вейвлет Добеши 4-ого порядка является наиболее используемым. Функция данного вейвлета и его масштабирующая функция представлены на рисунке 2.47.

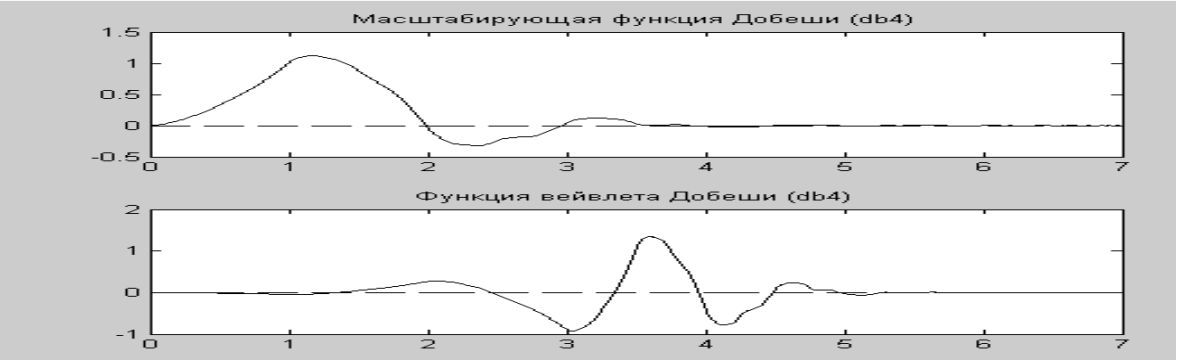


Рис. 2.47. Масштабирующая функция и функция вейвлета Добеши 4-го порядка

Масштабирующая функция и функция вейвлета Добеши задаются рекурсивно. Кроме того, функции удовлетворяют следующему требованию: масштабирующая функция имеет компактный носитель – отрезок [0 3], а также должна быть равна нулю вне этого отрезка. Эту функцию, как и вейвлет, можно задать рекурсивно, определив начальные условия.

Начальные значения:

$$\varphi(0)=0,\varphi(1)=\frac{1+\sqrt{3}}{2},\varphi(2)=\frac{1-\sqrt{3}}{2},\varphi(3)=0 \tag{2.17}$$

И рекурсивное соотношение:

$$\begin{aligned}&\varphi(r)=\frac{1+\sqrt{3}}{4}\varphi(2r)+\frac{3+\sqrt{3}}{4}\varphi(2r-1)+\frac{3-\sqrt{3}}{4}\varphi(2r-2)+\\&\frac{1-\sqrt{3}}{4}\varphi(2r-3)=\\&h_0\varphi(2r)+h_1\varphi(2r-1)+h_2\varphi(2r-2)+h_3\varphi(2r-3)=\\&(h_0,h_1,h_2,h_3)\cdot(\varphi(2r),\varphi(2r-1),\varphi(2r-2),\varphi(2r-3)).\end{aligned} \tag{2.18}$$

Сумма начальных значений равна единице.

$$\varphi(0)+\varphi(1)+\varphi(2)+\varphi(3)=0+\frac{1+\sqrt{3}}{2}+\frac{1-\sqrt{3}}{2}+0=1 \tag{2.19}$$

Далее функция вычисляется по шагам. Для этого на шаге 1 применяются условие компактного носителя, начальные значения и рекурсивные соотношения. На последующих шагах также применяется рекурсия. Для построения вейвлета также используется рекурсивное соотношение.

$$\psi(r)=-\frac{1+\sqrt{3}}{4}\varphi(2r-1)+\frac{3+\sqrt{3}}{4}\varphi(2r)-\frac{3-\sqrt{3}}{4}\varphi(2r+1)+\frac{1-\sqrt{3}}{4}\varphi(2r+2) \tag{2.20}$$

Использование рекурсивных соотношений при построении вейвлетов и масштабирующих функций способствует тому, что построенные кривые являются

самоподобными, то есть обладают фрактальными свойствами. Это делает возможным применение вейвлетов к анализу фрактальных последовательностей.

Фильтры $\{h_n\}n \in Z$ и $\{g_n\}n \in Z$ вейвлетов $\phi(x)$ и $\psi(x)$ есть коэффициенты масштабирующих уравнений $\phi(x) = \sqrt{2}\sum_n h_n \phi(2x-n)$ и $\psi(x) = \sqrt{2} g_n \phi(2x-n)$, где $g_n = (-1)^n \bar{h}_{N-n}$ (черта обозначает комплексное сопряжение). Фильтр h называется низкочастотным, а фильтр g – высокочастотным. Для восстановления используются фильтры $\{h_n\}$ и $\{g_n\}$ вейвлетов ϕ и ψ, а для восстановления с помощью фильтров – сопряженные (транспонированные) фильтры.

Математический аппарат непрерывного вейвлет-преобразования

Запись непрерывного вейвлет-преобразования (НВП) выглядит следующим образом:

$$Wf(a,b) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-b}{a} \right) dt \tag{2.21}$$

Областью определения вейвлет-преобразования Wf является множество:

$$\boldsymbol{R^2\_} = \{(a,b) \mid a \in \boldsymbol{R}^*, b \in \boldsymbol{R}\} \tag{2.22}$$

Параметр a определен в комплексной плоскости, параметр *b* – в области действительных чисел. В теории вейвлетов ось *a* масштабируется вертикально, *a* ось *b* – горизонтально. Часто область определения Wf ограничивается положительными значениями параметра *a*.

Главной задачей является нахождение вейвлет-коэффициентов. Использование свойств вейвлет-преобразования позволяет осуществить анализ сигналов. Например, слишком большие амплитуды значений вейвлет-коэффициентов указывают на положение перепадов, которые имеют резкие изменения интенсивности изображения. Различные масштабы описывают контуры структуры изображения меняющихся размеров. Такое выделение перепадов эффективно для распознавания образов при компьютерной визуализации [4].

Представляется необходимым продемонстрировать действие НВП на примере вейвлета Хаара.

Вейвлет Хаара может быть записан как:

$$\psi\left(\frac{t-b}{a}\right) = \begin{cases} 1, \left(b \le t \le b + \frac{a}{2}\right) \\ -1, \left(b + \frac{a}{2} \le t < b + a\right) \\ 0 \end{cases} \tag{2.23}$$

Вейвлет-преобразование определяется как:

$$Wf(a,b) = \frac{1}{\sqrt{a}}\left(\int_{b}^{b+a/2} f(t)dt - \int_{b+a/2}^{b+a} f(t)dt \right) = \frac{\sqrt{a}}{2}\left(\frac{2}{a}\int_{b}^{b+a/2} f(t)dt - \frac{2}{a}\int_{b+a/2}^{b+a} f(t)dt \right). \quad (2.24)$$

Это означает, что за исключением нормирующего множителя значение Wf(*a*,*b*) представляет собой разность между двумя средними значениями функции *f*, эти средние значения берутся по двум соседним интервалам длины *a*/2 в окрестности точки *b*.

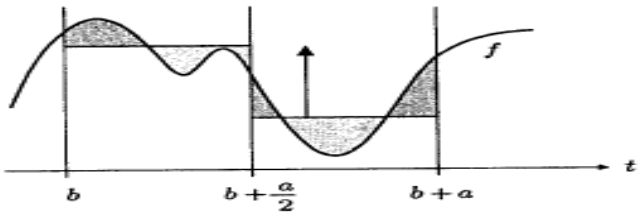


Рис. 2.48. Результат НВП в случае вейвлета Хаара

Величину Wf(a,b) можно рассматривать с другой точки зрения:

$$Wf(a,b) = \frac{1}{\sqrt{a}}\int_{a}^{b+a/2}\left(f(t) - f(t+\frac{a}{2}) \right) = -\frac{1}{\sqrt{a}}\int_{b}^{b+a/2}\left(\int_{t}^{t+a/2} f'(r)dr \right) = \ldots \quad (2.25)$$

$$= -\frac{1}{\sqrt{a}}\int_{a/2}^{a/2}\left(\frac{a}{2} - |r| \right) f'\left(b + \frac{a}{2} + r \right) dr$$

В этой форме величина Wf(*a*,*b*) проявляется как взвешенное среднее производной f’ на интервале [*b*,*b*+*a*].

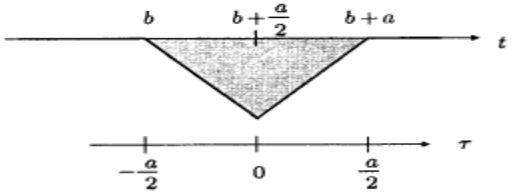


Рис. 2.49. Другая интерпретация величины Wf

Подставляя вместо f(t) конкретную функцию, можно вычислить ее НВП.

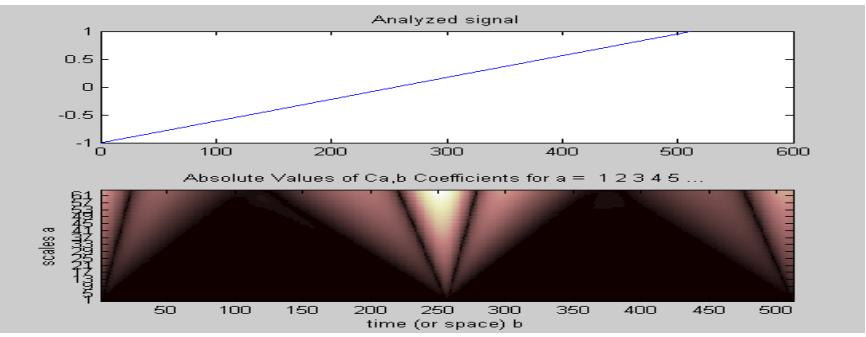


Рис. 2.50. Результат НПВ сигнала

По полученной спектрограмме можно судить о характере сигнала, например, наблюдать максимумы «всплесков энергии» в различные промежутки времени (показатель *b*) и при различных масштабах (показатель *a*).

Математический аппарат дискретного вейвлет-преобразования

Для обработки и просмотра сигналов и изображений при различных разрешениях также используется дискретное вейвлет-преобразование.

В общем виде прямое дискретное преобразование изображения f(x,y) размерности MxN можно выразить в общем виде следующим образом:

$$T(u,v,...) = \sum_{x,y} f(x,y) g_{u,v,...}(x,y) \qquad (2.26)$$

Здесь x и y – пространственные переменные (координаты пикселей изображения), а u,v – переменные в частотной области, характеризующие закономерность распределения пикселей изображения. Зная T(u,v…), можно восстановить функцию f(x,y) с помощью обратного преобразования:

$$f(x,y) = \sum_{u,v,..} T(u,v,...) h_{u,v,...}(x,y). \qquad (2.27)$$

Члены последовательностей g и h называются прямыми и обратными ядрами преобразования, определяющими природу, сложность и эффективность пары преобразований. Дискретное вейвлет-преобразование, в отличие от преобразования Фурье, обозначает целый класс преобразований, которые различаются не только своими ядрами (коэффициентами), но и свойствами этих коэффициентов (*с каким классом вейвлетов мы будем работать*) и способом применения этих коэффициентов (сколько требуется вычислить *различных решений*).

Можно охарактеризовать каждое ДВП с помощью ядра преобразования или, основываясь на множестве параметров, которые однозначно определяют пару ядер. Все преобразования являются родственными (их функции разложения представляют собой «маленькие волны» или «вейвлеты»), которые имеют переменную частоту колебаний и ограниченную длительность.

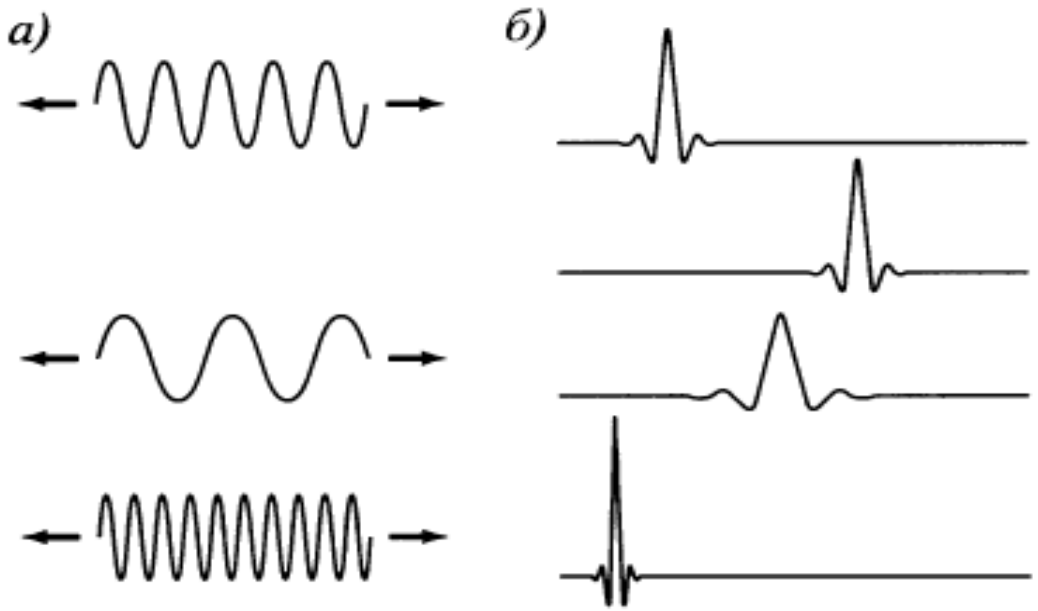


Рис. 2.51. а) Множество функций разложения Фурье – синусоиды с переменной частотой и бесконечной длительностью б) Функции разложения ДВП – «маленькие волны» с переменной частотой и конечной длительностью

Свойства «ядер», являющиеся общими для всех типов вейвлетов.

Свойство 1. *Разделимость, масштабируемость и переносимость*. Ядра можно представить в виде трех разделимых двумерных вейвлетов:

$$\begin{aligned} \psi^H(x,y) &= \psi(x)\varphi(y), \\ \psi^V(x,y) &= \varphi(x)\psi(y), \\ \psi^D(x,y) &= \psi(x)\psi(y). \end{aligned} \tag{2.28}$$

Здесь $\psi^H(x,y), \psi^V(x,y), \psi^D(x,y)$ называются горизонтальными, вертикальными и диагональными вейвлетами, а двумерная функция:

$$\varphi(\mathrm{x},\ \mathrm{y}) = \varphi(\mathrm{x})\varphi(\mathrm{y}) \tag{2.29}$$

Называется масштабирующей функцией. Каждая из этих двумерных функций является произведением двух одномерных масштабирующих функций [5].

$$\begin{aligned} \varphi_{j,k}(x) &= 2^{j/2}\varphi(2^j x - k), \\ \psi_{j,k}(x) &= 2^{j/2}\psi(2^j x - k). \end{aligned} \tag{2.30}$$

Параметры i и j являются целыми числами. Число k определяет положение одномерных функций на оси x, масштаб j – ширину по оси x, множитель 2j/2 отвечает за высоту и амплитуду.

Свойство 2. Кратномасштабная совместимость. Свойство оговаривает требования к выбранной масштабирующей функции.

а) Каждая функция $\varphi_{i,j}(x)$ ортогональная любому своему целочисленному сдвигу.

б) Множество функций, которые можно представить в виде рядов разложения по $\varphi_{j,k}(x)$ при малых масштабах (j мало), принадлежит множеству функций, представимых в более высоких значениях.

в) Каждую масштабирующую функцию можно представить с произвольной точностью при $j \to \infty$.

Свойство 3. Ортогональность. Функции разложения $\{\varphi_{i,j}(x)\}$ образуют ортогональный или биортогональный базис (отсюда и название еще одного типа вейвлетов – ортогональные и биортонональные) в пространстве одномерных измеримых, суммируемых в квадрате функций. Базис обладает тем свойством, что каждая представимая в этом базисе функция имеет единственный набор коэффициентов. Для вещественных ортогональных ядер имеет место равенство g = h. В биортогональном случае

$$\langle h_r, g_s \rangle = \delta_{rs} = \begin{cases} 1, r = s. \\ 0 \end{cases} \qquad (2.31)$$

Для практического применения важно следствие из приведенных свойств – функцию вейвлета и масштабирующую функцию можно представить *в виде линейной комбинации своих же копий с удвоенным разрешением:*

$$\varphi(x) = \sum_n h_\varphi(n)\sqrt{2}\varphi(2x-n), \qquad \psi(x) = \sum_n h_\psi(n)\sqrt{2}\psi(2x-n) \qquad (2.32)$$

В среде MATLAB дискретное вейвлет-преобразование реализуется с помощью процедур dwt и fwt (быстрое вейвлет-преобразований). Можно выполнять разложение как двумерных, так и одномерных сигналов.

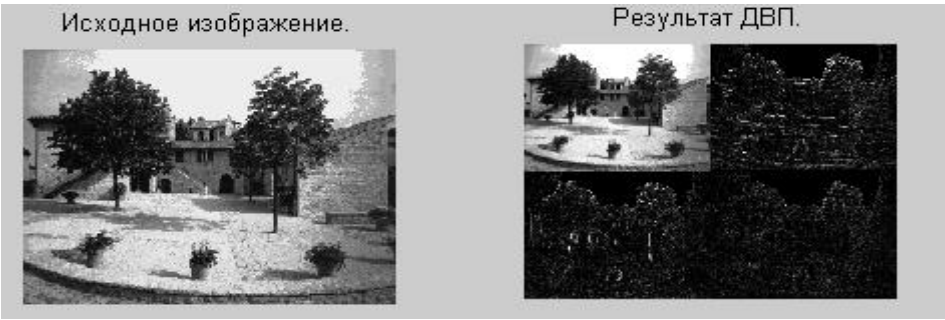


Рис. 2.52. Результат дискретного вейвлет-преобразования

Алгоритм быстрого вейвлет-преобразования

Обобщая преобразование на двумерный и многомерный уровень, можно представить схему преобразования (быстрого вейвлет-преобразования).

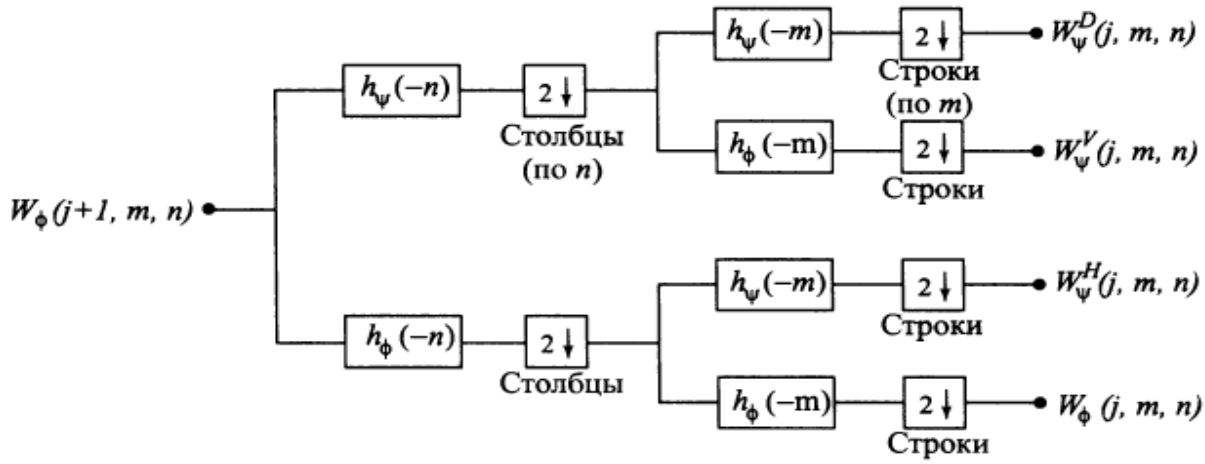


Рис. 2.53. Схема быстрого вейвлет-преобразования

Сначала обрабатываются столбцы, а затем строки. Используются при этом фильтры низких и высоких частот.

Каждый проход через блок фильтров на рис. 10 разлагает входные данные на четыре компоненты меньшего диапазона (или масштаба). Коэффициенты W_φ преобразования получаются двумя проходами низкочастотной фильтрации (т. е. с фильтром h_φ), и поэтому они называются коэффициентами приближения; коэффициенты $W_\psi^i, i = H, V, D$ называются, соответственно, коэффициентами горизонтальных, вертикальных и диагональных деталей. Поскольку само изображение f(x,y) представлено в наивысшем разрешении, то оно становится входом $W_\varphi(j+1, m, n)$ первой итерации процедуры. Кроме того, здесь имеется три переменные преобразованного пространства: масштаб j, горизонтальная трансляция п и вертикальная трансляция т. Эти переменные соответствуют обозначениям u,v, … в первых двух уравнениях (8), (9). Далее посмотрим примеры применения двумерного вейвлет-преобразования (dwt2, fwt) при обработке изображений.

При выполнении этапов преобразования используется операция свертки сигналов. Слово «свертка» обозначает совместное сворачивание двух величин или функций [2]. Дискретная свертка векторов f и g обозначается f*g и вычисляется с помощью соотношения:

$$(f * g)_i = \sum_j f_j g_{i-j} \qquad (2.33)$$

В результате выполнения операции получается один вектор, охватывающий диапазон входного сигнала. Поэтому вейвлет-преобразования называют ***поддиапазонными***. Дискретная свертка двух числовых последовательностей f(i) и g(i) задается неравенством:

$$h(i) = f(i) * g(i) = \sum_j f(i)g(i-j)$$

Свертка сигналов применяется, например, для удаления шума или сглаживания сигналов.

Использование приложения Wavelet Toolbox Matlab для сжатия информации путем вейвлет-преобразований [12] .

Пакет MATLAB Wavelet Toolbox представляет собой совокупность программ, позволяющих выполнять вейвлет-анализ и обработку сигналов средствами графического интерфейса пользователя. Пакет позволяет продемонстрировать практические приложения теории вейвлетов и вейвлет-преобразований. Остановимся на следующих пунктах:

1. Просмотр вейвлетов.
2. Одномерный дискретный вейвлет-анализ.
   - 2.1. Вейвлет-разложение сигнала.
   - 2.2. Интерпретация статистических характеристик сигнала.
   - 2.3. Гистограммы коэффициентов аппроксимации и детализации.
3. Сжатие одномерных сигналов.
4. Удаление шума из сигналов.

Главное меню пакета вызывается из MATLAB с помощью команды wavemenu.

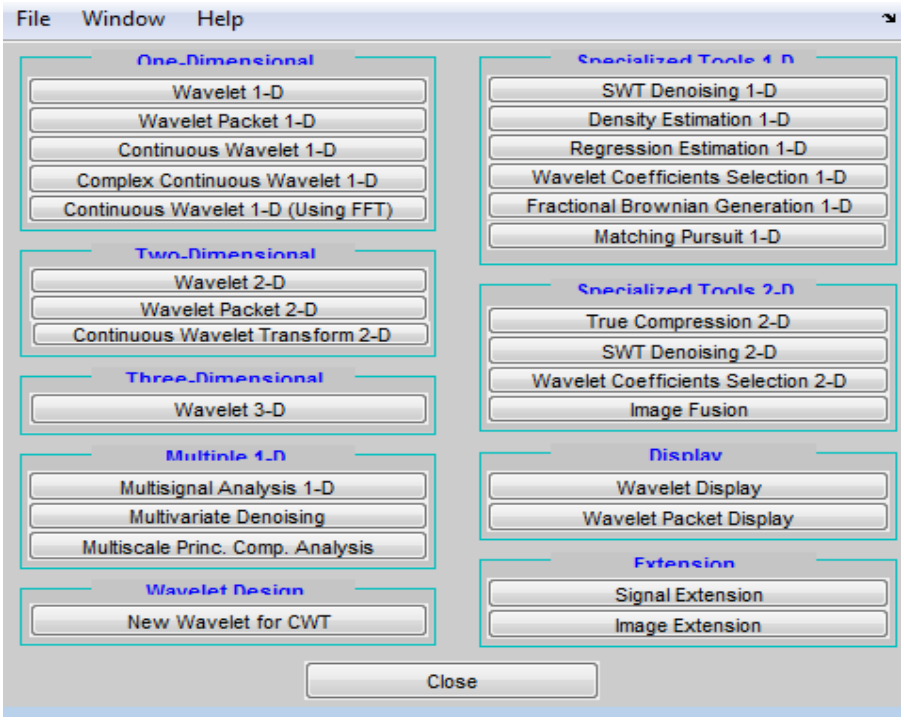


Рис. 2.54. Главное окно Wavelet Toolbox

Способы просмотра вейвлетов в Wavelet Toolbox

Способы просмотра вейвлетов – группа Display (кнопка Wavelet Display). Откроется следующее окно:

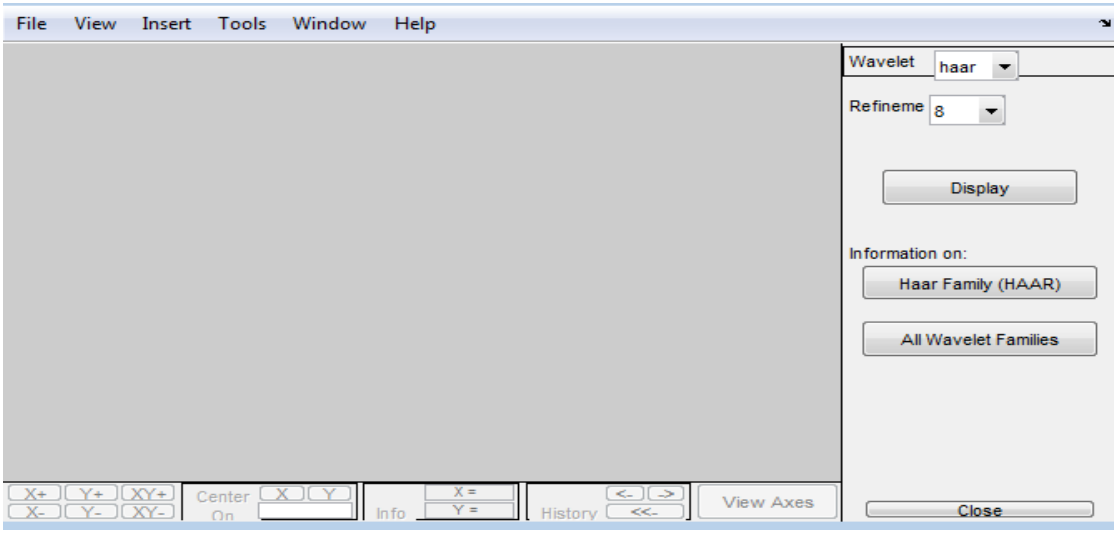


Рис. 2.55. Окно раздела Wavelet Display после запуска

Вверху расположено обычное меню, справа и внизу – меню, позволяющие управлять отображением нужной информации.

В правом меню можно из раскрывающегося списка тип вейвлета, задать уровень разрешения (параметр *Refinement*)

– определить шаг вычисления значений вейвлетов (этот параметр равен $1/2^N$, N по умолчанию равно 8).

Также можно посмотреть информацию по выбранному типу вейвлета.

Выбрав из списка тип вейвлета ‘haar’, следует нажать кнопку *Display(Refinement = 8).*

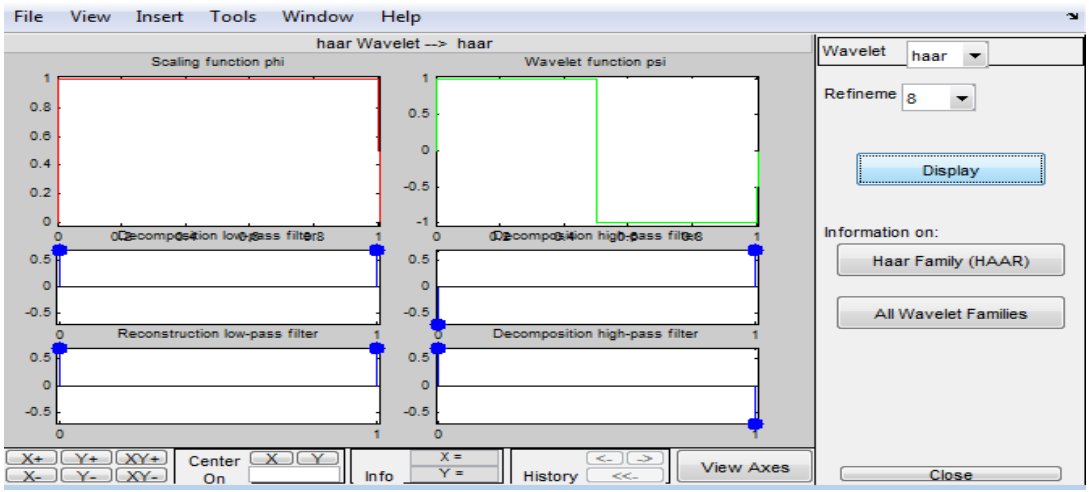


Рис. 2.56. Вейвлет и масштабирующая функция Хаара, фильтры восстановления и разложения

Вверху расположены графики масштабирующей функции и функции вейвлета. Ниже изображены фильтры восстановления и разложения вейвлета (низкочастотные и высокочастотные). Для восстановления используются фильтры (коэффициенты масштабирующих уравнений) h и g функций $\varphi(x)$ и $\psi(x)$, а для вейвлет-разложения – сопряженные (транспонированные) фильтры. Это хорошо видно на графиках фильтров. Для вейвлета Хаара фильтры простые, так как базовыми операциями являются нахождение средних и полуразностей. У других вейвлетов фильтры более сложные по своей структуре.

Например, если выбрать вейвлет Добеши 7-ого порядка, то получится следующее изображение:

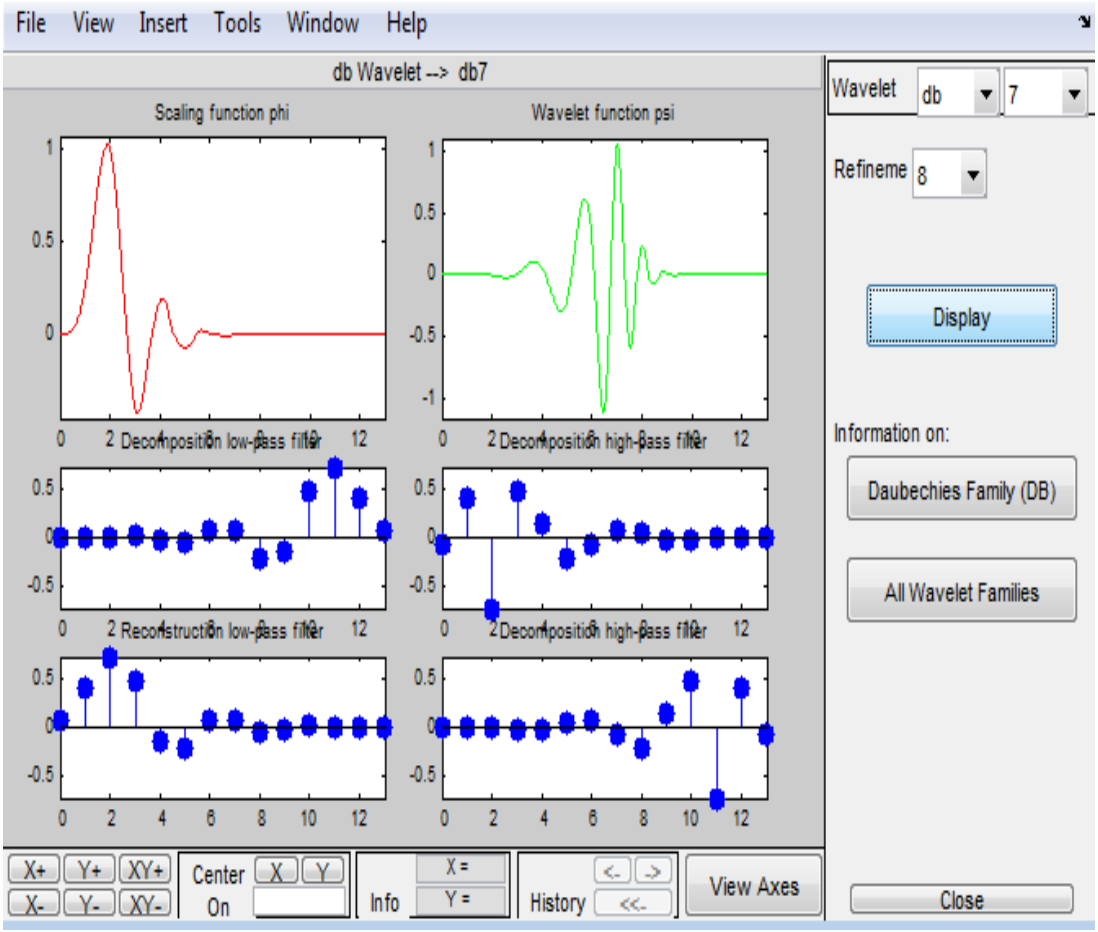


Рис. 2.57. Вейвлет 7-го порядка и масштабирующая функция Добеши, фильтры восстановления и разложения

Сжатие сигналов при помощи вейвлет-преобразований в Wavelet Toolbox

В главном окне необходимо нажать кнопку Wavelet 1-D, появится диалоговое окно, интерфейс которого позволяет осуществлять обработку одномерных сигналов.

Загрузка сигнала Frequency Breakdown (Load →Example Analysis → Basic Signals → Frequency Breakdown):

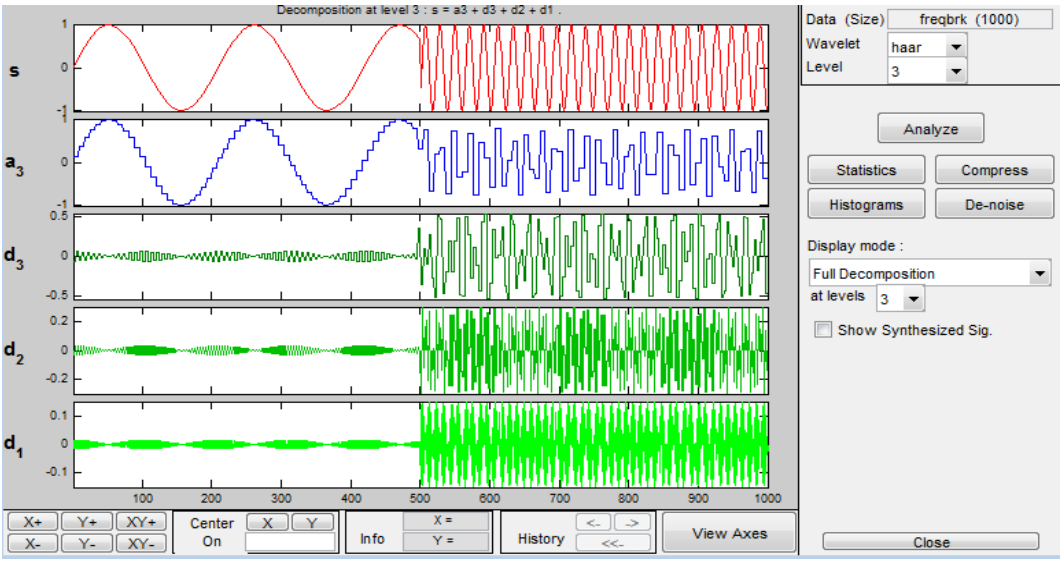


Рис. 2.58. Разложение сигнала

Кнопка ***compress*** позволяет осуществить сжатие одномерного сигнала.

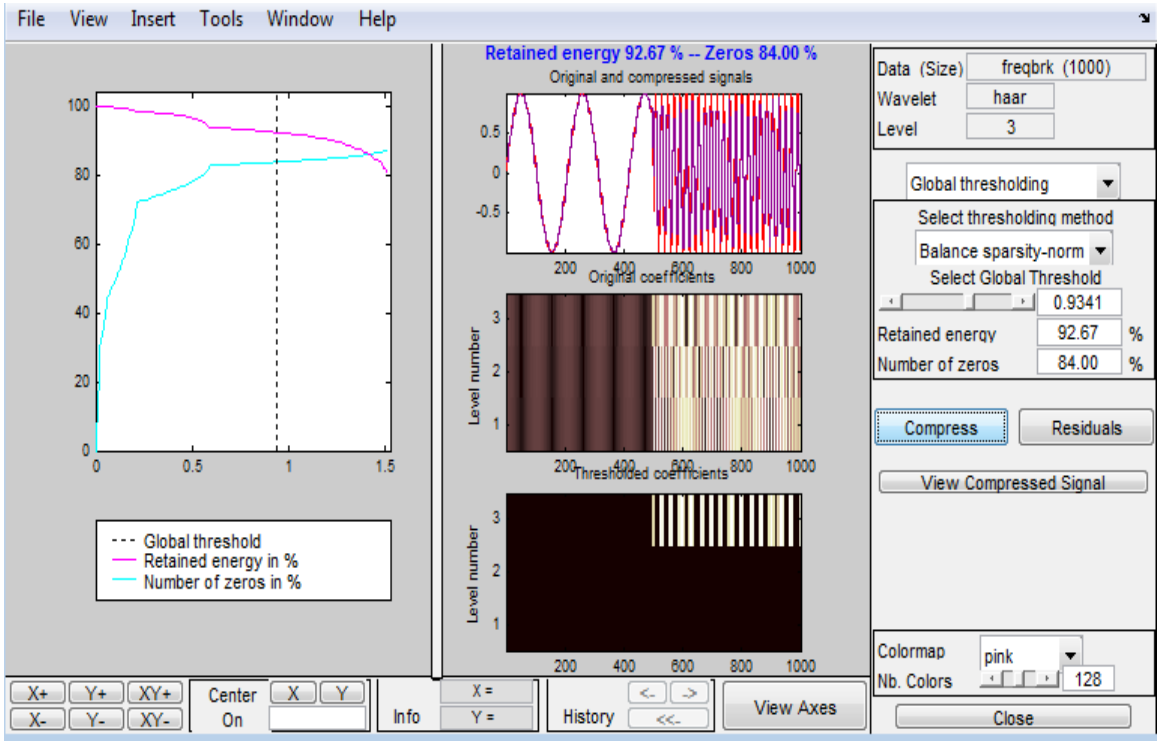


Рис. 2.59. Интерфейс окна, вызываемого нажатием кнопки ***Compress***

При сжатии одномерных сигналов используется метод глобальной пороговой обработки детализирующих коэффициентов. Управлять сжатием можно группой компонентов справа, устанавливая ползунком значение. Задав порог (процент сохраняемых коэффициентов), в результате получается сжатый сигнал.

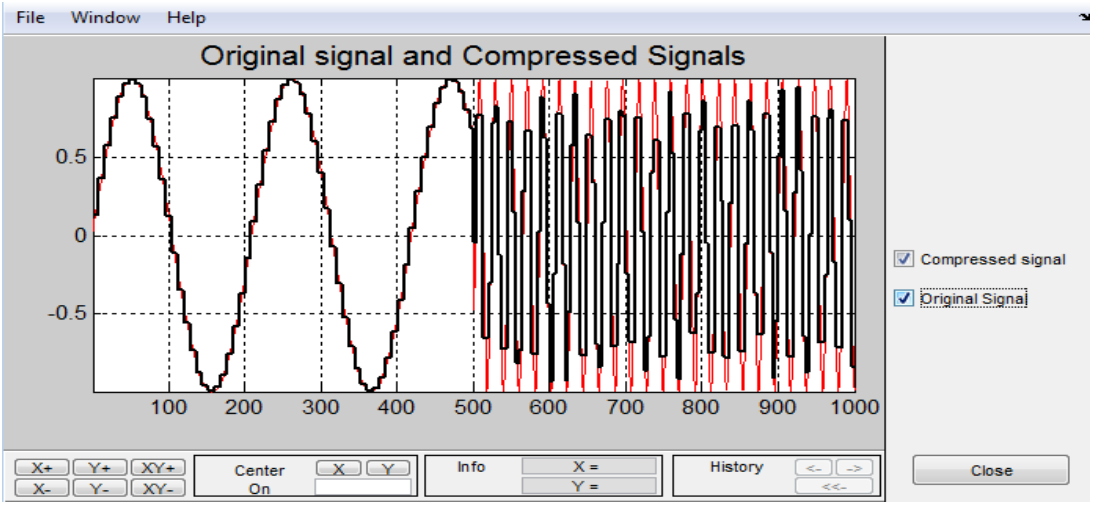


Рис. 2.60. Исходный и сжатый сигнал при помощи вейвлет-преобразования Хаара третьего уровня

Сжатие одномерного сигнала при помощи различных вейвлет-преобразований

На сегодняшний день существует большое количество математических алгоритмов вейвлет-преобразований, данный раздел содержит рассмотрение трех основных и сравнительный анализ их характеристик.

Вейвлет Хаара пятого уровня. Разложение исследуемого сигнала при помощи вейвлета Хаара пятого уровня представлено на рисунке 2.61

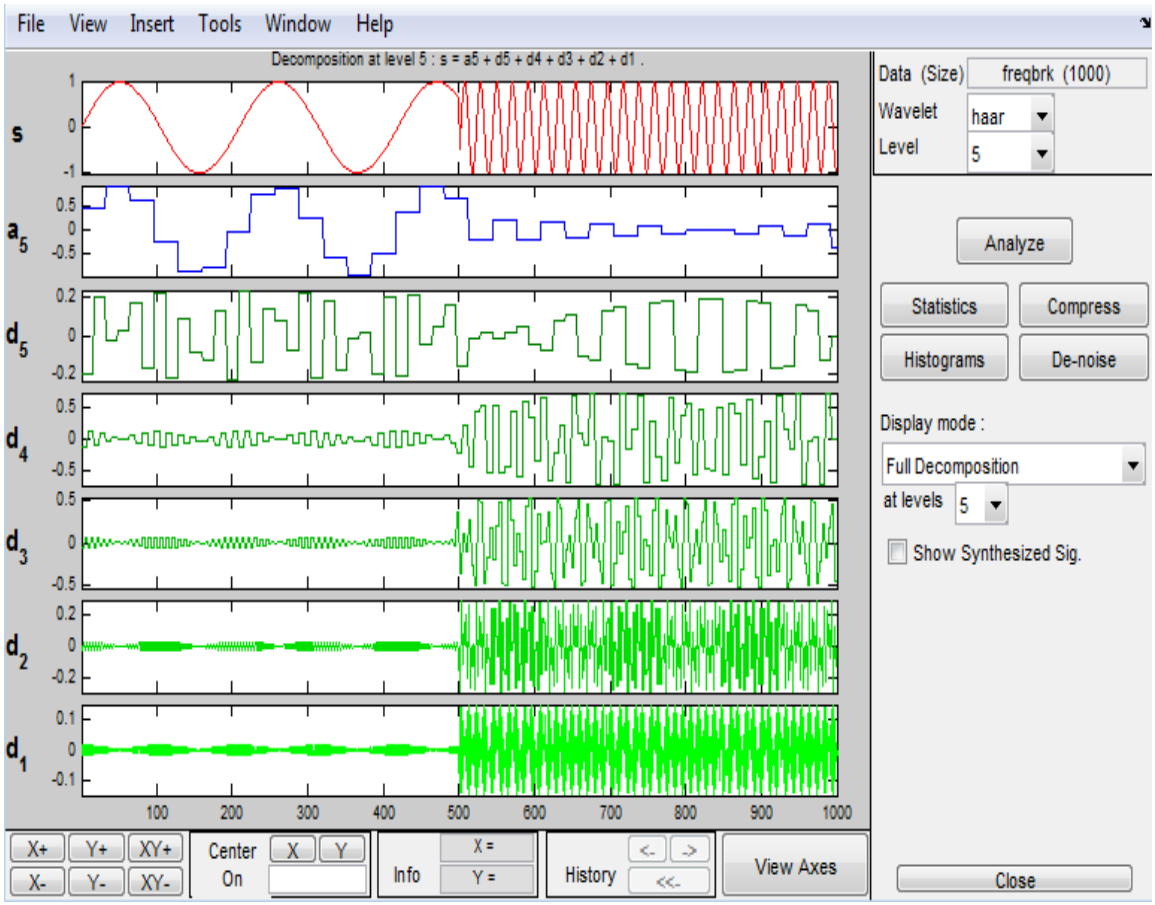


Рис. 2.61. Разложение сигнала при помощи вейвлета Хаара пятого уровня

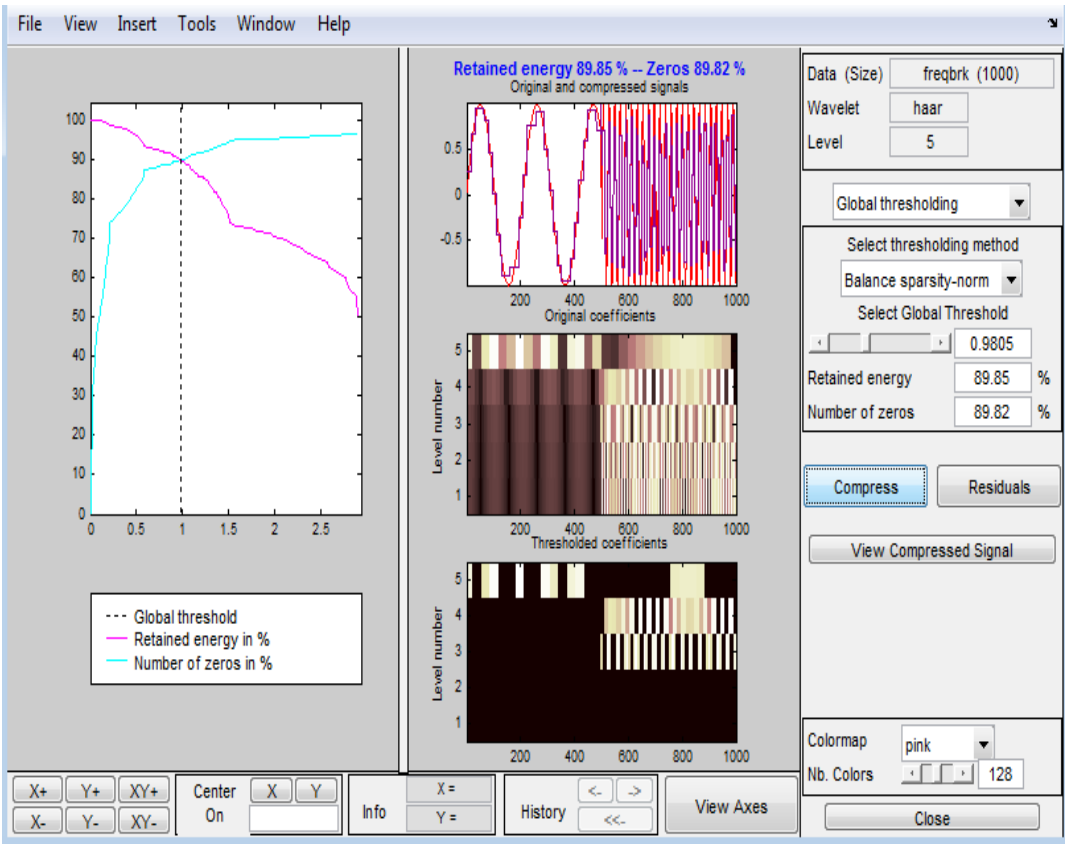


Рис. 2.62. Сжатие сигнала при помощи вейвлета Хаара пятого уровня

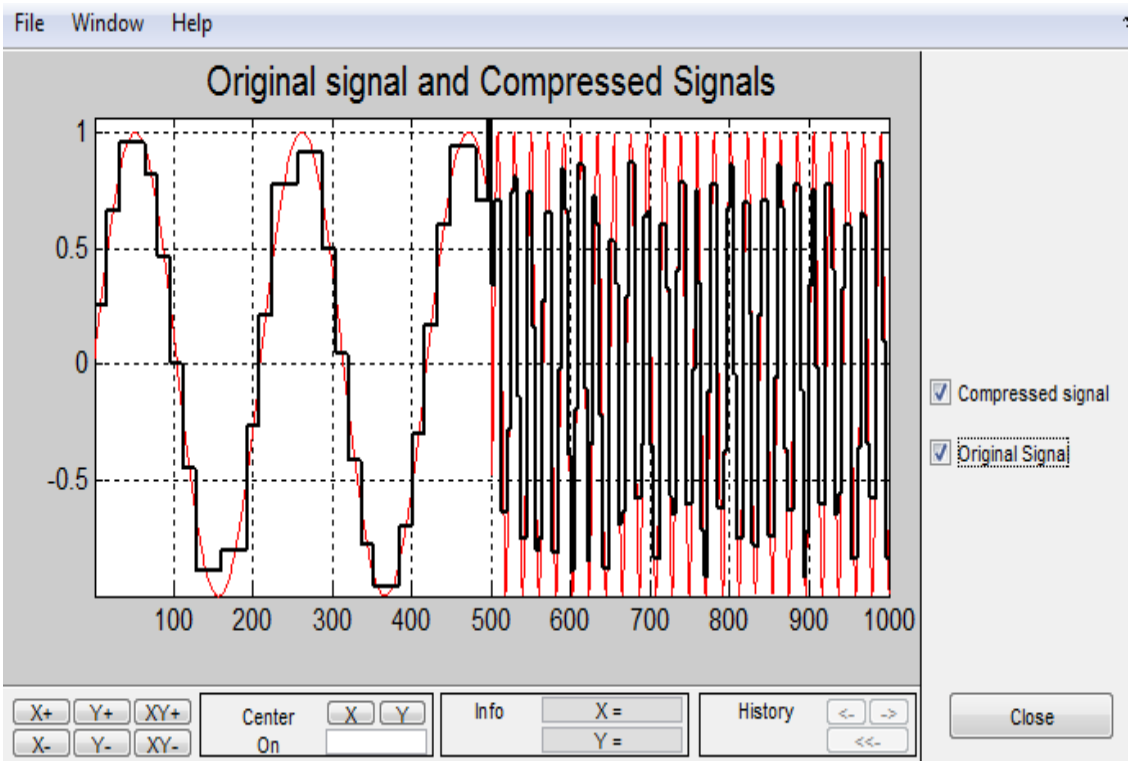


Рис. 2.63. Исходный и сжатый сигнал при помощи вейвлет-преобразования Хаара пятого уровня при коэффициенте сжатия 0.98

Далее необходимо проанализировать зависимость количества сохраненной энергии сигнала от коэффициента сжатия, результаты представлены в таблице 2.7.

Таблица 2.7. Зависимость количества сохраненной энергии от коэффициента сжатия для вейвлет-преобразования Хаара пятого порядка

$K_{сж}$	0,5	1	1,5	2	2,5
$Q_{сохр}$	95,93	89,47	75,37	70,36	65,32

Данную зависимость удобно представить в виде графика (рисунок 2.64):

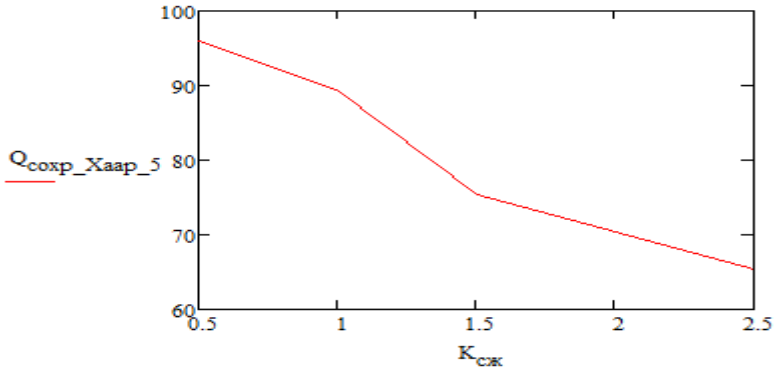


Рис. 2.64. Зависимость количества сохраненной энергии от коэффициента сжатия для вейвлет-преобразования Хаара пятого уровня

Вейвлет Добеши пятого уровня и пятого порядка. Разложение исследуемого сигнала при помощи вейвлета Добеши пятого уровня и пятого порядка представлено на рисунке 2.65.

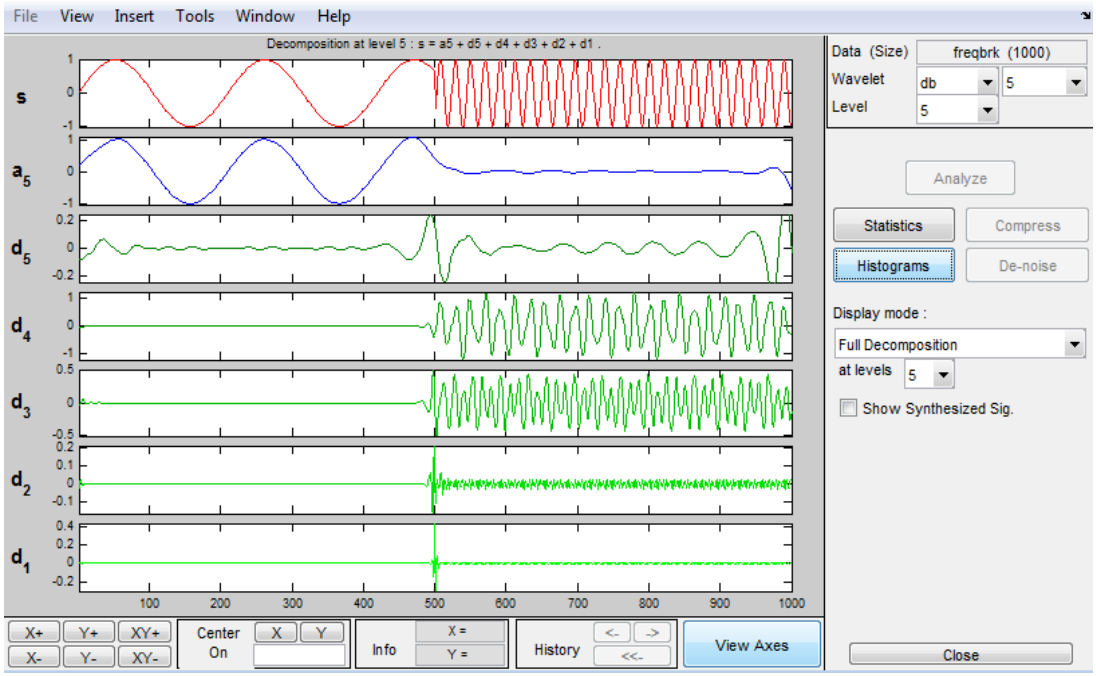


Рис. 2.65. Разложение сигнала при помощи вейвлета Добеши пятого уровня и пятого порядка

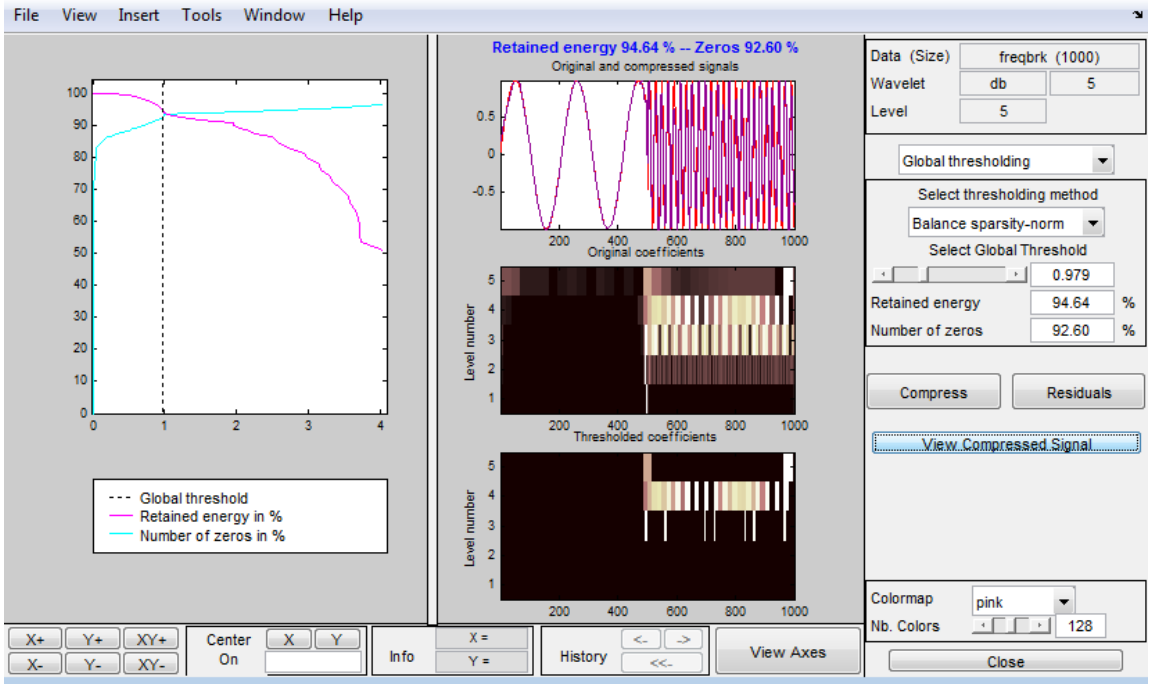


Рис. 2.66. Сжатие сигнала при помощи вейвлета Добеши пятого уровня и пятого порядка

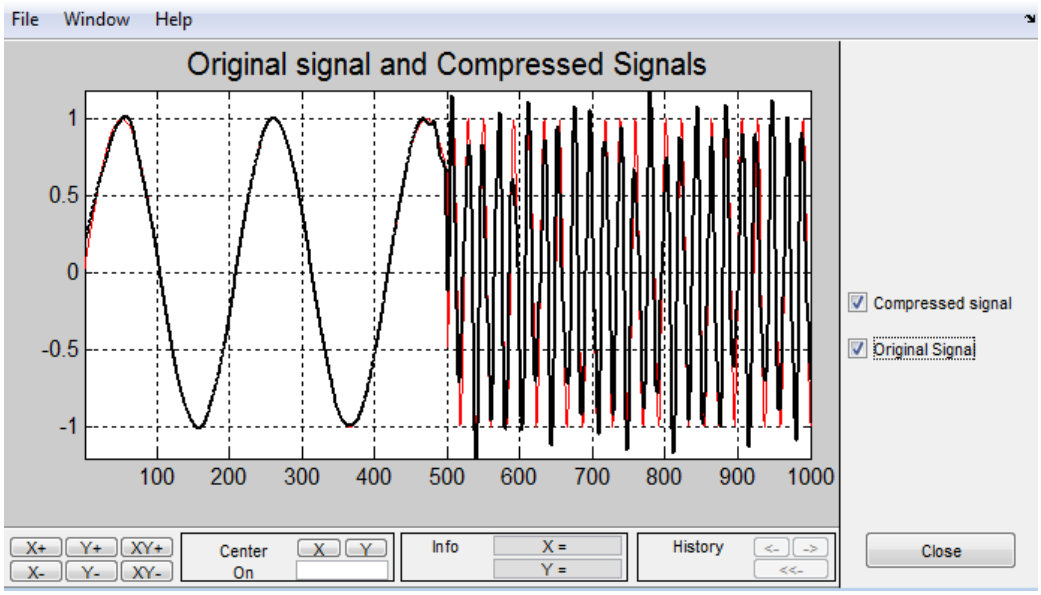


Рис. 2.67. Исходный и сжатый сигнал при помощи вейвлет-преобразования Добеши пятого уровня и пятого порядка при коэффициенте сжатия 0.98

Далее необходимо проанализировать зависимость количества сохраненной энергии сигнала от коэффициента сжатия, результаты представлены в таблице 2.7

Таблица 2.7 – Зависимость количества сохраненной энергии от коэффициента сжатия для вейвлет-преобразования Добеши пятого порядка

$K_{сж}$	0,5	1	1,5	2	2,5
$Q_{сохр}$	99,35	93,61	91,81	89,95	86,41

Данную зависимость удобно представить в виде графика (рисунок 2.68)

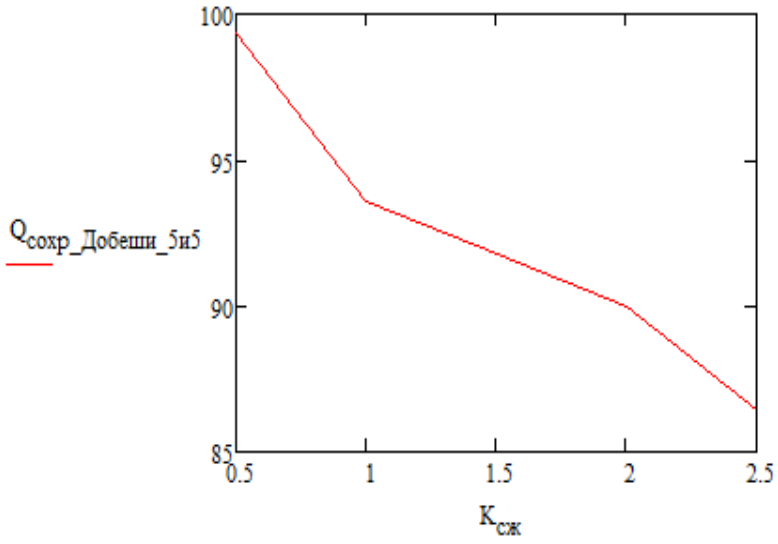


Рис. 2.68. Зависимость количества сохраненной энергии от коэффициента сжатия для вейвлет-преобразования Добеши пятого уровня и пятого порядка

Биортогональное преобразование пятого уровня и порядка 5,5. Разложение исследуемого сигнала при помощи биортогонального преобразования пятого уровня и порядка 5,5 представлено на рисунке 2.70.

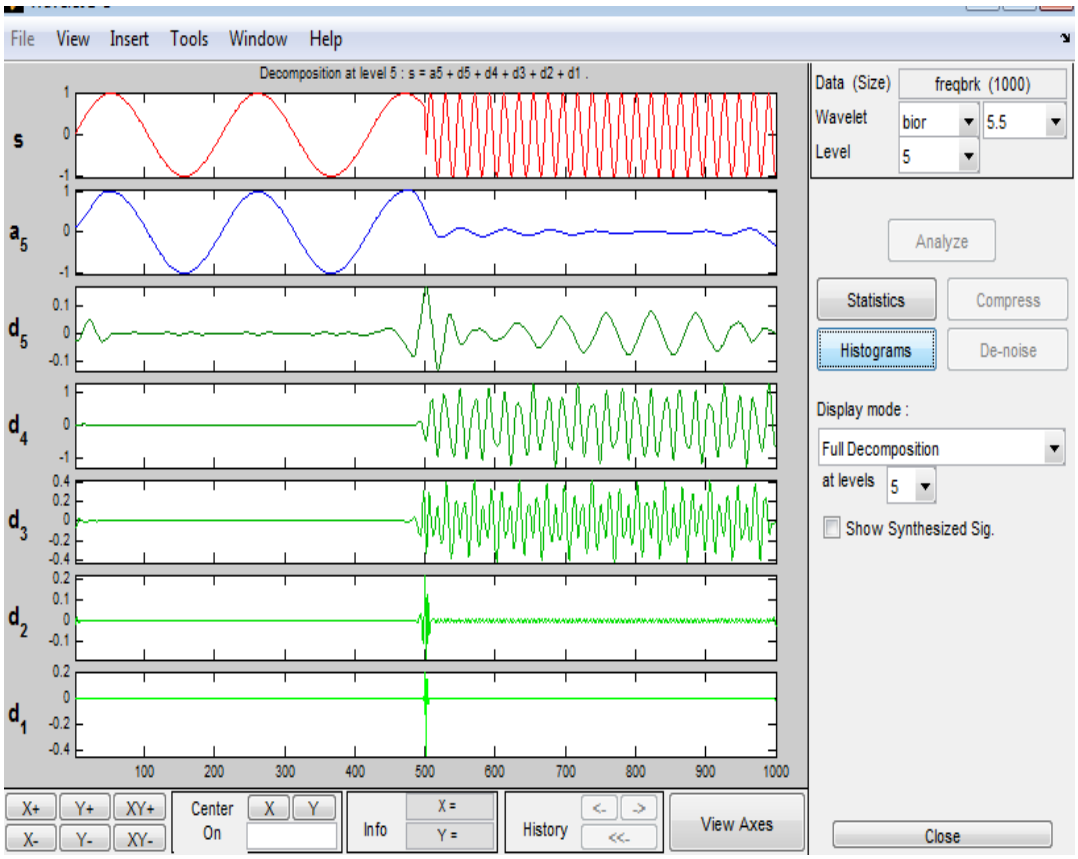


Рис. 2.70. Разложение сигнала при помощи биортогонального преобразования пятого уровня и порядка 5,5

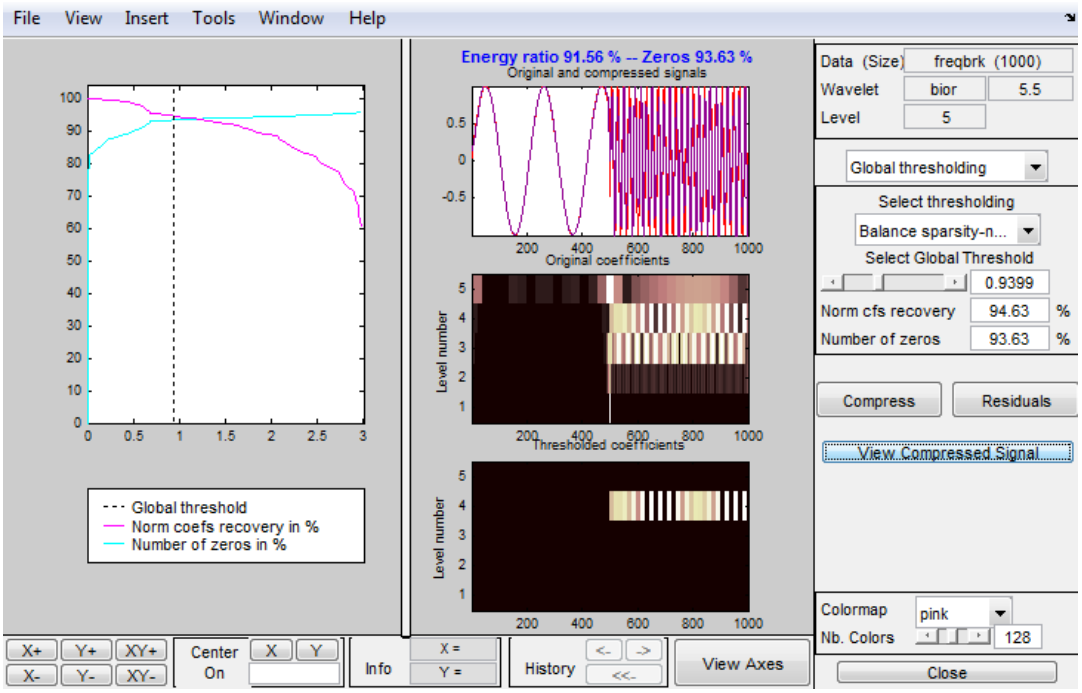


Рис. 2.71. Сжатие сигнала при биортогональном вейвлет-преобразовании пятого уровня и порядка 5,5

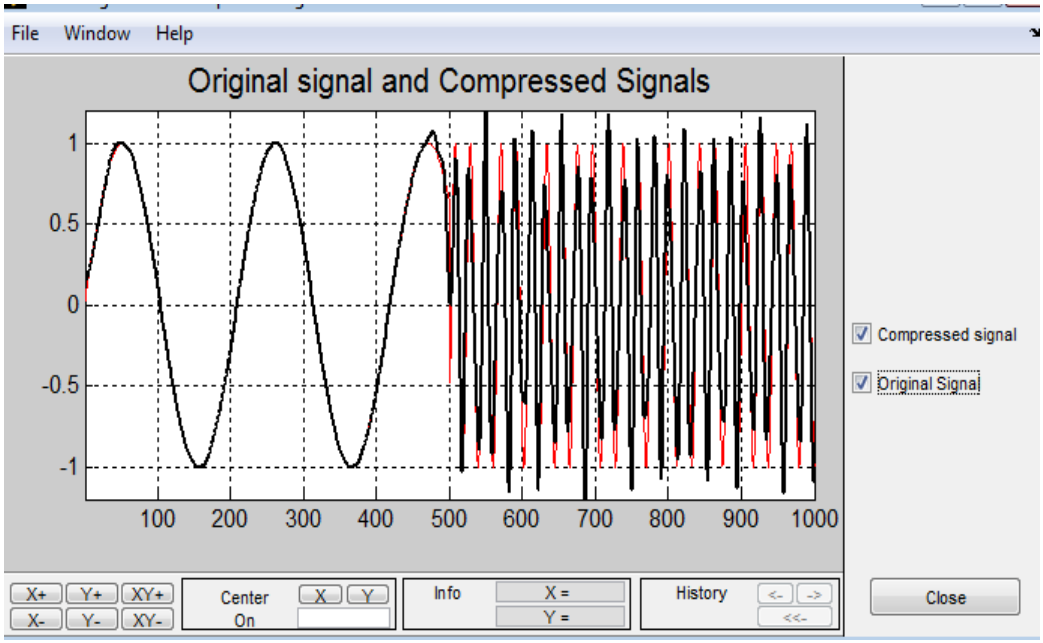


Рис. 2.72. Исходный и сжатый сигнал при помощи биортогонального вейвлет-преобразования пятого уровня и 5,5 порядка при коэффициенте сжатия 0.98

Далее необходимо проанализировать зависимость количества сохраненной энергии сигнала от коэффициента сжатия, результаты представлены в таблице 2.8.

Таблица 2.8. Зависимость количества сохраненной энергии от коэффициента сжатия для вейвлет-преобразования Хаара пятого порядка

$K_{сж}$	0,5	1	1,5	2	2,5
$Q_{сохр}$	98,68	94,63	92,71	88,65	82,28

Данную зависимость удобно представить в виде графика (рисунок 2.73):

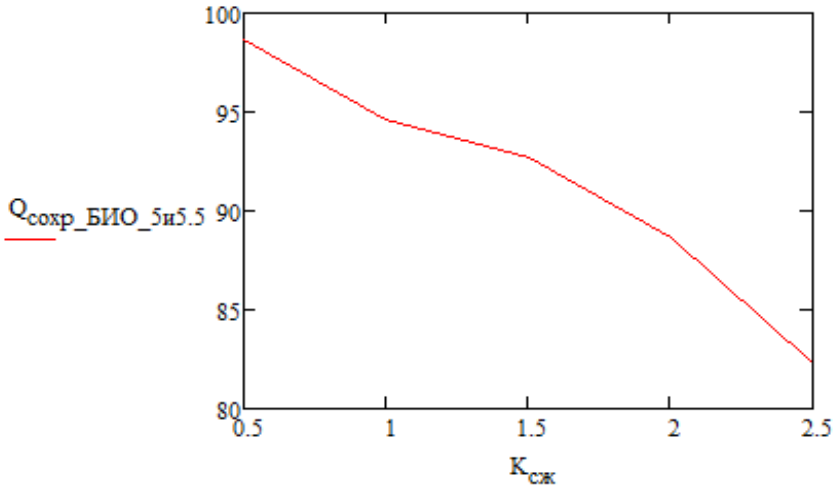


Рис. 2.73. Зависимость количества сохраненной энергии от коэффициента сжатия для биортогонального пятого уровня и 5,5 порядка

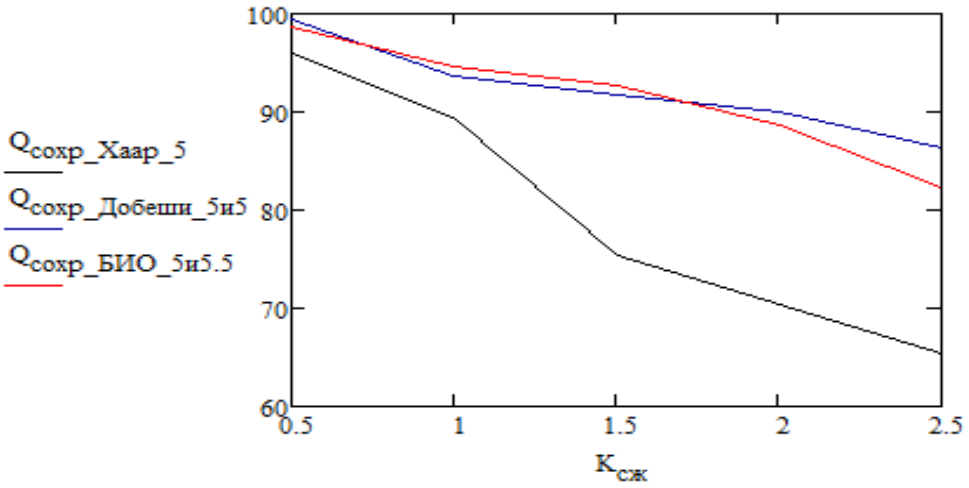


Рис. 2.74. Сравнение энергетических характеристик сжатых сигналов при различных методах вейвлет-преобразований

В результате выполнения данного индивидуального задания был проведен теоретический обзор современных алгоритмов вейвлет-преобразований для сжатия одномерных сигналов.

Было установлено, что использование вейвлет-преобразований является эффективным средством для преобразования информации для передачи по каналам связи, так в современном телекоммуникационном оборудовании предусмотрено использование помехоустойчивых алгоритмов кодирования с использованием избыточности.

Так же, были приобретены практические навыки работы с приложением Wavelet Toolbox Matlab, которое позволяет наглядно продемонстрировать принцип действия сжатия информации путем вейвлет-преобразований.

Были построены графики зависимостей количества сохраненной энергии сигнала от коэффициента сжатия для различных алгоритмов вейвлет- преобразований.

В том числе, было установлено, что использование вейвлет-преобразований Добеши и Хаара эффективнее использования биортогонального вейвлет-преобразования с точки зрения сохранной энергии сигнала.

Данная работа имеет практическое значения, так как была разработана методика для выполнения лабораторной работы по сжатию информации с использованием вейвлет-преобразований при помощи программного обеспечения Matlab 2015.

Таким образом, можно сделать обоснованный вывод о том, что вейвлет-анализ и обработка сигналов на сегодняшний день быстро развивающаяся область прикладной науки, которая в будущем может стать ведущим способом обработки информации. На основе вейвлет-преобразования могут быть построены высокоэффективные алгоритмы сжатия информации, что актуально в наши дни.

ГЛАВА 3. ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

Корректирующие коды в телекоммуникационных системах

В теории современных телекоммуникационных систем значительное внимание уделяется изучению методов кодирования информации. В технике электрической связи широко используют результаты теории кодирования.

Кодирование - операция отождествления символов или групп символов одного кода с символами или группами символов другого кода. Необходимость кодирования возникает, прежде всего, из потребности приспособить форму сообщения к данному каналу связи или к какому-либо другому устройству, предназначенному для преобразования или хранения информации.

Типичная структурная схема системы передачи дискретной информации (СПДИ) приведена на рис. 3.1. Источник вырабатывает сообщения, которые необходимо передавать по каналу СПДИ. Это могут быть последовательности дискретных сообщений (данные, телеграфные сообщения и т.д.) либо непрерывные сообщения (речь, телевидение, результаты телеизмерений и др.), преобразованные в цифровую форму.

Рис. 3.1. Структурная схема системы передачи дискретной информации

Реальные сообщения содержат избыточность и для согласования источника с каналом передачи информации используют кодер источника. Совместно с декодером они образуют кодек источника. Методы кодирования источников изучались в модуле 2. Основная задача любой СДПИ - передача информации с заданными верностью и скоростью передачи. Эти требования находятся в противоречии, причем, повышение скорости передачи информации в реальных СПДИ приводит к снижению помехоустойчивости и верности передачи.

Согласно известных теорем К. Шеннона, в принципе возможно сколь угодно большое повышение верности передачи информации, если скорость передачи по каналу R_{Kan} не превышает пропускной способности канала $C_к$. Достигается это применением достаточно длинных корректирующих кодов (КК).

Корректирующие коды - это коды, позволяющие обнаруживать или исправлять ошибки, возникающие при передаче сообщений по каналам связи.

С этой целью в структуру КК вводится ***избыточность.***

Кодек КК *(кодер и декодер канала)* приведены на рис.3.1. В реальных условиях длина кода ограничена допустимой сложностью устройств кодирования и, прежде всего, декодирования, поэтому эффект от применения корректирующих кодов зависит от параметров кода и ограничений на реализацию кодека канала.

Современная теория предлагает широкий набор корректирующих кодов, различных по структуре, принципам построения и корректирующей способности. В последующих разделах рассмотрены классы кодов, для которых разрабо-

таны достаточно простые и эффективные алгоритмы кодирования/декодирования и которые наиболее перспективны для использования в каналах телекоммуникационных систем.

Классификация корректирующих кодов

В теории и технике помехоустойчивого кодирования известно множество корректирующих кодов, которые могут быть классифицированы по различным признакам. Классификация кодов приведена на рис. 3.2.

По способу формирования КК подразделяются на блоковые и непрерывные. Формирование ***блоковых кодов*** предусматривает разбиение передаваемых цифровых последовательностей на отдельные блоки, которые подаются на вход кодера. Каждому такому блоку на выходе кодера соответствует блок кодовых символов, работа кодера определяется правилом, или ***алгоритмом кодирования***. Формирование ***непрерывных кодов*** осуществляется непрерывно во времени, без разделения на блоки, что и определяет наименование этого класса кодов. Блоковые коды исторически были предложены и изучены ранее, на заре развития теории кодирования.

В классе непрерывных кодов следует отметить ***сверточные коды***, которые по характеристикам превосходят блоковые коды, и, по этой причине, находят широкое применение в телекоммуникационных системах. Многие коды носят имена ученых, которые их предложили и исследовали. Таким примером является непрерывный код Финка-Хагельбаргера, предложенный советским ученым J1.M. Финком и немецким специалистом Р. Хагельбаргером. Длительное время этот код служил в литературе показательным примером непрерывного кода с простым алгоритмом кодирования/декодирования, но после открытия сверточных кодов уступил им место.

Для описания процедур кодирования/декодирования как блоковых, так и сверточных кодов используют адекватный математический аппарат. Для описания ***линейных кодов*** используется хорошо разработанный аппарат линейной

алгебры. Формирование нелинейных кодов производится с применением нелинейных процедур. Такой подход позволяет в некоторых случаях получить ***нелинейные коды*** с рядом специальных свойств. В теории и технике кодирования важной является проблема сложности реализации процедур кодирования/декодирования и, в особенности, процедур декодирования. Поэтому некоторые классы кодов (коды Хемминга, циклические коды Боуза-Чоудхури- Хоквингема, Рида-Соломона, Файра и др.) были разработаны совместно с алгоритмами декодирования, связанными со структурными свойствами этих кодов. И, наоборот, разработка новых алгоритмов декодирования сверточных кодов (алгоритм А. Витерби, последовательное декодирование, пороговое декодирование) инициировала поиски соответствующих сверточных кодов. Отличительные преимущества корректирующих кодов (как блоковых, так и сверточных) побуждали поиски новых подходов к реализации путей

повышения помехоустойчивости и эффективности телекоммуникационных систем. На рис. 3.2 отмечены, соответственно, ***новые методы кодирования***: сигнально-кодовые конструкции, турбокоды, пространственно-временные коды и т.п.

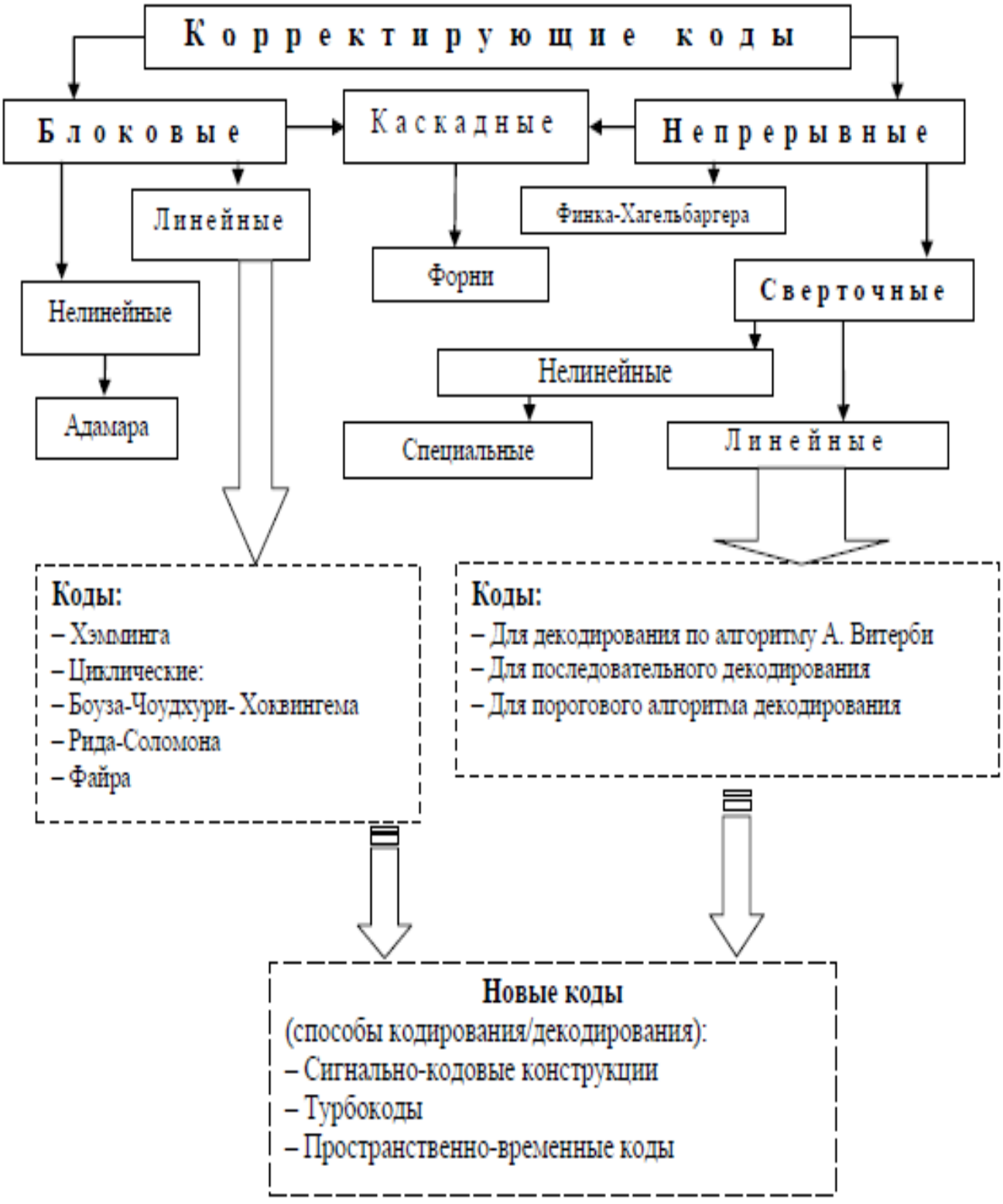


Рис. 3.2. Классификация корректирующих кодов

3.1 Исследование кодов Хемминга, БЧХ (Боуза-Чоудхури-Хоквенгема), Рида-Соломона на базе MATLAB 2015

Код Хэмминга

Наиболее известным линейным кодом является **блоковый код Хэмминга**. В данном пункте, будет рассмотрен (7,4)-код Хэмминга. Линейный блочный код, для которого выполняется неравенство Хэмминга, является кодом Хэмминга, то есть код (7,4) не единственный код, который является данным кодом.

Введем некоторые обозначения, которые составляют основу помехоустойчивого кодирования:

k – количество информационных бит

n– количество бит на выходе кодера

r – количество проверочных(избыточных) бит

Вместо ***k*** бит информационного вектора в канал передается ***n*** бит кодового вектора. В этом случае говорят об избыточном кодировании со скоростью: $R = \frac{n}{k}$.

Чем ниже скорость, тем больше избыточность кода, и тем большими возможностями для защиты от ошибок он обладает. Однако, следует учитывать, что с увеличением избыточности затраты на передачу информацию также возрастают.

В кодировании и декодировании участвуют генераторная и проверочная матрицы, структура которых следующая:

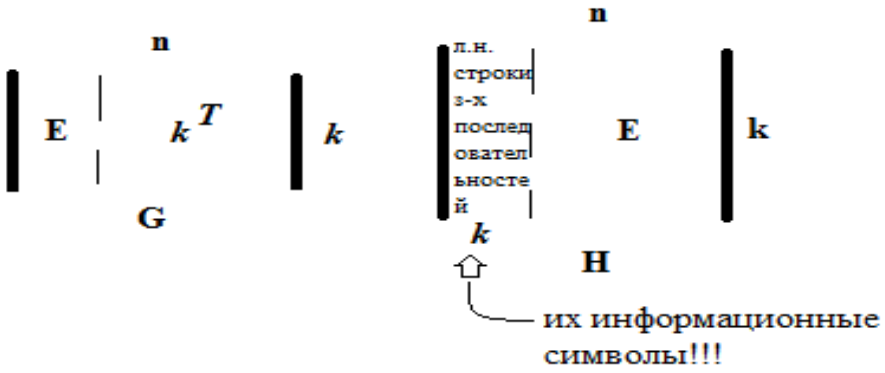


Рис. 3.3. Генераторная и проверочная матрицы

Также, код Хэмминга имеет минимальное кодовое расстояние в пространстве равное трем, их этого следует, что данный код умеет обнаруживать любые двукратные ошибки и исправлять любые однократные.

Рассмотрим код Хэмминга (7,4) в теории.

Для этого кода генераторная и проверочная матрицы уже заготовлены и имеют следующий вид:

$$G_{4x7} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} H_{3x7} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Рис. 3.4. Структура генераторной и проверочной матрицы для кода (7,4)

Кодирование осуществляется следующим образом:

Кодовое слово ***n*** и информационное слово ***k*** связаны соотношением:

$$\mathrm{n} = \mathrm{v} = \mathrm{k} * \mathrm{G}$$

где *G* — порождающая матрица, структура которой была описана выше.

Например, информационный вектор ***k*** = (1010) отобразится в кодовый вектор следующим образом:

$$v = (1010) \times \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} = (0011010)$$

Рис. 3.5. Кодирование кода (7,4)

Легко заметить, что последние четыре разряда кодового вектора совпадают с информационным вектором. Это свойство называется систематичностью кода.

Здесь первые три бита последовательности являются проверочными (избыточными).

Декодирование осуществляется на получении **синдрома** и производится это все следующим образом:

Система проверочных уравнений выглядит:

$$\mathrm{s} = \mathrm{n} = \mathrm{v} * \mathrm{H}^{\mathrm{T}}$$

Вектор ***s*** принято называть **синдромом**. Таким образом, ошибка будет обнаружена, если хотя бы одна из компонент ***s*** не равна нулю.

При передаче информационного слова ***a*** = (1010) по каналу без шумавыходной вектор был ***n*** = (0011010). Можем убедиться, что в этом случае синдром равен *0*.

$$s = (0011010) \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = (000)$$

Рис. 3.6. Получение синдрома для кода (7,4)

Если, например, в кодовом слове произошла одиночная ошибка на четвертой позиции (***r*** = (0010010)), то синдромом является четвертая строка транспонированной проверочной матрицы.

$$s = (0010010) \circ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = (110)$$

Рис. 3.7. Нахождение ошибочного бита для кода (7,4)

Перебрав все возможные позиции одиночной ошибки, получим полную таблицу синдромов однократных ошибок - таблицу соответствий номера ошибочного разряда получающемуся при этом синдрому. Таким образом, производится декодирование. Если ошибки найдены, то они соответственно исправляются.

Проведение эксперимента и обработка результатов в MATLAB 2015

Задание:

1. Собрать схему
2. Подготовить схемы для реализации кодов Хэмминга (7,4) и (15,11) основываясь на примере, представленном в отчете.

При коде (7,4) в параметрах кодера и декодера MessagelengthK, or M-degreeprimitivepolynomial: устанавливается gfprimfd(3,'min'), а при коде (15,11) gfprimfd(4,'min').

3. Для полученных кодов изменять вероятность ошибки в пределах от 0 до 1 (не менее 4-х точек) и снимать с дисплеев полученные входные последовательности, закодированные последовательности, декодированные последовательности и ошибки.

4. Построить графики зависимости числа обнаруженных ошибок от вероятности ошибки для кодов (7,4) и (15,11).

В рабочем поле необходимо собрать схему для работы кода Хэмминга (7,4). Схема представлена на рисунке 3.8.

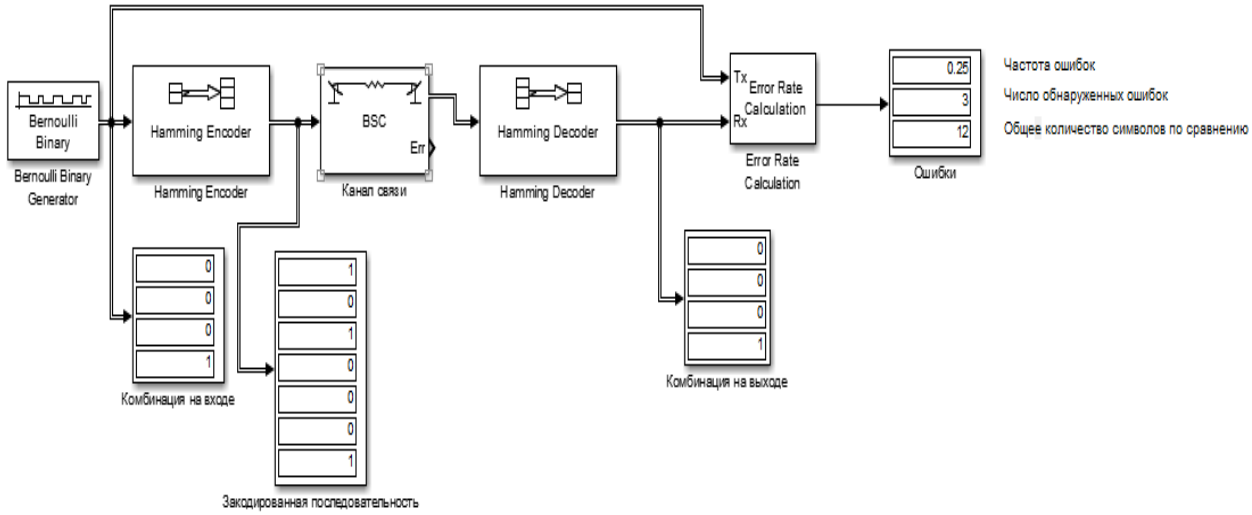


Рис. 3.8. Линия передачи с применением кода Хэмминга

В состав линии с кодированием входят:

1. BernoulliBinaryGenerator
2. HammingEncoder
3. Binary Symmetric Channel (каналпередачи)
4. HammingDecoder
5. Error RateCalculation (анализаторошибок)
6. Display

Устанавливаем характеристики блоков для кода (7,4)

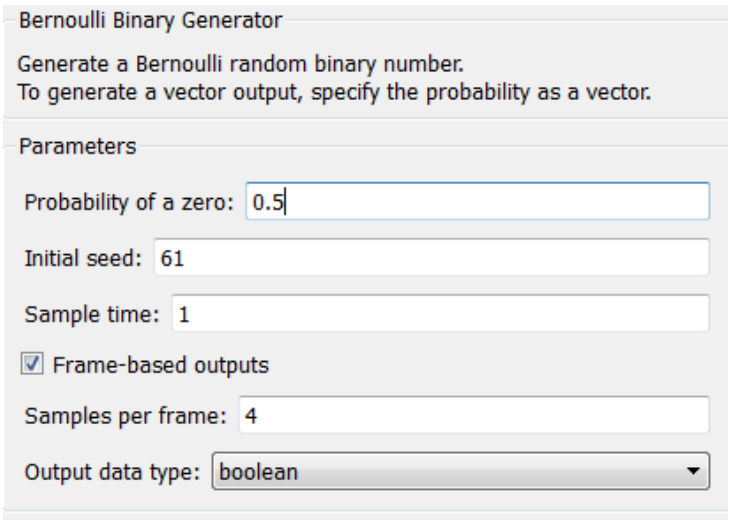


Рис. 3.9. Параметры Bernoulli Binary Generator

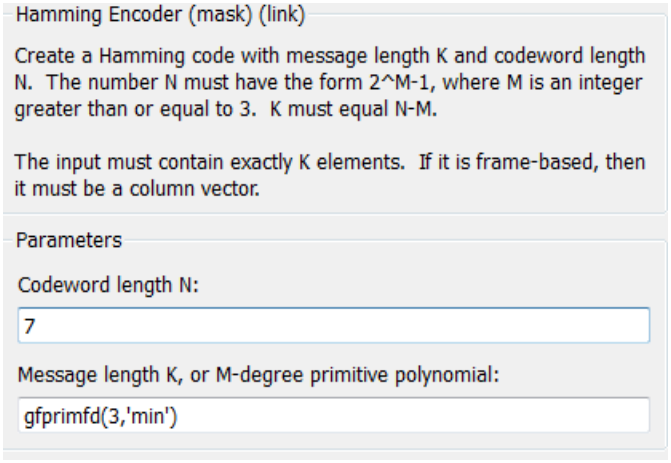


Рис. 3.10. Параметры Hamming Encoder

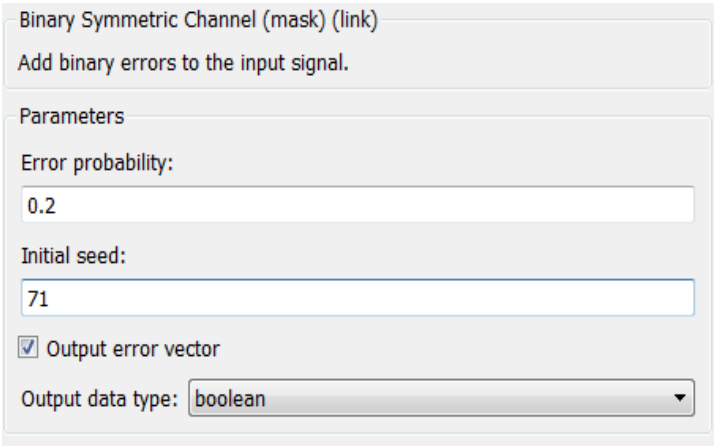


Рис. 3.11. Параметры Binary Symmetric Channel

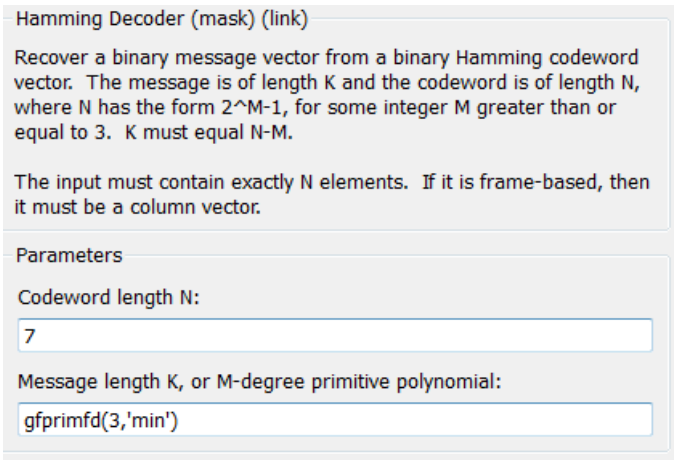


Рис. 3.12. Параметры Hamming Decoder

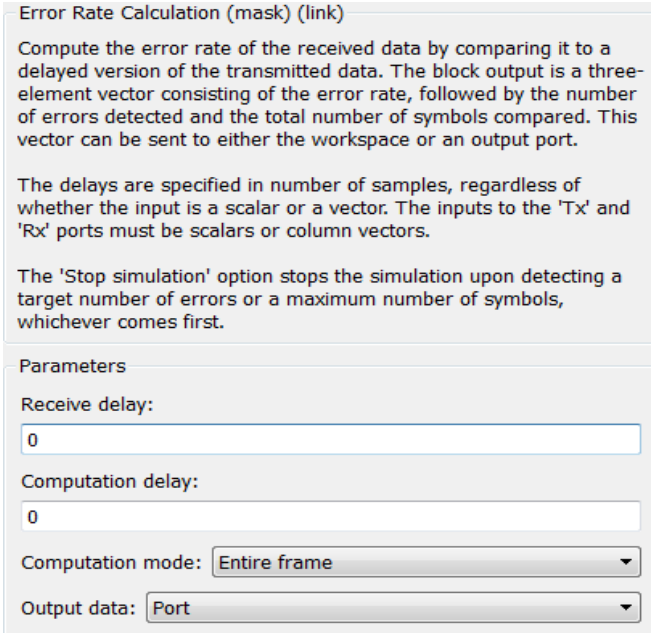


Рис. 3.13 Параметры Error Rate Calculation

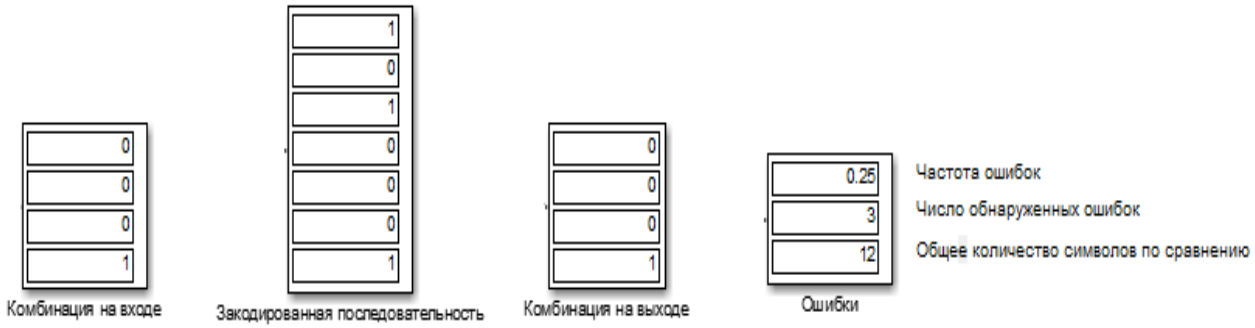


Рис. 3.14. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,2)

Анализируя рисунок выше, можно сделать вывод, что комбинация на входе совпадает с комбинацией на выходе, таким образом, передача осуществилась удачно. Что качается

ошибок, то их частота равна 0,25, число обнаруженных ошибок равно 3, общее количество символов по сравнению равно 12. Кодирование и декодирование здесь осуществляется методом описанном в выше.

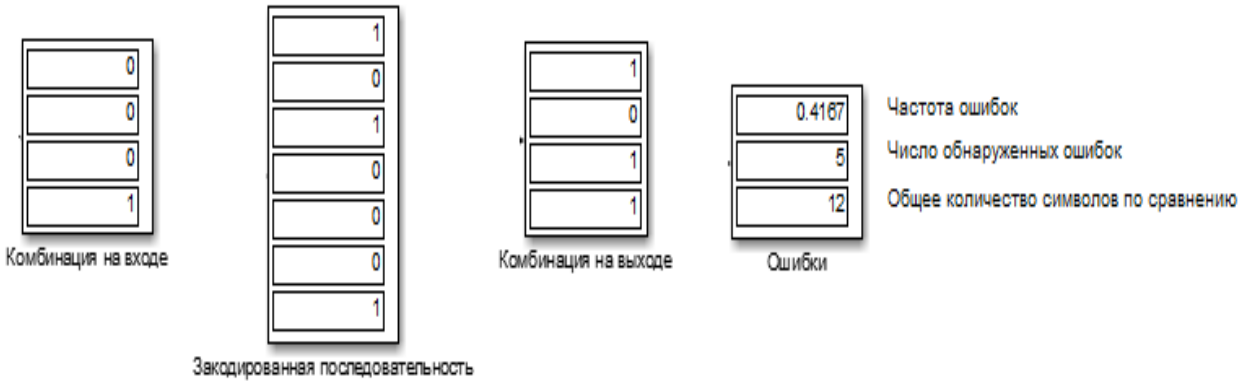


Рис. 3.15. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,4)

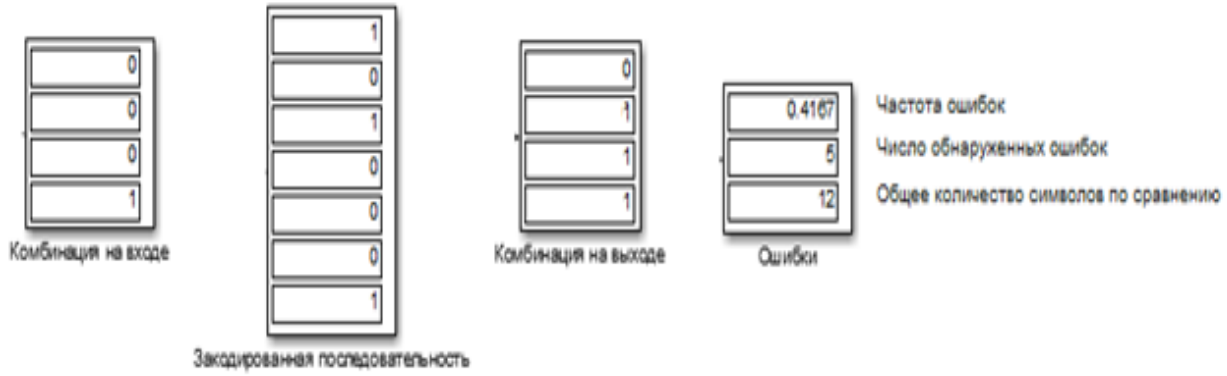


Рис. 3.16. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, ошибки (вероятность ошибок равна 0,6)

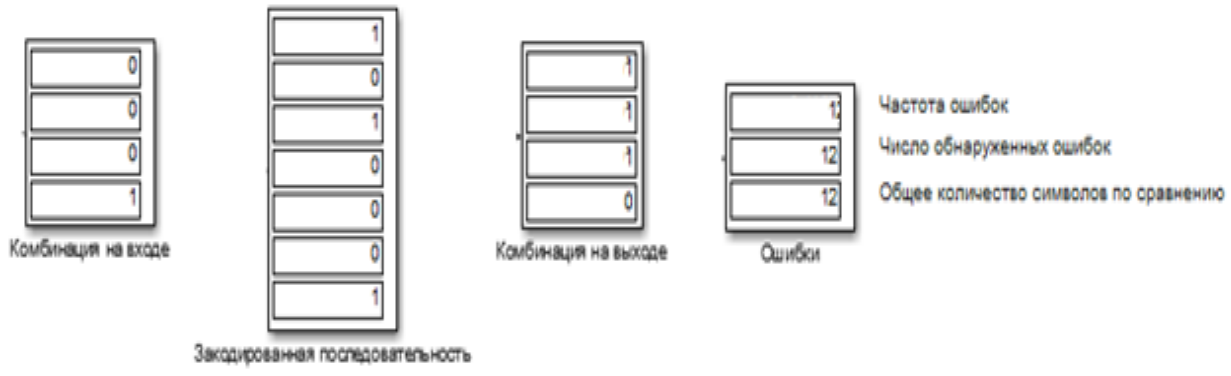


Рис. 3.17. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,8)

Рис. 3.18. График зависимости числа ошибок (OY) от вероятности ошибки (OX) для кода Хэмминга (7,4)

Устанавливаем характеристики блоков для кода (15,11)

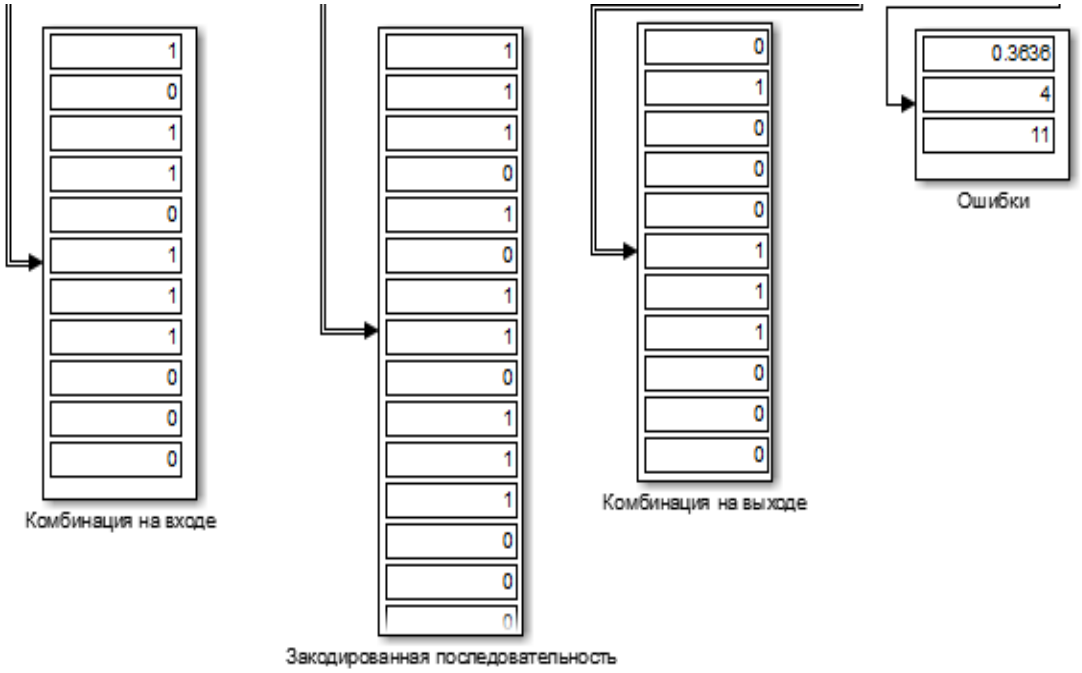


Рис. 3.19. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,2)

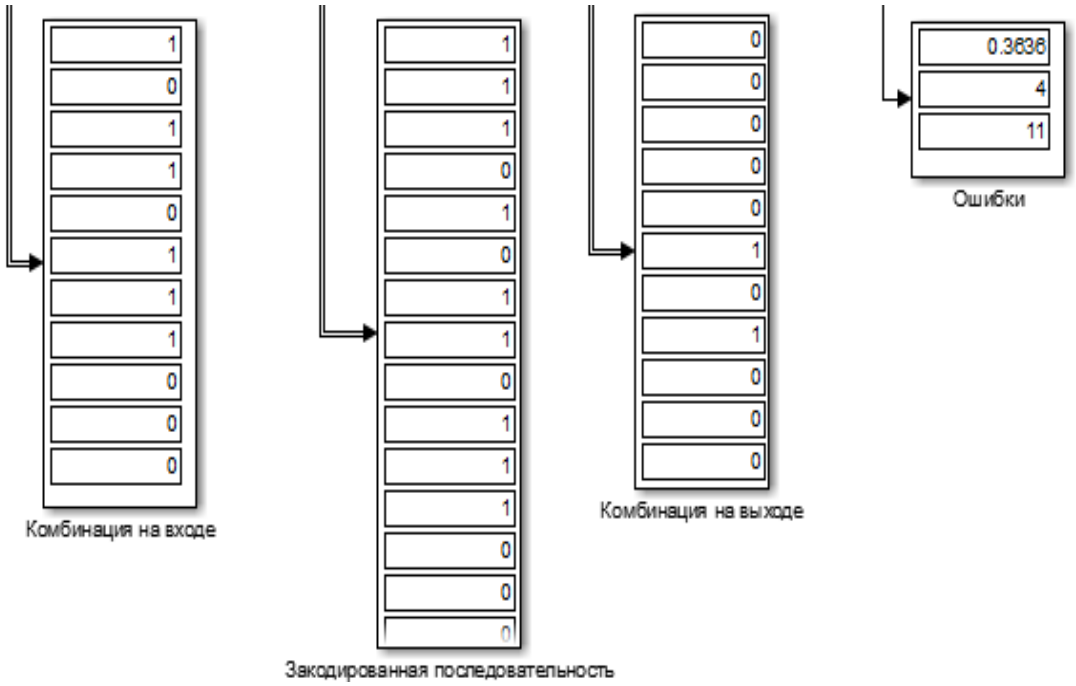


Рис. 3.20. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,4)

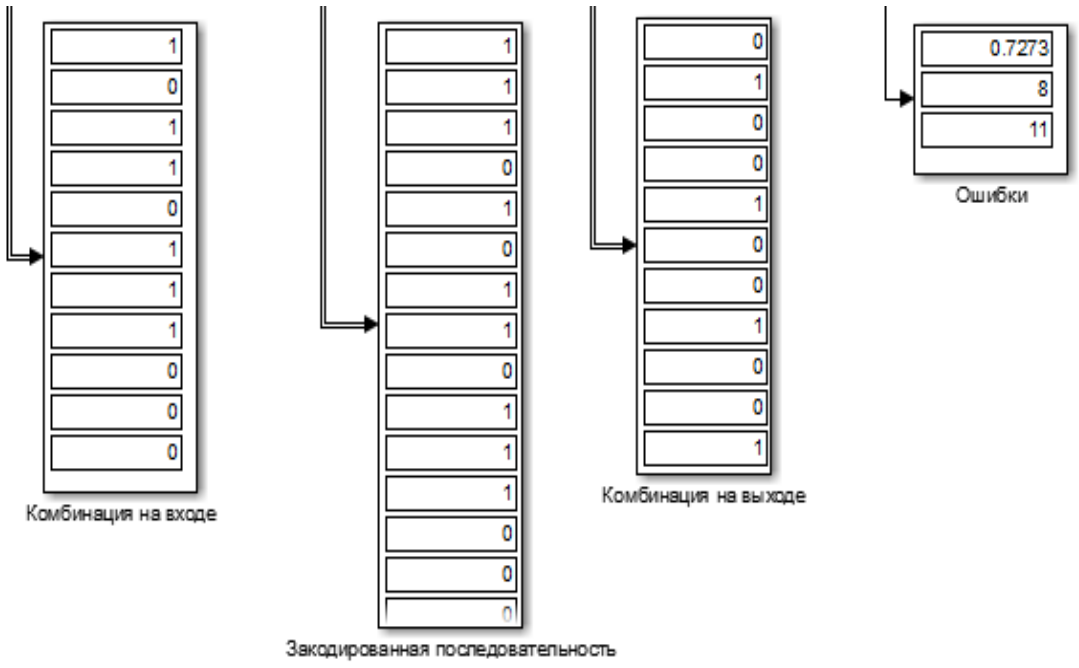


Рис. 3.21. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,6)

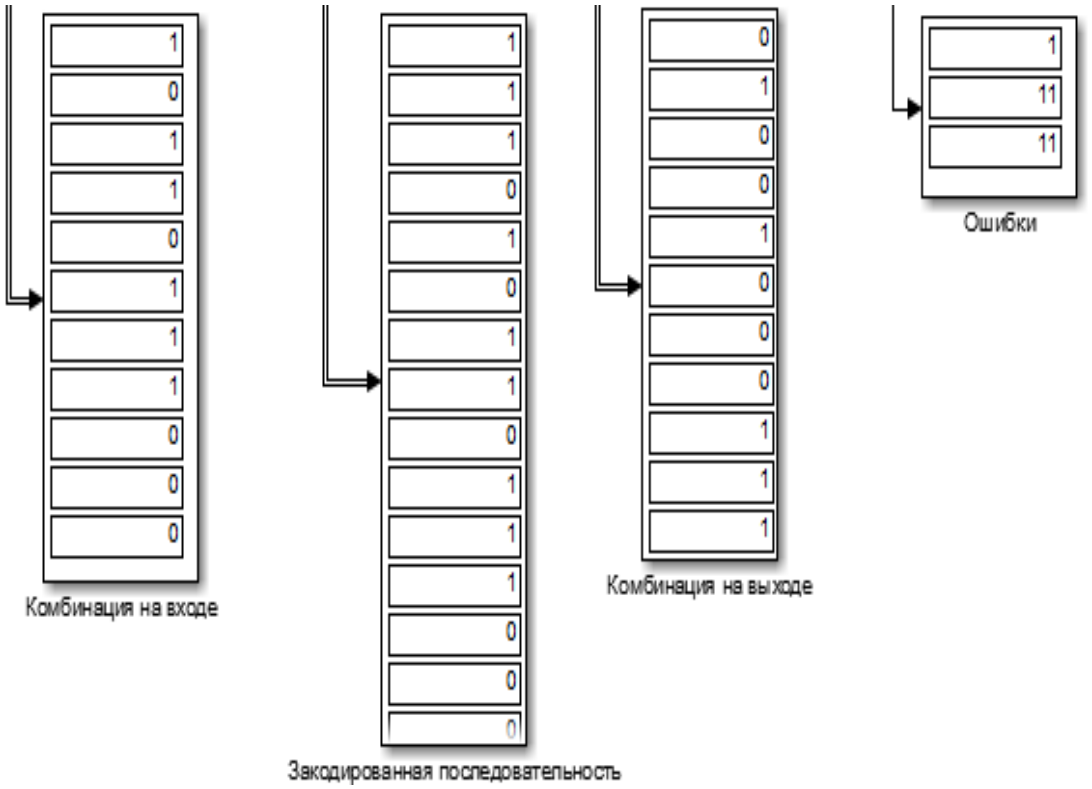


Рис. 3.22. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,8)

Рис. 3.23. График зависимости числа ошибок (OY) от вероятности ошибки (OX) для кода Хэмминга (15,11)

В результате проверки построена схема линии передачи с кодированием Хэмминга в среде Simulink. Построены графики зависимостей числа ошибок на выходе декодера от вероятности ошибки в канале связи для кодов (7,4) и (15,11).

Из графиков (рисунок 3.18 и рисунок 3.23) видно, что число ошибок увеличивается с ростом вероятности ошибок.

Код БЧХ (Боуза-Чоудхури-Хоквенгема)

Код БЧХ является циклическим кодом. С циклическим кодом Хэмминга у них много чего общего, а именно алгоритм кодирования, который отличается только нахождением генераторного полинома, а процесс декодирования полностью схож.Необходимо начать с небольшого введения в код БЧХ.*Многочлен* g(x) степени называется *примитивным*, если $x^j + 1$ делится на g(x) без остатка для $j = 2^k - 1$ и не делится ни для какого меньшего значения (где $\boldsymbol{k}$ – количество информационных бит).Например, *многочлен* $1 + x^2 + x^3$ примитивен: он делит $x^7 + 1$, но не делит $x^j + 1$ при $j < 7$. Примитивен также *многочлен* $1 + x^3 + x^4$ - он делит $x^{15} + 1$, но не делит $x^j + 1$ при $j < 15$(для кода (7,15)).

Кодирующий *многочлен* g(x) для БЧХ-кода, *длина* кодовых слов которого $\boldsymbol{n}$, строится так. Находится примитивный *многочлен* минимальной степени такой, что$n \le 2^q - 1$ или$q \ge \log_2(n+1)$.Пусть $\boldsymbol{\alpha}$ -корень этого многочлена, тогда рассмотрим кодирующий *многочлен* $g(x) = НОК(m_1(x), \dots, m_{d-1}(x))$,где $m_1(x), \dots, m_{d-1}(x)$ - многочлены минимальной степени, имеющие корнями соответственно$\alpha, \alpha^2, \dots, \alpha^{d-1}$.

Построенный кодирующий *многочлен* производит код с минимальным расстоянием между кодовыми словами, не меньшим$\boldsymbol{d}$, и длиной кодовых слов $\boldsymbol{n}$.

Кодирование:

Например, нужно построить БЧХ-код с длиной кодовых слов $\boldsymbol{n=15}$и минимальным расстоянием между кодовыми словами $\boldsymbol{d=5}$. Степень примитивного многочлена равна$q = \log_2(n+1) = 4$ и сам он равен $1 + x^3 + x^4$(примитивный многочлен 4-ой степени для кода (7,15)). Пусть $\boldsymbol{\alpha}$ - его корень, тогда α^2 и α^4 - также его корни. Минимальным многочленом для α^3 будет $1 + x + x^2 + x^3 + x^4$. Следовательно,

$$g(x) = НОК(1 + x^3 + x^4, 1 + x + x^2 + x^3 + x^4) = (1 + x^3 + x^4)(1 + x + x^2 + x^3 + x^4)$$
$$= 1 + x + x^2 + x^4 + x^8$$

Степень полученного многочлена равна 8, построенный БЧХ-код будет (7,15) кодом. Слово 1000100 или $a(x) = x^4 + 1$будетзакодировано кодовым словом $a(x)g(x) = x^{12} + x^6 + x^5 + x^2 + x + 1$ или 111001100000100. На практике будет рассмотрен код БЧХ (15,7) и БЧХ (15,11).

Декодирование:

Декодирование производится путем деления закодированной последовательности на генераторный полином, который использовался при кодировании. Полученная последовательность при делении и будет декодированной последовательностью. В лучшем случае, остатка при делении не будет, это значит, что ошибок не выявлено.

Если же остаток есть, называется он не иначе как ***синдром,*** в таком случае ошибки присутствуют в закодированной последовательности. В этом случае поступают так. Закодированную последовательность складывают с вектором ошибок по модулю два, и исправляют ошибочный бит. Вектор ошибок формируется с помощью специальных схем, которые анализируя закодированную последовательность, формируют данный вектор.

Еще одним способ исправления ошибок является следующий метод. На основании полученного генераторного полинома строится схема и на каждом такте ее работы определяется ***синдром.*** В данном случае необходимо получить два синдрома, один из них это остаток, от деления полученный при декодировании, а второй это комбинация 100. Далее производится вычитание номера такта комбинации 100 и номера такта остатка. Полученная разность и является номером бита в закодированной комбинации.

Проведение эксперимента и обработка результатов

Задание:

1. Собрать схему
2. Подготовить схемы для реализации кодов БЧХ (15,7) и (15,11) основываясь на примере, представленном в отчете.
3. Для полученных кодов изменять вероятность ошибки в пределах от 0 до 1 (не менее 4-х точек) и снимать с дисплеев полученные входные последовательности, закодированные последовательности, декодированные последовательности и ошибки.
4. Построить графики зависимости числа обнаруженных ошибок от вероятности ошибки для кодов (15,7) и (15,11).
5. Все поэтапное исследование представить в отчете.

В рабочем поле необходимо собрать схему для работы кода БЧХ (15,7). Схема представлена на рисунке 3.24.

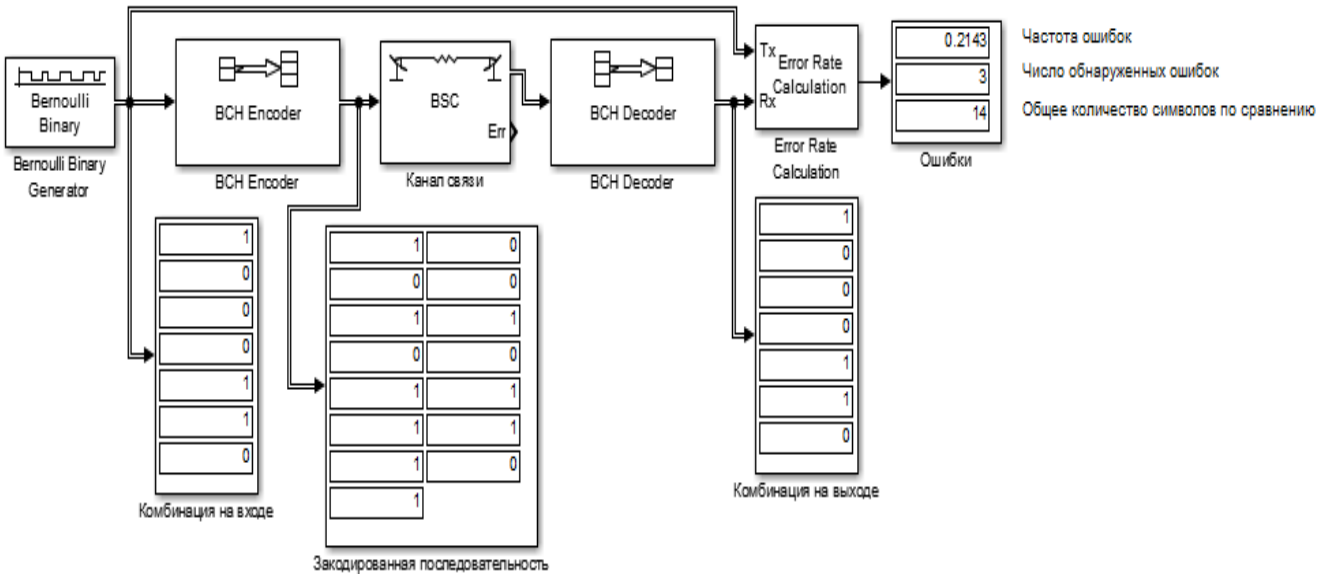


Рис. 3.24. Линия передачи с применением кода БЧХ (15,7)

В состав линии с кодированием входят:

1. BernoulliBinaryGenerator
2. BCHEncoder
3. Binary Symmetric Channel (каналпередачи)
4. BCHDecoder
5. Error RateCalculation (анализаторошибок)
6. Display

Устанавливаем характеристики блоков для кода (15,7)

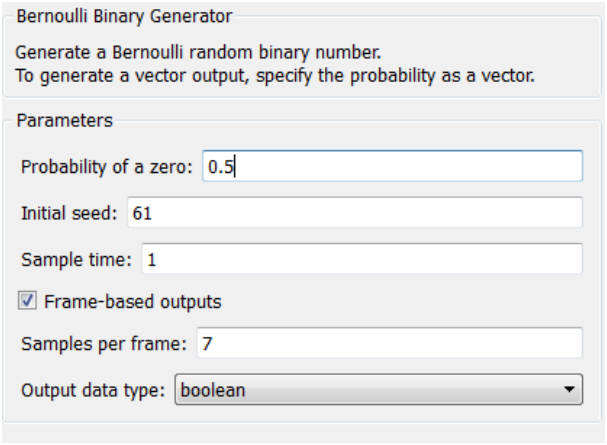


Рис. 3.25. Параметры Bernoulli Binary Generator

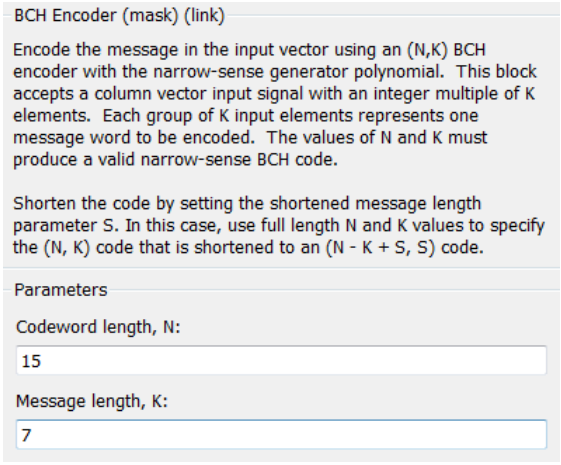


Рис. 3.26. Параметры BCH Encoder

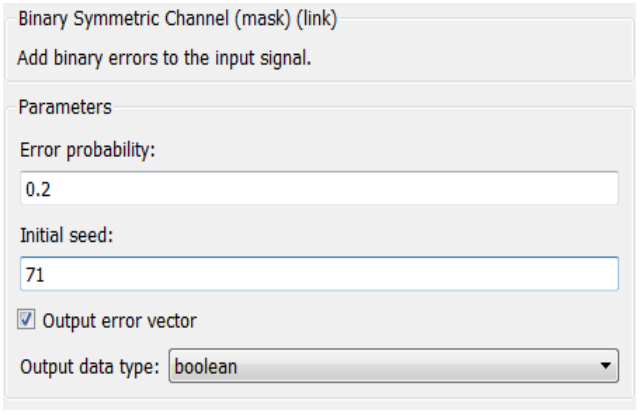


Рис. 3.27. Параметры Binary Symmetric Channel

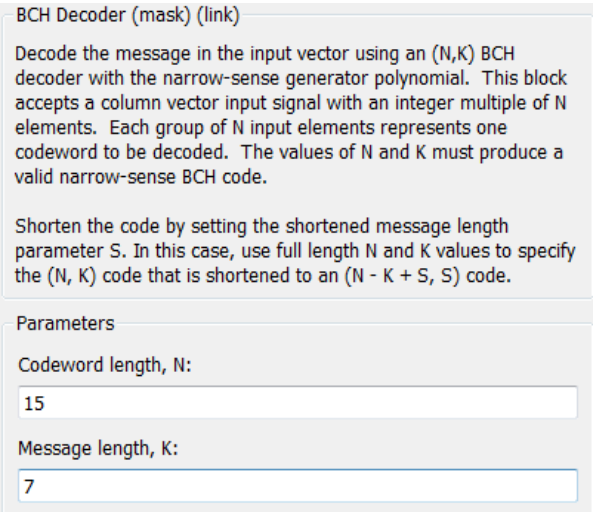


Рис. 3.28 Параметры BCH Decoder

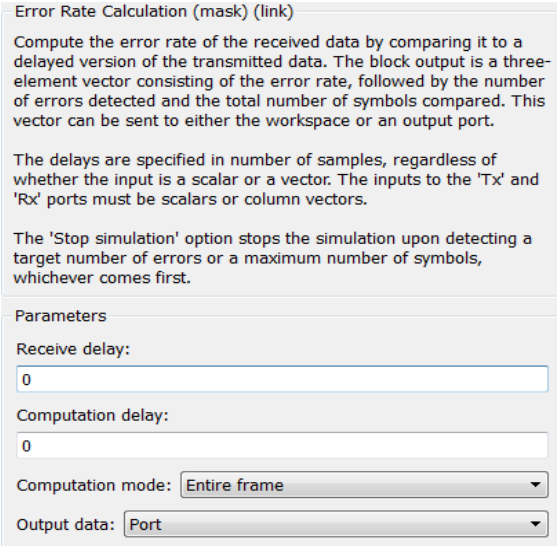


Рис. 3.29. Параметры Error Rate Calculation

Представим полученные результаты

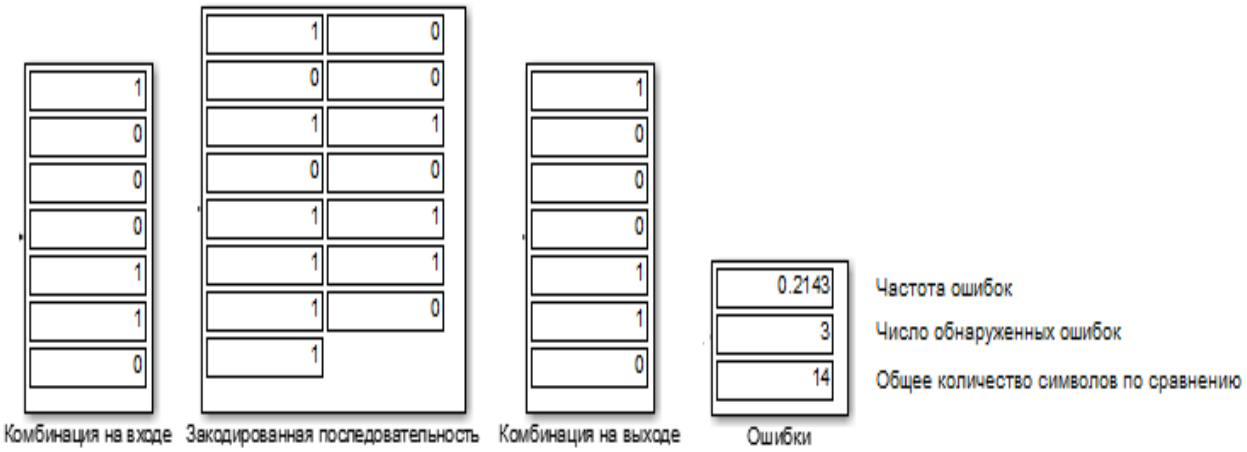


Рис. 3.30. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,2)

Можно сделать вывод, что комбинация на входе совпадает с комбинацией на выходе, таким образом, передача осуществилась удачно. Что качается ошибок, то их частота равна 0,2143, число обнаруженных ошибок равно 3, общее количество символов по сравнению равно 14. Кодирование и декодирование здесь осуществляется методом описанном выше.

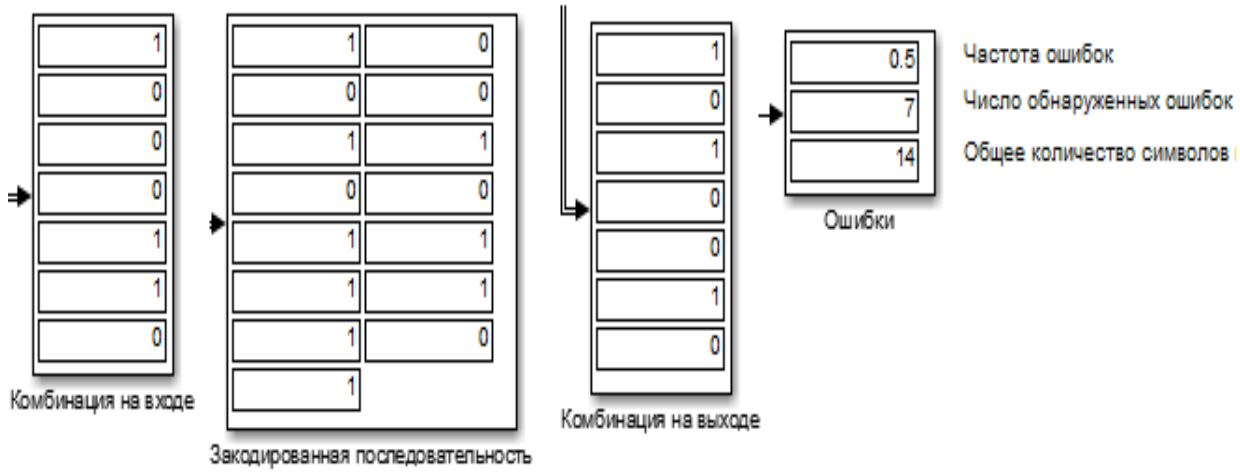


Рис. 3.31. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,4)

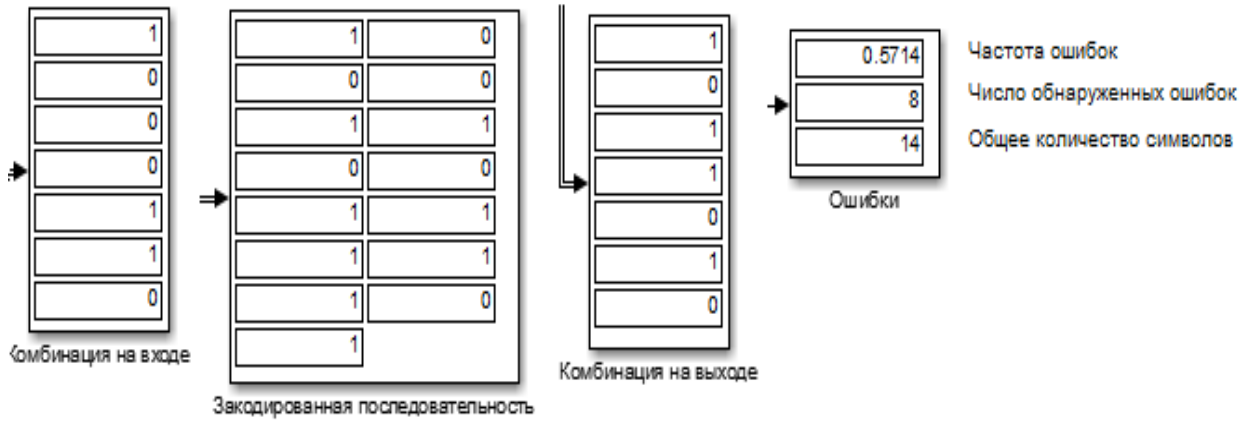


Рис. 3.32. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,6)

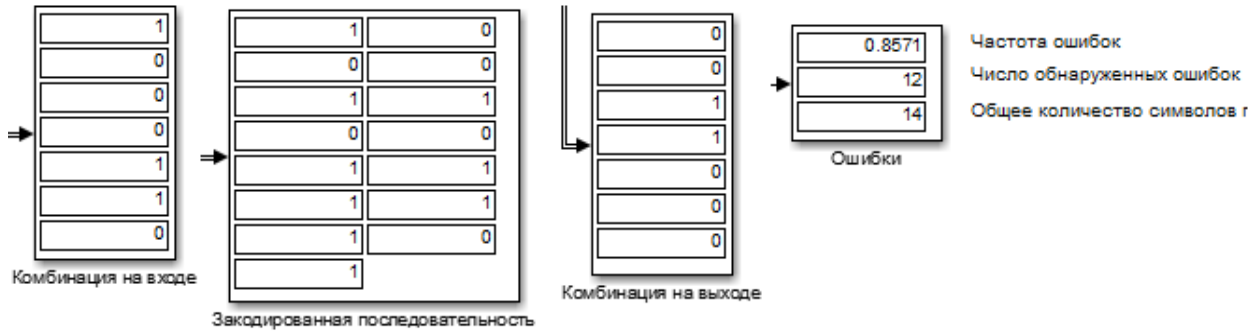


Рис. 3.33. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,8)

Устанавливаем характеристики блоков для кода (15,11)

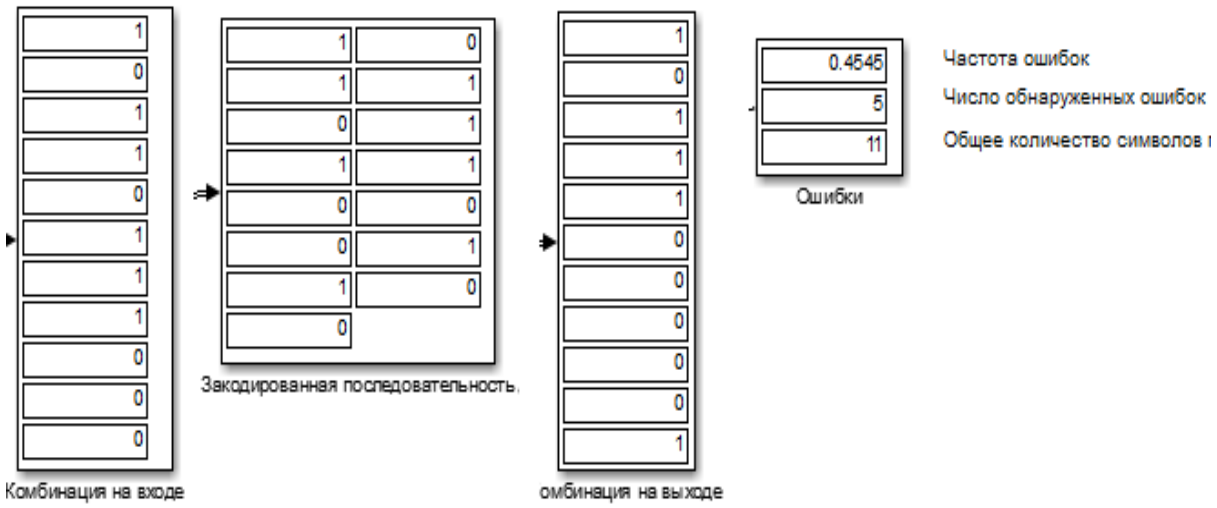


Рис. 3.34. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,2)

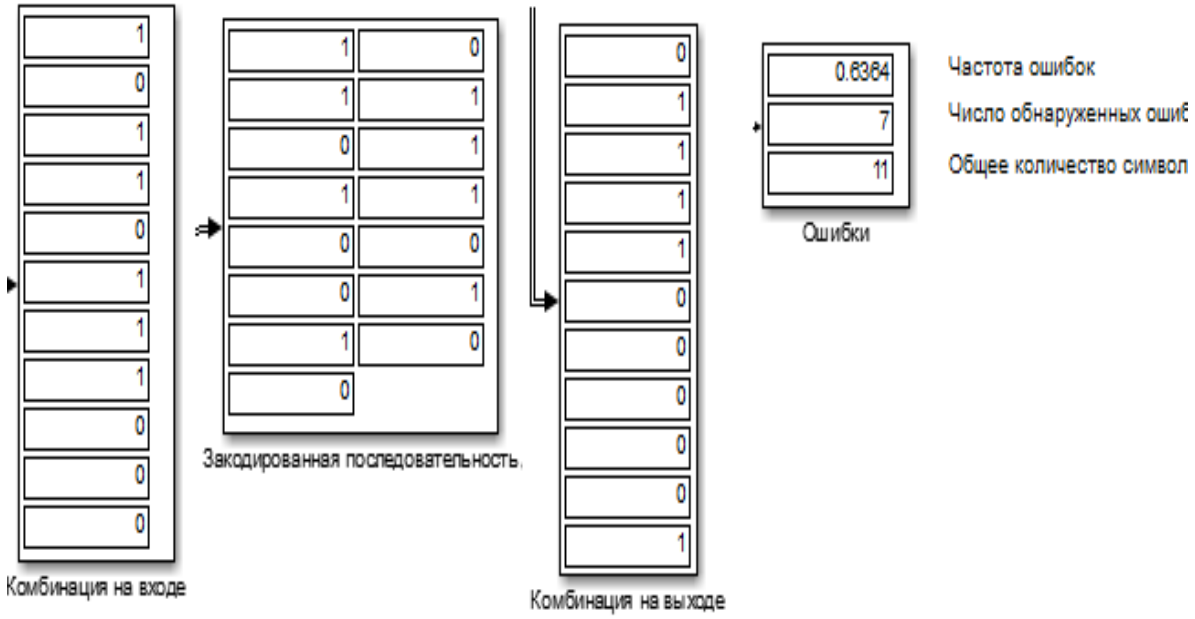


Рис. 3.35. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,4)

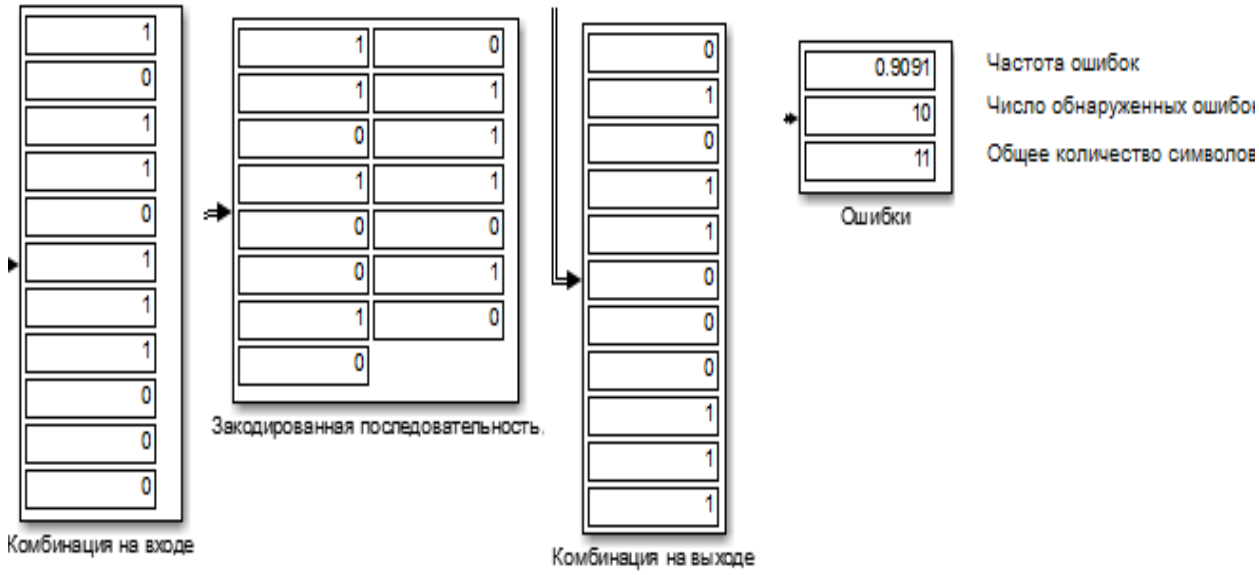


Рис. 3.36 Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,6)

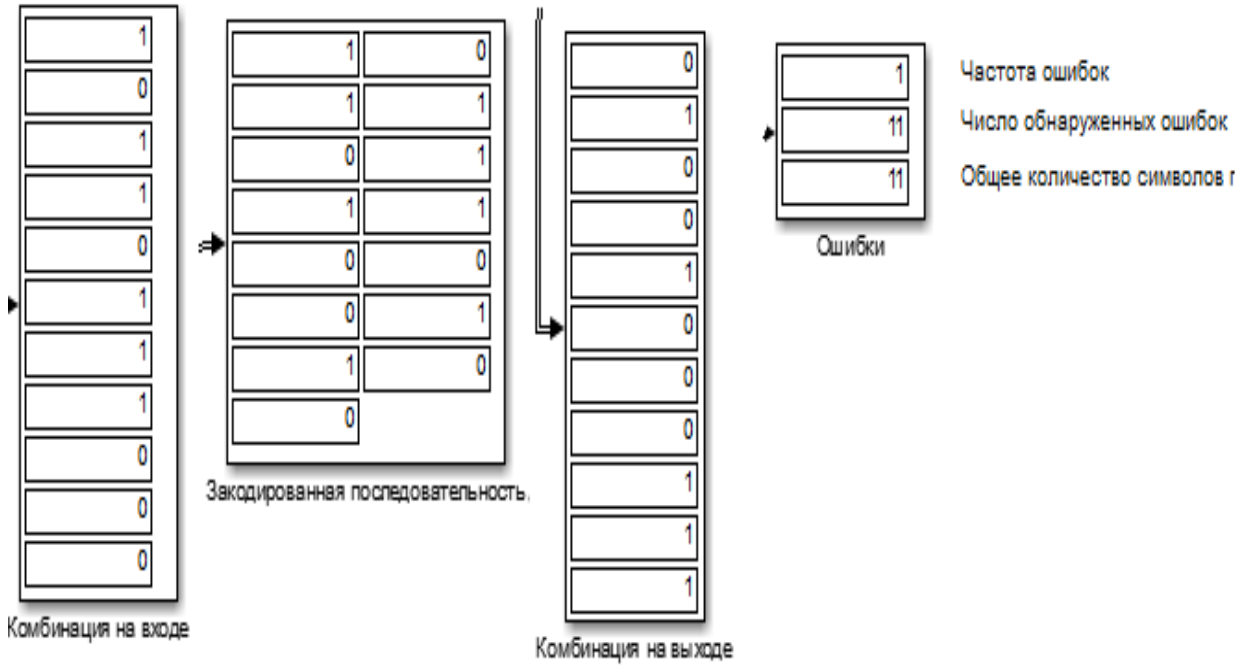


Рис. 3.37. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,8)

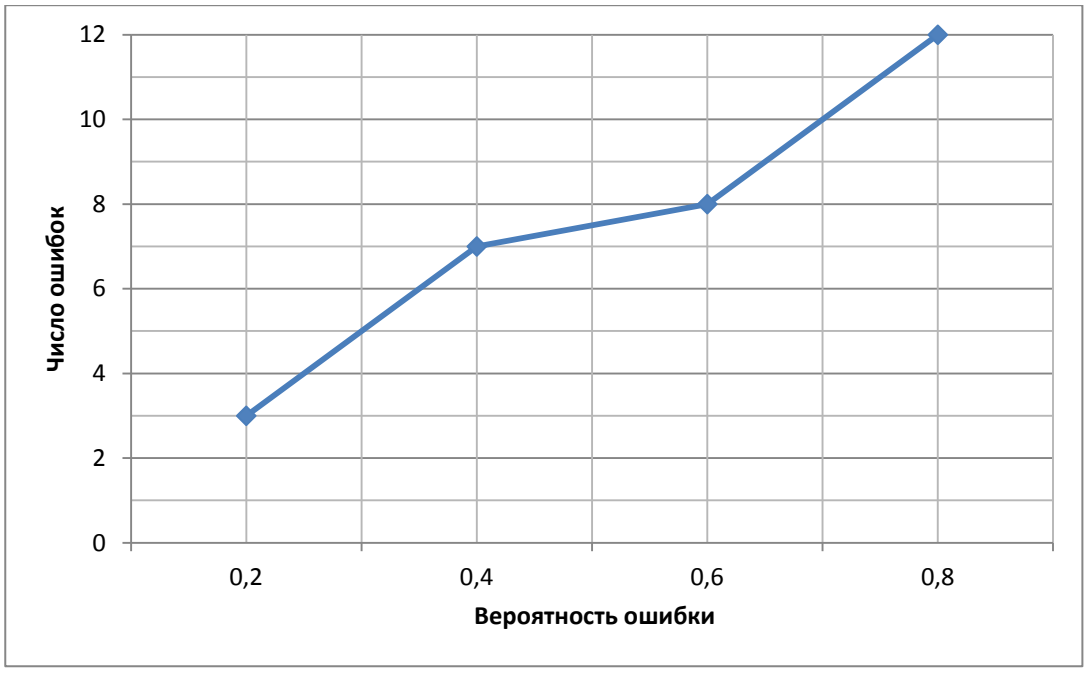


Рис. 3.38. График зависимости числа ошибок от вероятности ошибки для кода БЧХ (15,7)

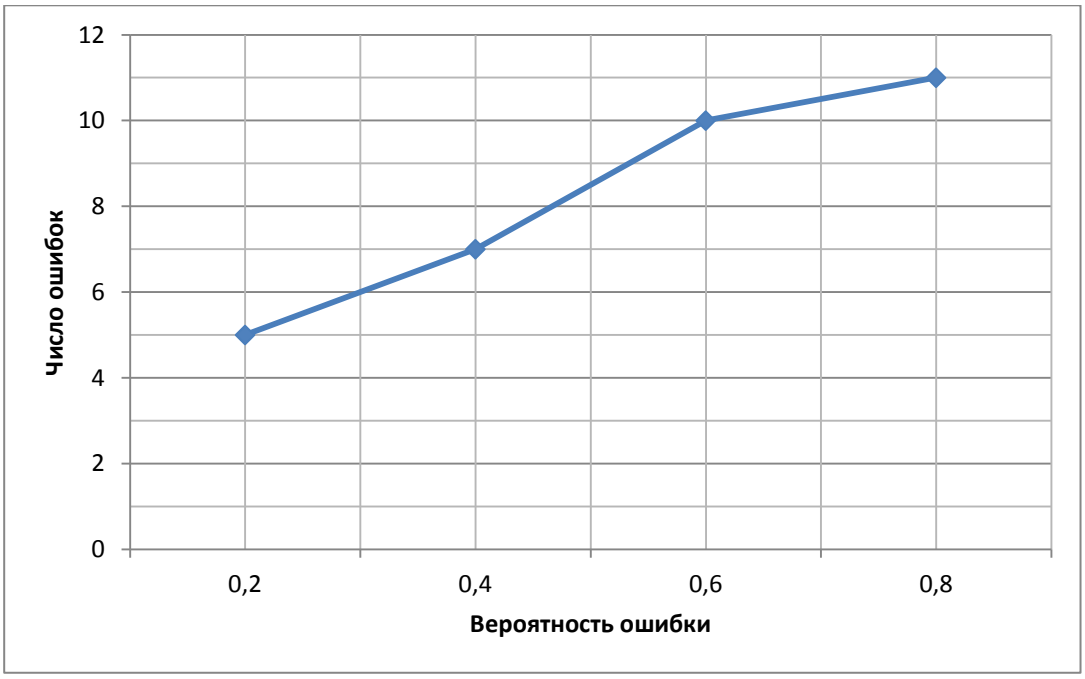


Рис. 3.39. График зависимости числа ошибок от вероятности ошибки для кода БЧХ (15,11)

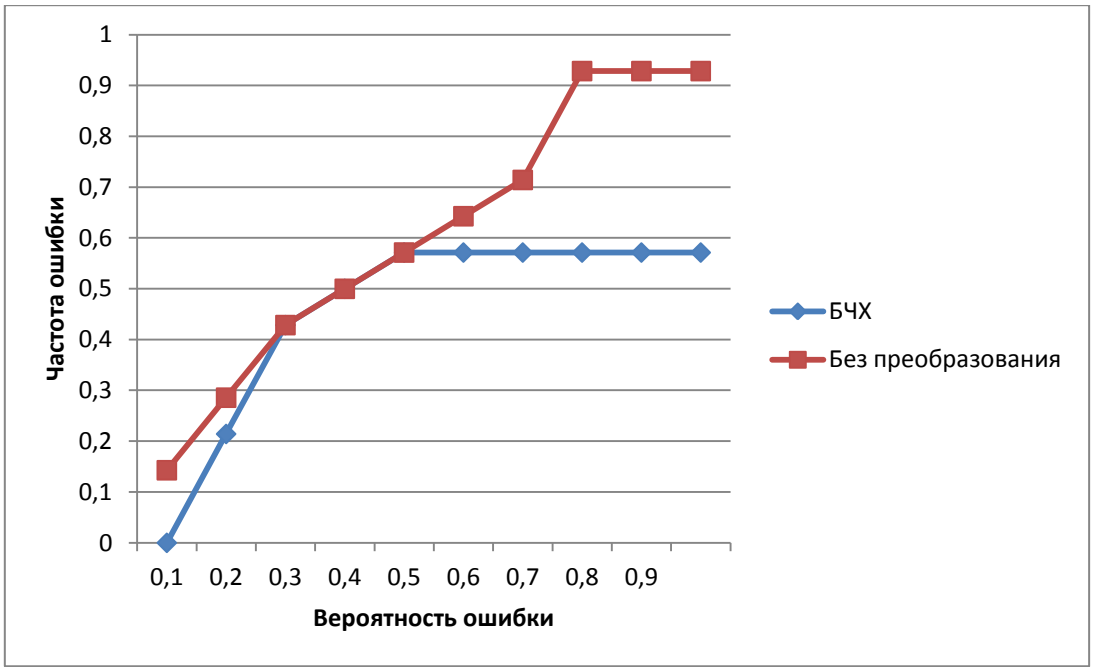


Рис. 3.40а. Зависимость вероятности ошибки от частоты ошибки

В результате практической работы построена схема линии передачи с кодированием БЧХ в среде Simulink. Построены графики зависимостей числа ошибок на выходе декодера от вероятности ошибки в канале связи для кодов (15,7) и (15,11). Из графиков (рисунок 3.38 и рисунок 3.39) видно, что число ошибок увеличивается с ростом вероятности ошибок. При этом код (15,7) оказался лучше по способности обнаружения ошибок.

Код Рида-Соломона

Кодировщик Рида-Соломона берет блок цифровых данных и добавляет дополнительные "избыточные" биты. Ошибки происходят при передаче по каналам связи или по разным причинам при запоминании (например, из-за шума или наводок, царапин на CD и т.д.). Декодер Рида-Соломона обрабатывает каждый блок, пытается исправить ошибки и восстановить исходные данные. Число и типы ошибок, которые могут быть исправлены, зависят от характеристик кода Рида-Соломона.

Свойства кодов Рида-Соломона

Коды Рида-Соломона являются субнабором кодов БЧХ и представляют собой линейные блочные коды. Код Рида-Соломона специфицируются как RS (n,k) ***s*** - битных символов.

Это означает, что кодировщик воспринимает ***k*** информационных символов по s битов каждый и добавляет символы четности для формирования ***n*** символьного кодового слова. Декодер Рида-Соломона может корректировать до ***t*** символов, которые содержат ошибки в кодовом слове, где $2t = n–k$.

Диаграмма, представленная ниже, показывает типовое кодовое слово Рида-Соломона:

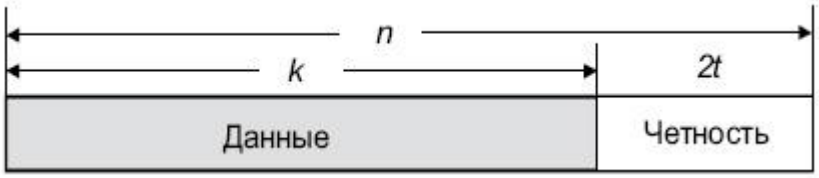


Рис. 3.40. Кодовое слово для кода Р-С

Популярным кодом Рида-Соломона является RS (255, 223) с 8-битными символами. Каждое кодовое слово содержит 255 байт, из которых 223 являются информационными и 32 байтами четности. Для этого кода ***n = 255, k = 223, s = 8, 2t = 32, t = 16***.

Декодер может исправить любые 16 символов с ошибками в кодовом слове: то есть ошибки могут быть исправлены, если число искаженных байт не превышает 16.

При размере символа ***s***, максимальная длина кодового слова ***n*** для кода Рида-Соломона равна $n = 2^s - 1$.

Например, максимальная длина кода с 8-битными символами (s=8) равна 255 байтам.

Коды Рида-Соломона могут быть в принципе укорочены путем обнуления некоторого числа информационных символов на входе кодировщика (передавать их в этом случае не нужно). При передаче данных декодеру эти нули снова вводятся в массив.

Код (255, 223), описанный выше, может быть укорочен до (200, 168). Кодировщик будет работать с блоком данных 168 байт, добавит 55 нулевых байт, сформирует кодовое слово (255, 223) и передаст только 168 информационных байт и 32 байта четности.

Объем вычислительной мощности, необходимой для кодирования и декодирования кодов Рида-Соломона, зависит от числа символов четности. Большое значение t означает, что большее число ошибок может быть исправлено, но это потребует большей вычислительной мощности по сравнению с вариантом при меньшем ***t***.

Ошибки в символах

Одна ошибка в символе происходит, когда 1 бит символа оказывается неверным или когда все биты неверны.

Код RS(255,223) может исправить до 16 ошибок в символах. В худшем случае, могут иметь место 16 битовых ошибок в разных символах (байтах). В лучшем случае, корректируются 16 полностью неверных байт, при этом исправляется 16 * 8 = 128 битовых ошибок.

Коды Рида-Соломона особенно хорошо подходят для корректировки кластеров ошибок (когда неверными оказываются большие группы бит кодового слова, следующие подряд).

Декодирование

Алгебраические процедуры декодирования Рида-Соломона могут исправлять ошибки и потери. Потерей считается случай, когда положение неверного символа известно. Декодер

может исправить до ***t*** ошибок или до ***2t*** потерь. Данные о потере (стирании) могут быть получены от демодулятора цифровой коммуникационной системы, т.е. демодулятор помечает полученные символы, которые вероятно содержат ошибки.

Когда кодовое слово декодируется, возможны три варианта.

1. Если ***2s + r < 2t*** (s ошибок, r потерь), тогда исходное переданное кодовое слово всегда будет восстановлено. В противном случае:
2. Декодер детектирует ситуацию, когда он не может восстановить исходное кодовое слово. Или же:
3. Декодер некорректно декодирует и неверно восстановит кодовое слово без какого-либо указания на этот факт.

Вероятность каждого из этих вариантов зависит от типа используемого кода Рида-Соломона, а также от числа и распределения ошибок.

Архитектура кодирования и декодирования кодов Рида-Соломона

Кодирование и декодирование Рида-Соломона может быть выполнено аппаратно или программно.

Арифметика конечного поля Галуа

Коды Рида-Соломона базируются на специальном разделе математики – полях Галуа (GF) или конечных полях. Арифметические действия (+,-, x, / и т.д.) над элементами конечного поля дают результат, который также является элементом этого поля. Кодировщик или декодер Рида-Соломона должны уметь выполнять эти арифметические операции. Эти операции для своей реализации требуют специального оборудования или специализированного программного обеспечения.

Образующий полином

Кодовое слово Рида-Соломона формируется с привлечением специального полинома. Все корректные кодовые слова должны делиться без остатка на эти образующие полиномы. Общая форма образующего полинома имеет вид

$$g(x) = (x - a^{i})(x - a^{i+1})\dots(x - a^{i+1+2t})$$

а кодовое слово формируется с помощью операции

$$c(x) = g(x) * i(x)$$

где g(x) является образующим полиномом, i(x) представляет собой информационный блок, c(x) – кодовое слово, называемое простым элементом поля.

Архитектура кодировщика

2t символов четности в кодовом слове Рида-Соломона определяются из следующего соотношения:

$$p(x) = i(x) * x^{n-k} \bmod g(x)$$

Ниже показана схема реализации кодировщика для версии RS(255,249):

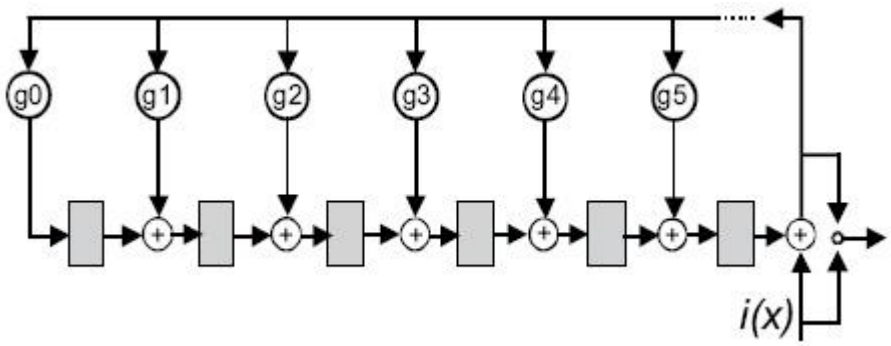


Рис. 3. 41. Схема кодировщика Р-С

Каждый из 6 регистров содержит в себе символ (8 бит). Арифметические операторы выполняют сложение или умножение на символ как на элемент конечного поля.

Архитектура декодера

Общая схема декодирования кодов Рида-Соломона показана ниже на рис. 3.42.

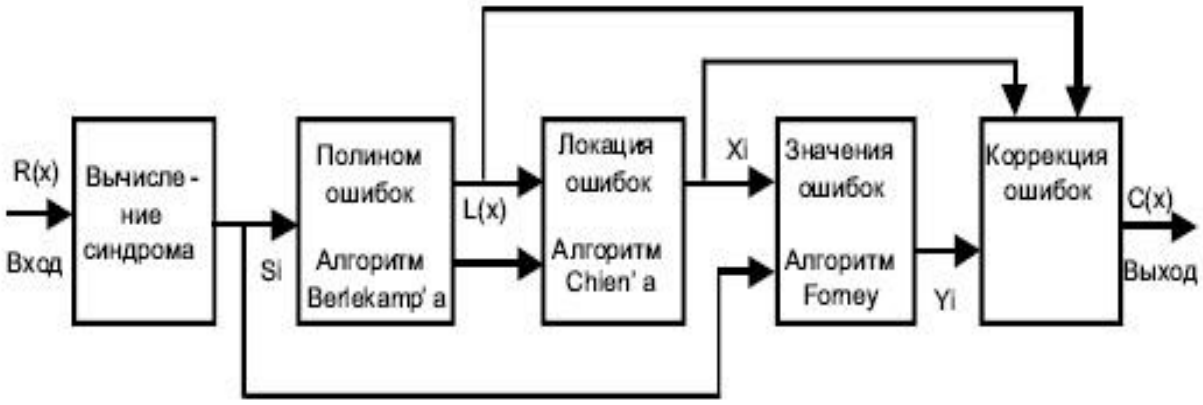


Рис. 3.42. Схема работы с кодами Рида-Соломона

Обозначения:

- r(x) – Полученное кодовое слово
- Si – Синдромы
- L(x) – Полином локации ошибок
- Xi – Положения ошибок
- Yi – Значения ошибок
- c(x) – Восстановленное кодовое слово
- v – Число ошибок

Полученное кодовое слово r(x) представляет собой исходное (переданное) кодовое слово c(x) плюс ошибки:

$r(x) = c(x) + e(x)$

Декодер Рида-Соломона пытается определить позицию и значение ошибки для t ошибок (или ***2t*** потерь) и исправить ошибки и потери.

Вычисление синдрома

Вычисление синдрома похоже на вычисление четности. Кодовое слово Рида-Соломона имеет 2t синдромов, это зависит только от ошибок (а не передаваемых кодовых слов). Синдромы могут быть вычислены путем подстановки ***2t*** корней образующего полинома g(x) в r(x).

Нахождение позиций символьных ошибок

Это делается путем решения системы уравнений с t неизвестными. Существует несколько быстрых алгоритмов для решения этой задачи. Эти алгоритмы используют особенности структуры матрицы кодов Рида-Соломона и сильно сокращают необходимую вычислительную мощность. Делается это в два этапа.

1. Определение полинома локации ошибок.

Это может быть сделано с помощью алгоритма Berlekamp-Massey или алгоритма Эвклида. Алгоритм Эвклида используется чаще на практике, так как его легче реализовать, однако алгоритм Berlekamp-Massey позволяет получить более эффективную реализацию оборудования и программ.

2. Нахождение корней этого полинома. Это делается с привлечением алгоритма поиска Chien.

Нахождение значений символьных ошибок

Здесь также нужно решить систему уравнений с t неизвестными. Для решения используется быстрый алгоритм Forney.

Реализация кодировщика и декодера Рида-Соломона. Аппаратная реализация

Существует несколько коммерческих аппаратных реализаций. Имеется много разработанных интегральных схем, предназначенных для кодирования и декодирования кодов Рида-Соломона. Эти ИС допускают определенный уровень программирования (например, RS (255, k), где t может принимать значения от 1 до 16).

Программная реализация

До недавнего времени программные реализации в "реальном времени" требовали слишком большой вычислительной мощности практически для всех кодов Рида-Соломона. Главной трудностью в программной реализации кодов Рида-Соломона являлось то, что процессоры общего назначения не поддерживают арифметические операции для поля Галуа. Однако оптимальное составление программ в сочетании с возросшей вычислительной мощностью позволяют получить вполне приемлемые результаты для относительно высоких скоростей передачи данных.

Практическая часть исследования

На данном этапе необходимо реализовать и проанализировать описанные выше алгоритмы кодирования на практике в программной среде Matlab 15. Необходимо построить соответствующие схемы, которые будут в себя включать источник сообщения, кодер, декодер, канал передачи, анализатор ошибок и соответственно устройства визуализации полученных результатов. Так же необходимо сравнить ошибки передачи с линией, которая не будет никак преобразовывать передаваемую информацию, а просто передавать ее по каналу связи и при увеличении вероятности ошибки в канале до 0,3. Коды, которые подлежать реализации это: код Хэмминга (7,4), код БЧХ (15,7), и код Рида-Соломона (255,223). Всю задачу выполнить в среде Matlab 15. Представить проделанную работу в отчете.

Для начала необходимо запустить саму программу. Происходит это следующим образом:

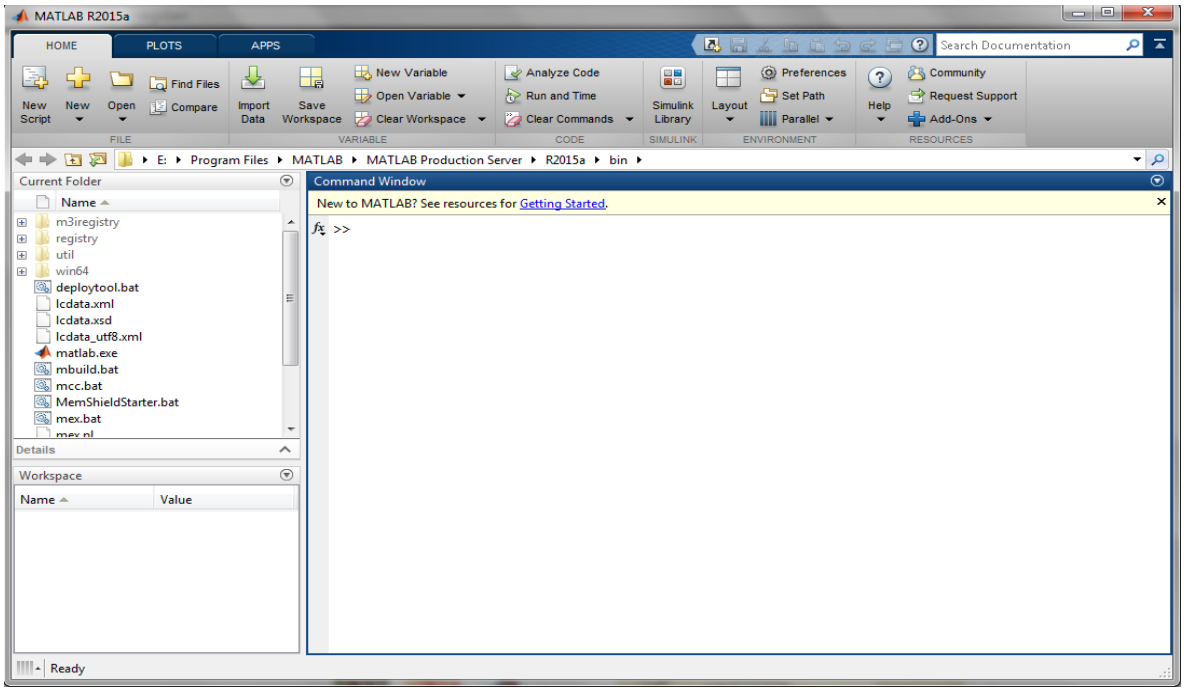


Рис. 3.43. Окно Matlab 15

Далее необходимо нажать на вкладку «Simulink Library», которая находится на панели инструментов в открывшемся окне. Откроется следующее окно:

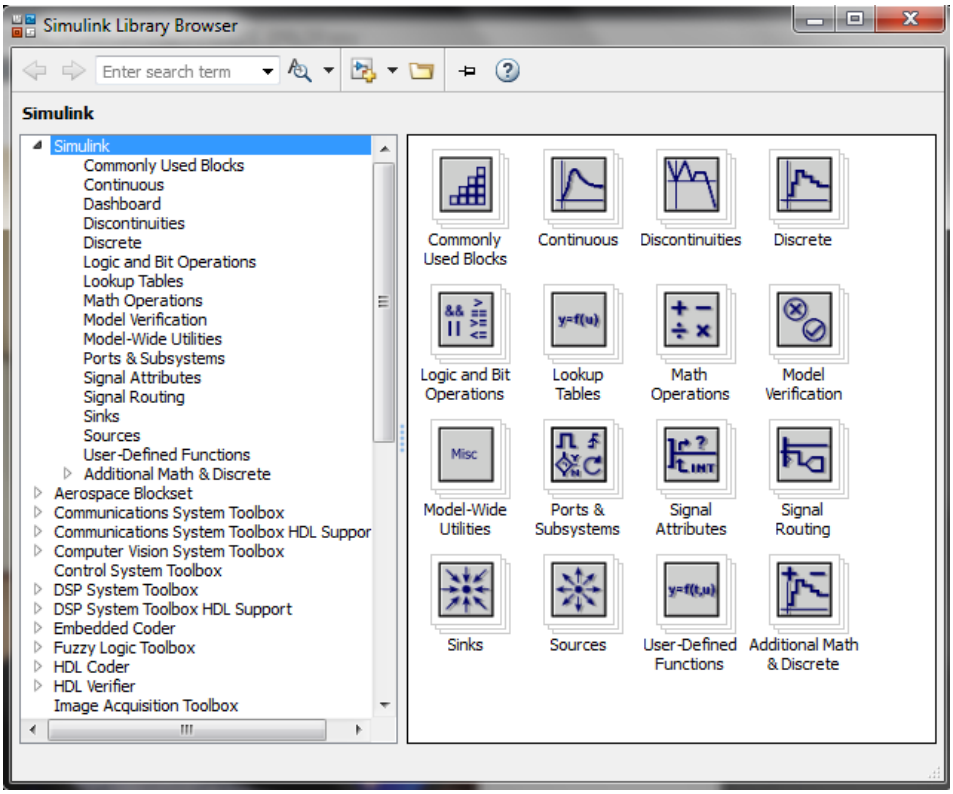


Рис. 3.44. Окно «Simulink Library»

Далее создаем новую модель (вкладка на панели инструментов) и приступаем к реализации описанных выше алгоритмов.

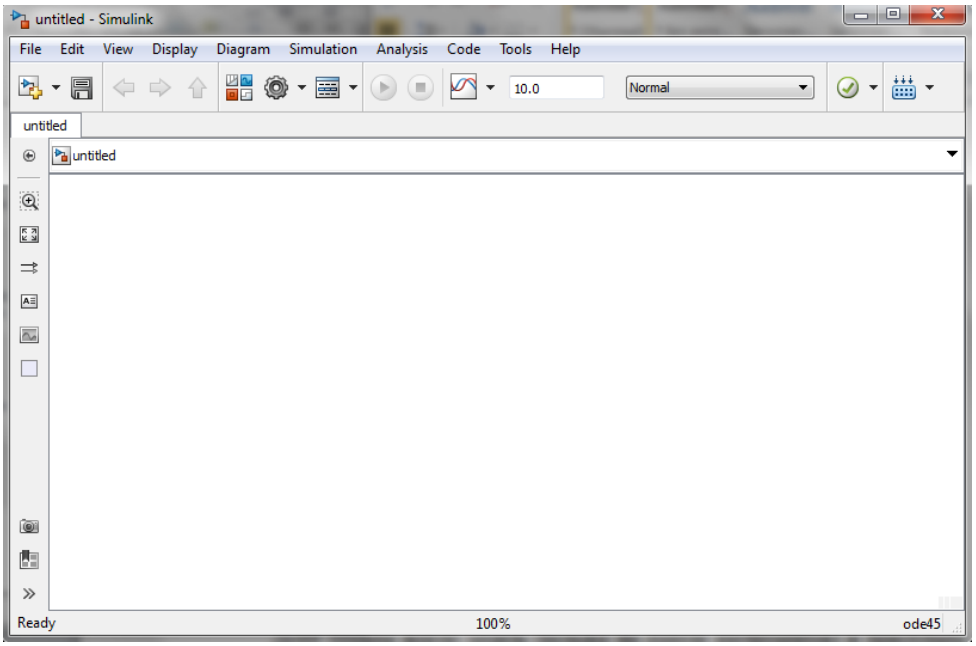


Рис. 3.45. Рабочее поле

Окно, представленное на рисунке 3.44, необходимо для поиска необходимых компонентов для реализации данной работы (кодеры, декодеры и т.д.)

Реализуем канал передачи такой же кодовой комбинации без преобразований.

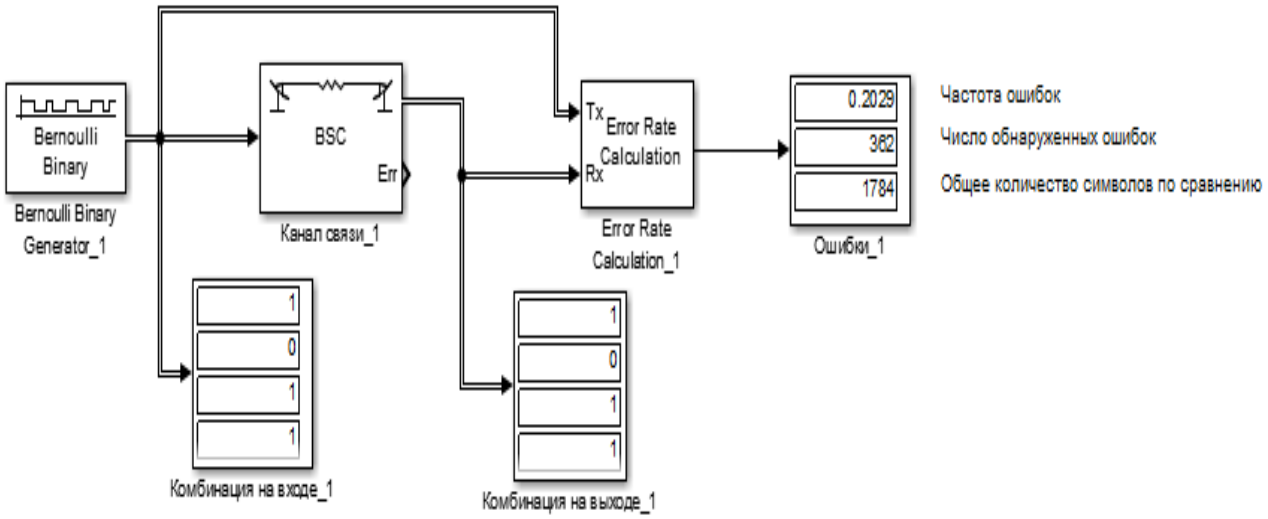


Рис. 3.46. Линия передачи без преобразований

В состав линии без преобразований входят:

1. BernoulliBinaryGenerator
2. Binary Symmetric Channel (каналпередачи)
3. Error RateCalculation (анализаторошибок)
4. Display

Характеристики блоков выставить следующие:

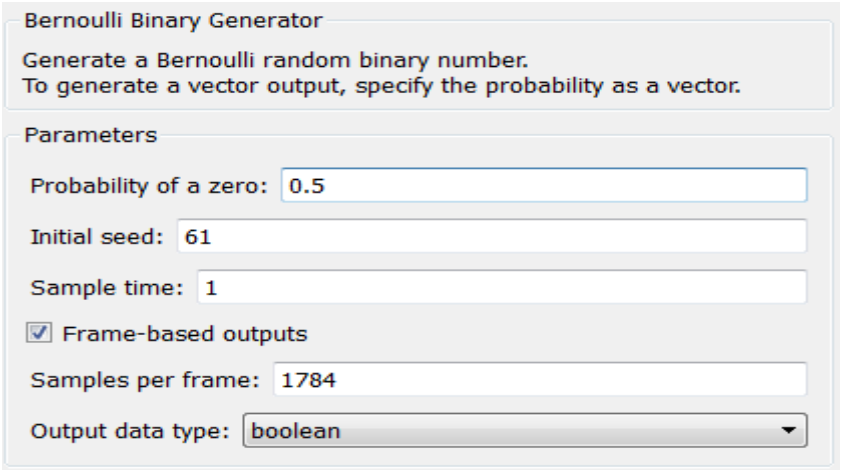


Рис. 3.47. Bernoulli Binary Generator

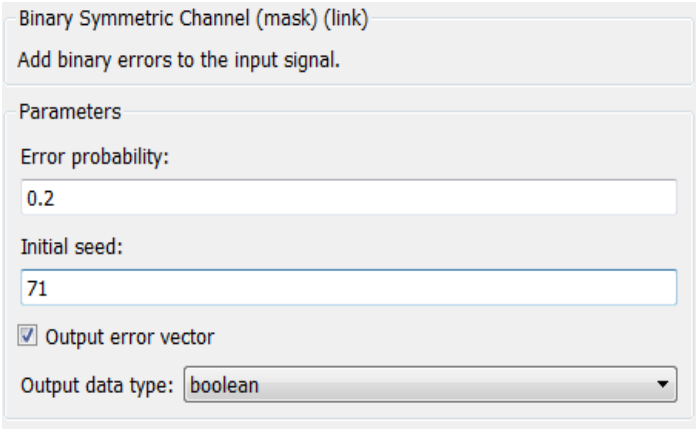


Рис. 3.48. Binary Symmetric Channel

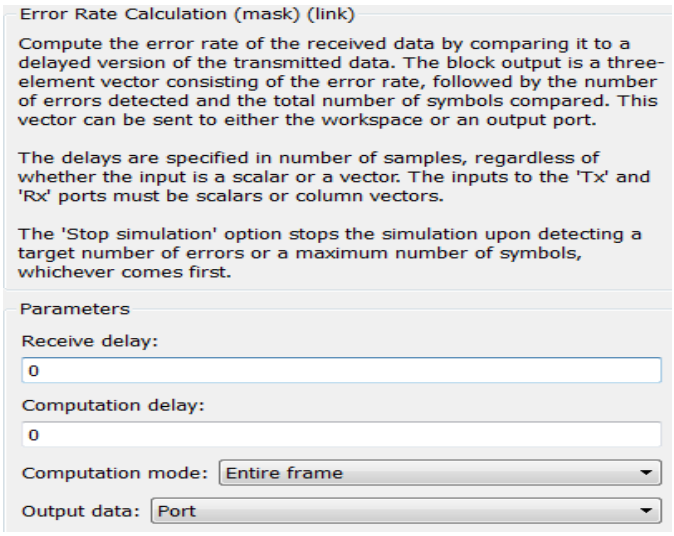


Рис. 3.49. ErrorRateCalculation (анализаторошибок)

Представим полученные результаты:

Входная последовательность:

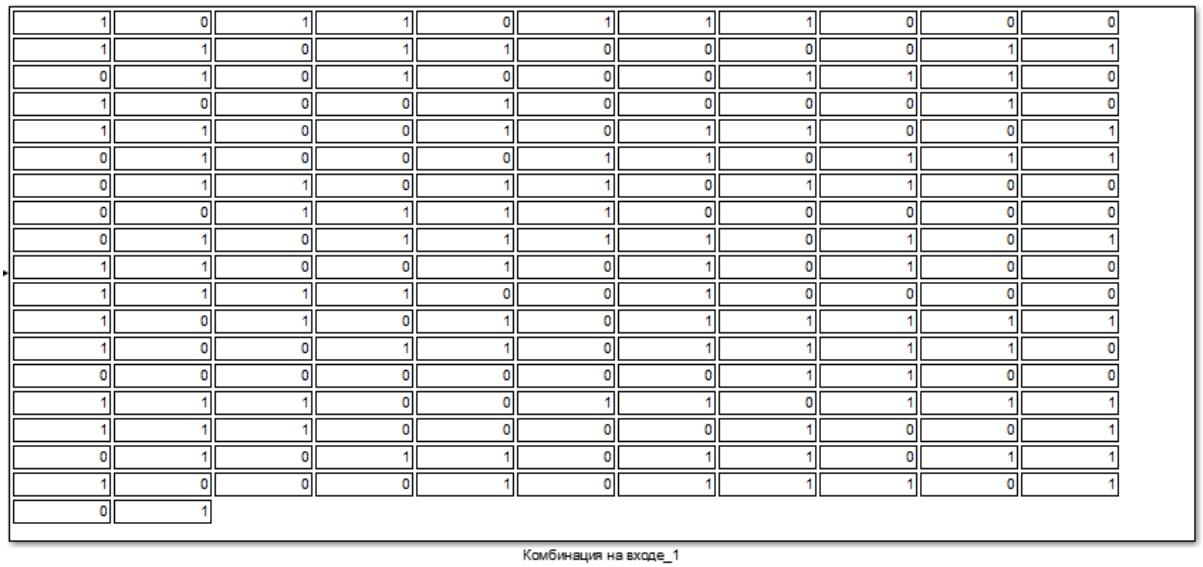


Рис. 3.50. Входная последовательность

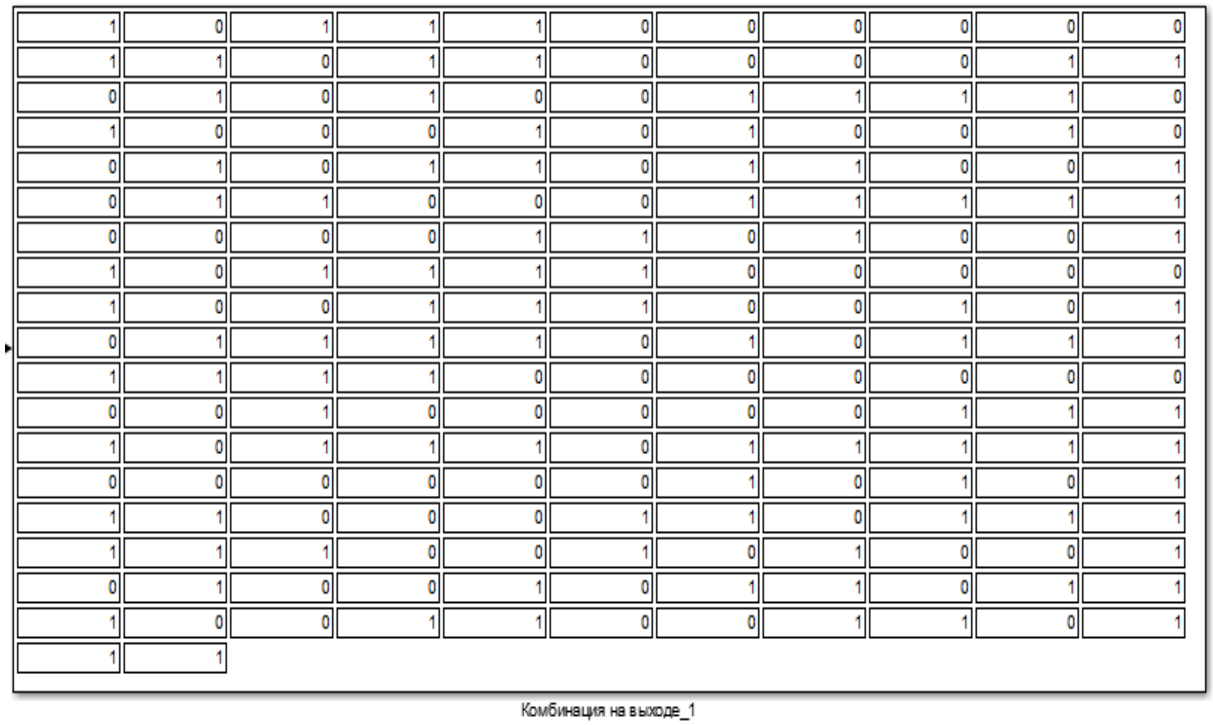


Рис. 3.51. Выходная последовательность

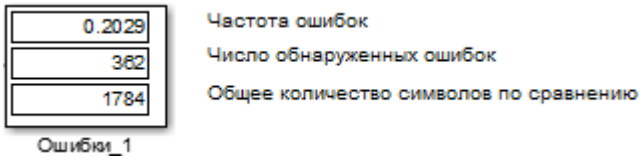


Рис. 3.52. Счетчик ошибки

Анализируя рисунок выше, можно сделать вывод, что комбинация на входе совпадает с комбинацией на выходе, таким образом, передача осуществилась удачно. Что качается ошибок, то их частота равна 0,2029, число обнаруженных ошибок равно 362, общее количество символов по сравнению равно 1784. В данной схеме осуществляется передача информации без преобразований. Если сравнивать этот результат с результатом с кодированием при вероятности ошибки 0,2, то они идентичны. При увеличении вероятности ошибки до 0,3 в декодированной последовательности были найдены ошибки.

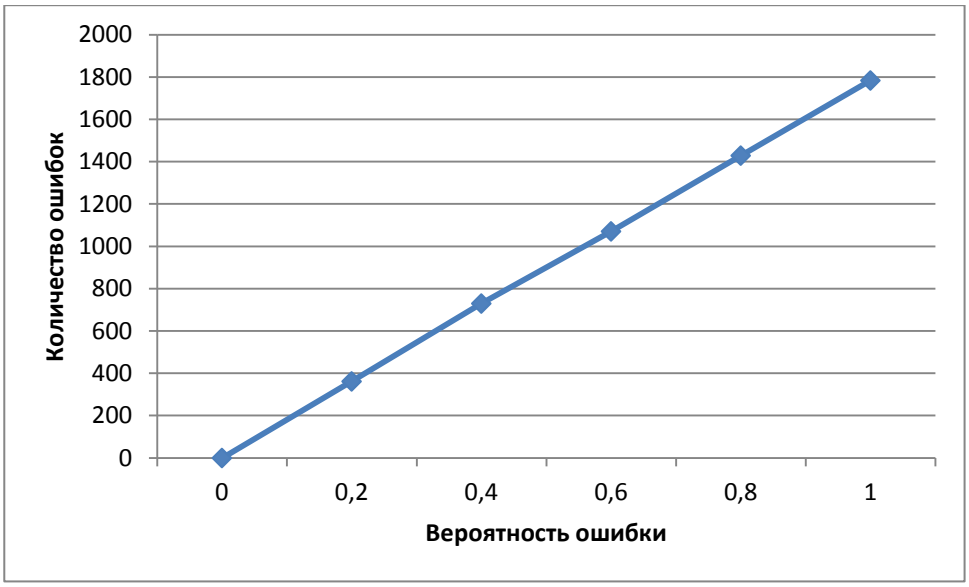


Рис. 3. 53. График зависимости числа обнаруженных ошибок от вероятности ошибки для кода (255,223)

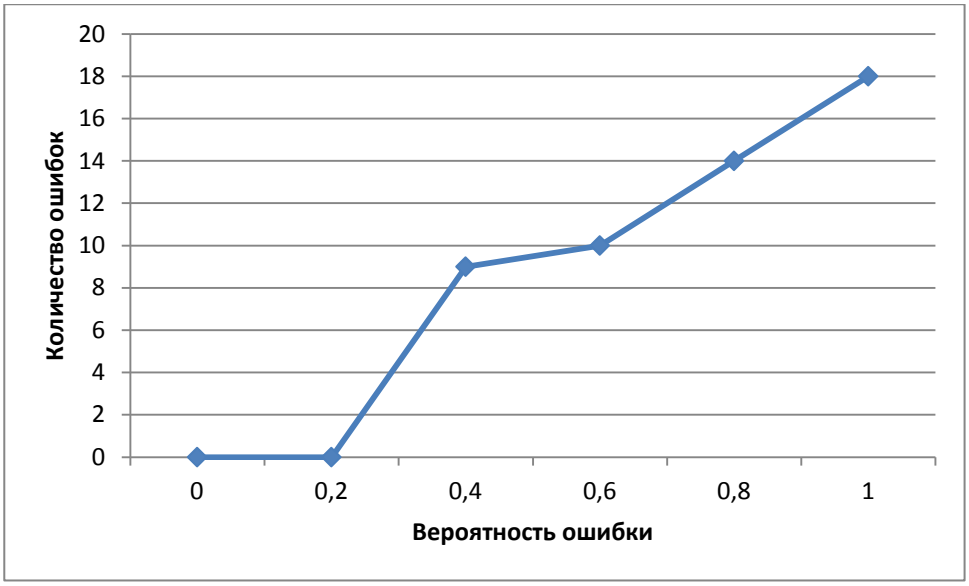


Рис. 3.54. График зависимости числа обнаруженных ошибок от вероятности ошибки для кода (7,3)

Были рассмотрены особенности кода Рида-Соломона. Построили графики зависимости числа обнаруженных ошибок от вероятности ошибки для кодов (255,223) и (7,3).

Изучена структура кодера и декодера.

1. Код Рида-Соломона — код с 8-битными символами и проверкой на чётность. В коде (255,223) 223 байта информационных символов, 32 байта проверки на четность.

2. Объем вычислительной мощности, необходимой для кодирования и декодирования кодов Рида-Соломона, зависит от числа символов четности. Большое значение t означает, что

большее число ошибок может быть исправлено, но это потребует большей вычислительной мощности по сравнению с вариантом при меньшем ***t.***

3. Код RS(255,223) может исправить до 16 ошибок в символах. В худшем случае, могут иметь место 16 битовых ошибок в разных символах (байтах). В лучшем случае, корректируются 16 полностью неверных байт, при этом исправляется 16 * 8 = 128 битовых ошибок.

4. В результате анализа графиков рисунков 2.19-2.20 можно сделать вывод, что код с длинными последовательностями исправляет больше ошибок.

В разделе начало взяло аналитика выбора тех или иных параметров кодов, таких как, выбор размерности кодов, вероятность ошибки в канале и т.д. Сначала были выполнен предварительный анализ, позволившие выбрать и обосновать структурные схемы кодов по исходным данным технического задания.

Путем проведения компьютерной симуляции, была проверена достоверность расчетов. В программе Matlab 15 были собраны схемы:

1) Кода Хэмминга
2) Кода БЧХ
3) Кода Рида-Соломона

С помощью компьютерной симуляции была дана оценка ошибкам при передаче информации по каналу.

Также я познакомился с различным программным обеспечением, для построения различного вида схем.

Подведя итог своего индивидуального задания, можно сказать следующее, то, что я сделал, является основополагающим делом к дальнейшим, более трудным вещам. Индивидуальная работа была весьма увлекательна и полезна. С поставленными целями справился успешно.

Таблица 3.1. Результаты

	Вер-ть ошибки 0,2	Вер-ть ошибки 0,3
Код Хэмминга (7,4)	Частота ошибок: 0,25 Число обнаруженных ошибок: 3 Общее кол-во символов по сравнению: 12	Частота ошибок: 0,4167 Число обнаруженных ошибок: 5 Общее кол-во символов по сравнению: 12
Код БЧХ	Частота ошибок: 0,2143	Частота ошибок: 0,4286

(15,7)	Число обнаруженных ошибок: 3 Общее кол-во символов по сравнению: 14	Число обнаруженных ошибок: 6 Общее кол-во символов по сравнению: 14
Код Р-С (255,223)	Частота ошибок: 0,2029 Число обнаруженных ошибок: 362 Общее кол-во символов по сравнению: 1784	Частота ошибок: 0,31 Число обнаруженных ошибок: 553 Общее кол-во символов по сравнению: 1784

3.2. Циклические избыточные коды CRC (Cyclic redundancy check) [16]

Циклический избыточный код (CRC). Наиболее известными из методов обнаружения ошибок передачи данных являются [1]:

- *Посимвольный контроль чётности*, называемый также поперечным, подразумевает передачу с каждым байтом дополнительного бита, принимающего единичное значение по чётному или нечётному количеству единичных бит в контролируемом байте. Посимвольный контроль чётности прост как в программной, так и в аппаратной реализации, но его вряд ли можно назвать эффективным методом обнаружения ошибок, так как искажение более одного бита исходной последовательности резко снижает вероятность обнаружения ошибки передачи. Этот вид контроля обычно реализуется аппаратно в устройствах связи.
- *Поблочный контроль чётности*, называемый продольным. Схема данного контроля подразумевает, что для источника и приёмника информации заранее известно, какое число передаваемых символов будет рассматриваться ими как единый блок данных. В этой схеме контроля для каждой позиции разрядов в символах блока (поперёк блока) рассчитываются свои биты чётности, которые добавляются в виде обычного символа в конце блока. По сравнению с посимвольным контролем чётности, поблочный контроль чётности обладает большими возможностями по обнаружению и даже корректировке ошибок передачи, но всё равно ему не удаётся обнаруживать определённые типы ошибок.
- *Вычисление контрольных сумм*. В отличие от предыдущих методов, для метода контрольных сумм нет чёткого определения алгоритма. Каждый разработчик трактует понятие контрольной суммы по-своему. В простейшем виде контрольная сумма – это арифметическая сумма двоичных значений контролируемого блока символов. Но этот метод обладает практически теми же недостатками, что и предыдущие, самый главный из которых

– нечувствительность контрольной суммы к чётному числу ошибок в одной колонке и самому порядку следования символов в блоке.

- *Контроль циклически избыточным кодом* – CRC. Это гораздо более мощный и широко используемый метод обнаружения ошибок передачи информации. Он обеспечивает обнаружение ошибок с высокой вероятностью. Кроме того, этот метод обладает рядом других полезных моментов, которые могут найти своё воплощение в практических задачах.

Циклический избыточный код (англ. Cyclic redundancy code, CRC) – алгоритм вычисления контрольной суммы, предназначенный для проверки целостности передаваемых данных. Алгоритм CRC обнаруживает все одиночные ошибки, двойные ошибки и ошибки в нечётном числе бит. Понятие циклических кодов достаточно широкое, однако на практике его обычно используют для обозначения только одной разновидности, использующей циклический контроль (проверку) избыточности.

Главная особенность значения CRC состоит в том, что оно однозначно идентифицирует исходную битовую последовательность и поэтому используется в различных протоколах связи, а также для проверки целостности блоков данных, передаваемых различными устройствами. Благодаря относительной простоте алгоритм вычисления CRC часто реализуется на аппаратном уровне.

При передаче пакетов по сетевому каналу могут возникнуть искажения исходной информации вследствие разных внешних воздействий: электрических наводок, плохих погодных условий и многих других. Сущность методики в том, что при хороших характеристиках контрольной суммы в подавляющем числе случаев ошибка в сообщении приведёт к изменению его контрольной суммы. Если исходная и вычисленная суммы не равны между собой, принимается решение о недостоверности принятых данных, и можно запросить повторную передачу пакета.

Основная идея алгоритма CRC состоит в представлении сообщения в виде огромного двоичного числа, делении его на другое фиксированное двоичное число и использовании остатка этого деления в качестве контрольной суммы [2]. Получив сообщение, приёмник может выполнить аналогичное действие и сравнить полученный остаток с «контрольной суммой».

На рисунках 2.1-2.2 изображено графическое представление кодирования и декодирования CRC-кода [3]:

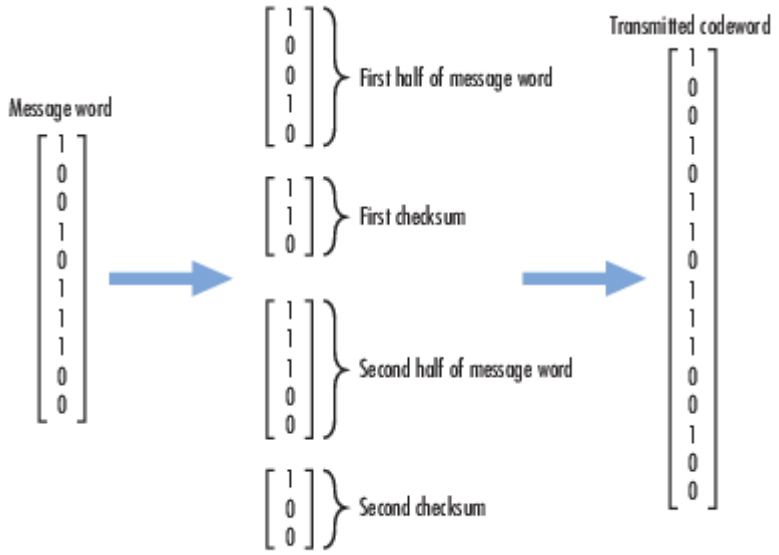


Рис. 3.55. Принцип работы кодера CRC

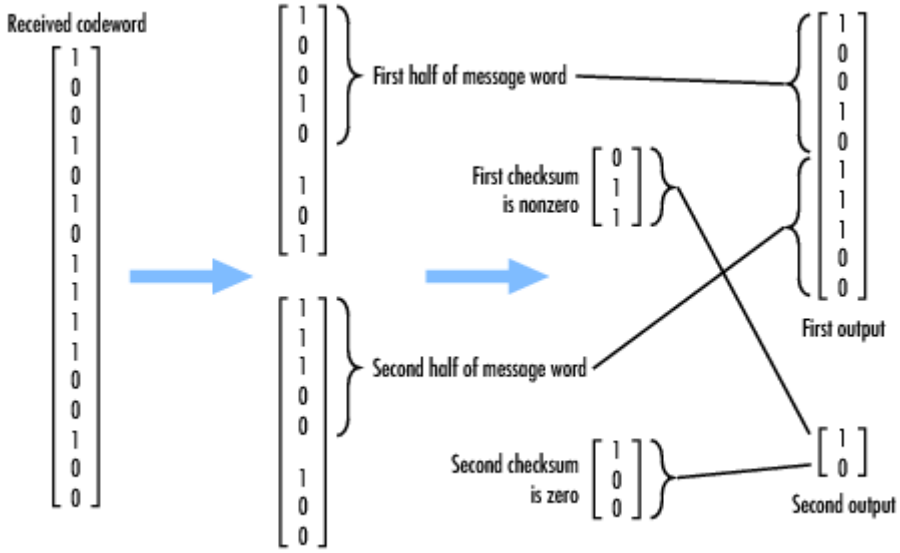


Рис. 3.56. Принцип работы декодера CRC

Степенью CRC-полинома *W* называют позицию самого старшего единичного бита. Например, степень полинома 10011_2 равна 4.

Для вычисления CRC используют полиномиальную арифметику. Вместо представления делителя, делимого (сообщения), частного и остатка в виде положительных целых чисел, можно представить их в виде полиномов с двоичными коэффициентами или в виде строки бит, каждый из которых является коэффициентом полинома.

Например, десятичное число 23 в 16-ричной и 2-ичной системах будет иметь вид $23_{10}=17_{16}=10111_2$, что совпадает с полиномом: $1\cdot x^4+0\cdot x^3+1\cdot x^2+1\cdot x^1+1\cdot x^0$ или упрощённо: $x^4+x^2+x^1+1.$

И сообщение, и делитель могут быть представлены в виде полиномов, с которыми можно выполнять любые арифметические действия без переносов.

Как правило, контрольная сумма добавляется к исходному сообщению и полученное расширенное сообщение передаётся через канал связи.

На другом конце канала приёмник может сделать одно из возможных действий (оба варианта совершенно равноправны):

1. Выделить текст полученного сообщения, вычислить для него контрольную сумму и сравнить её с переданной.

2. Вычислить контрольную сумму для всего переданного сообщения, и посмотреть, получится ли в результате нулевой остаток.

Поскольку исходное сообщение может быть очень большим (до нескольких Мбайтов) и так же из-за того, что для получения CRC используется CRC-арифметика, использовать обычную компьютерную операцию деления нельзя.

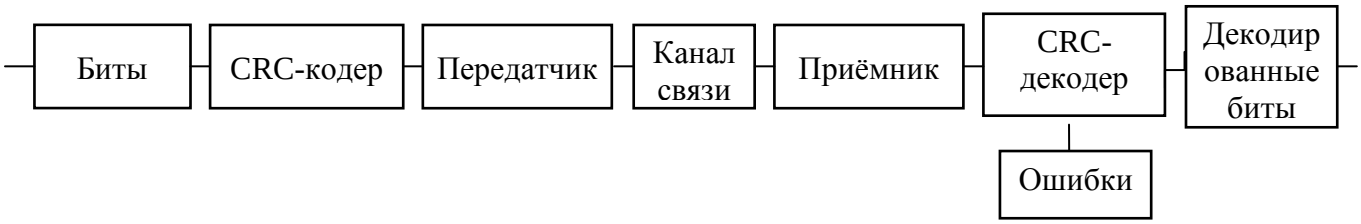


Рис. 3.57. Обобщенная структурная схема исследования CRC кодов

Самый популярный и рекомендуемый IEEE полином для CRC-32 используется в Ethernet, FDDI; также этот многочлен является генератором кода Хемминга. Использование другого полинома — CRC-32C — позволяет достичь такой же производительности при длине исходного сообщения от 58 бит до 131 кбит, а в некоторых диапазонах длины входного сообщения может быть даже выше — поэтому в наши дни он тоже пользуется популярностью. К примеру, стандарт ITU-T использует CRC-32C с целью обнаружения ошибок в полезной нагрузке.

Ниже в таблице перечислены наиболее распространённые многочлены — генераторы CRC [4]:

Таблица 3.2. Распространённые полиномы CRC кодов

Название	Полином
CRC-1	$x+1$ (используется в аппаратном контроле ошибок, также известен как бит чётности)
CRC-4-ITU	x^4+x+1
CRC-5-ITU	$x^5+x^4+x^2+1$
CRC-5-USB	x^5+x^2+1
CRC-6-ITU	x^6+x+1
CRC-7	x^7+x^3+1 (системы телекоммуникации, ITU-T G.707, ITU-T G.832, MMC, SD)
CRC-8	$x^8 + x^7 +x^6+ x^4+x^2+1$
CRC-16-IBM	$x^{16}+x^{15}+x^2+1$ (Bisync, Modbus, USB, ANSI X3.28)
CRC-16-CCITT	$x^{16}+x^{12}+x^5+1$ (X.25, HDLC, XMODEM, Bluetooth, SD)
CRC-30	$x^{30}+x^{29}+x^{21}+x^{20}+x^{15}+x^{13}+ x^{12}+ x^{11}+ x^8+ x^7+ x^6+ x^2+ x+1$ (CDMA)

Программная реализация виртуальных моделей кодирования

Описание реализации циклического избыточного кода (CRC)

Виртуальная модель передачи данных с обнаружением ошибок при помощи CRC-кода была реализована в среде Simulink Matlab. Модель демонстрирует работу CRC-кодера и декодера, позволяет исследовать обнаруживающую способность кода для разных генераторных полиномов.

На рисунке 3.58 приведена разработанная модель:

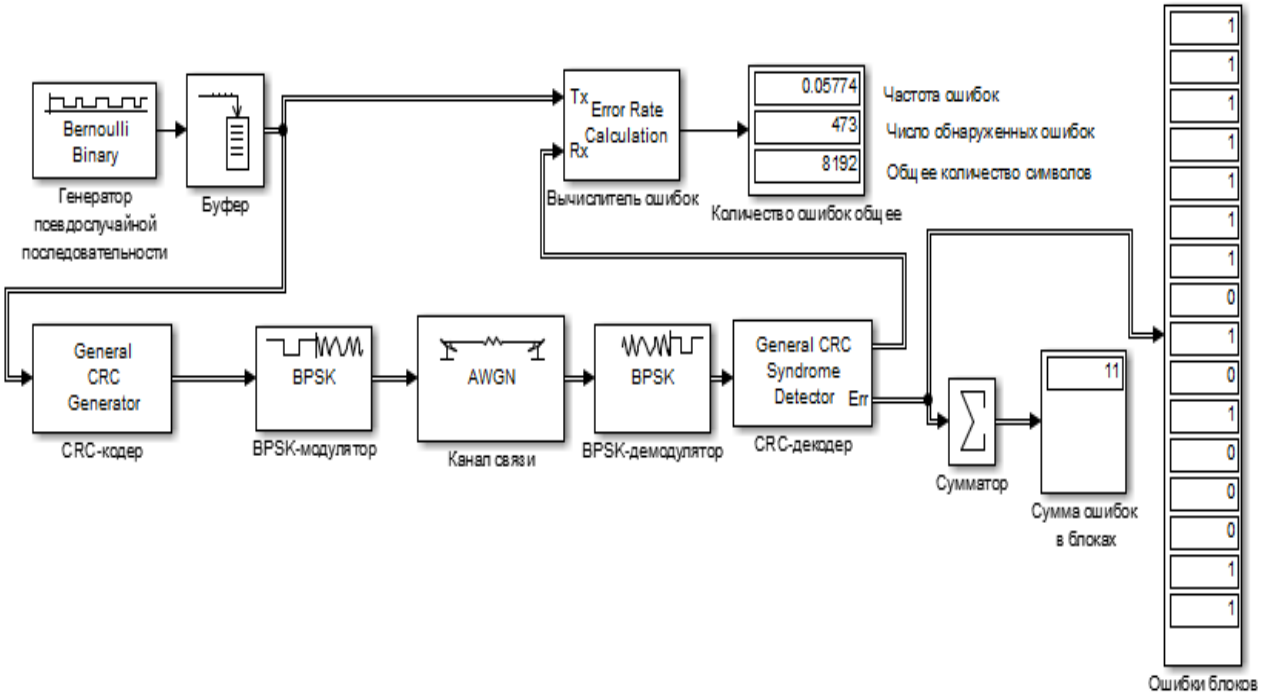


Рис. 3.58. Разработанная модель исследования CRC-кодов

В её основу положены следующие элементы, встроенные в библиотеку Simulink:

- Bernoulli Binary Generator;
- General CRC Generator;
- BPSK Modulator Baseband;
- AWGN Channel;
- BPSK Demodulator Baseband;
- General CRC Syndrome Detector;
- Error Rate Calculation;
- Buffer;
- Add;
- Display (Дисплей, отражающий ошибки).

Далее представлено описание основных блоков:

Bernoulli Binary Generator (генератор псевдослучайной последовательности) – генерирует случайную бинарную последовательность (рисунок 3.59).

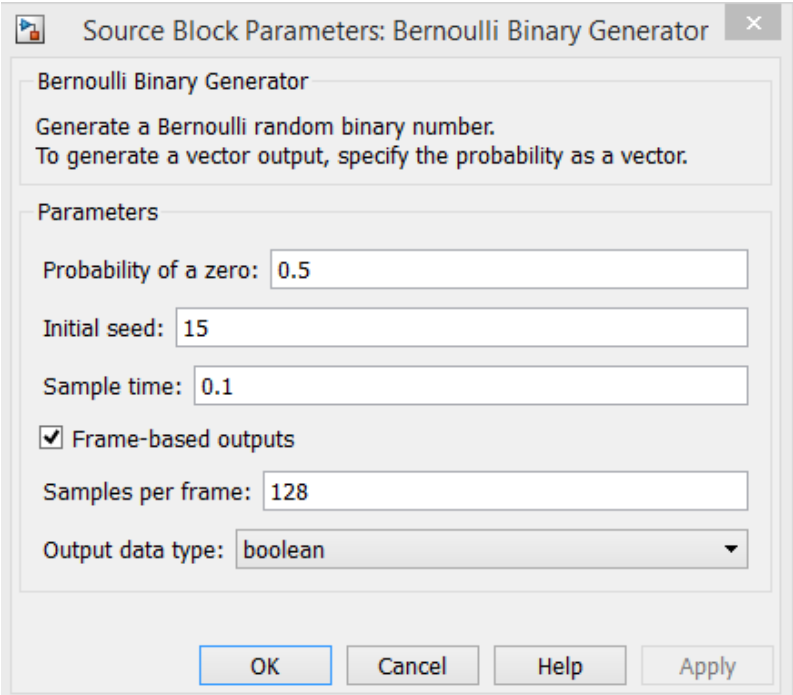


Рис. 3.59. Параметры блока «Bernoulli Binary Generator»

«Probability of a zero» - вероятность появления нуля;

«Initial seed» - начальное значение для генерации;

«Sample time» - длительность сэмпла;

«Samples per frame» - размер фрейма.

General CRC Generator (CRC-кодер) – циклический избыточный кодер (рисунок 3.3).

«Generator polynomial» - генераторный полином, может быть задан в 3 формах:

1) В обычной записи, например: x^3 + x^2 + x + 1.

2) в виде матрицы-строки с указанием степеней с ненулевыми коэффициентами, например: [4 1 0] = x^4 + x + 1.

3) в виде матрицы-строки с указанием нулевых и ненулевых коэффициентов, например: [1 1 0 1 1] = x^4 + x^3 + x + 1.

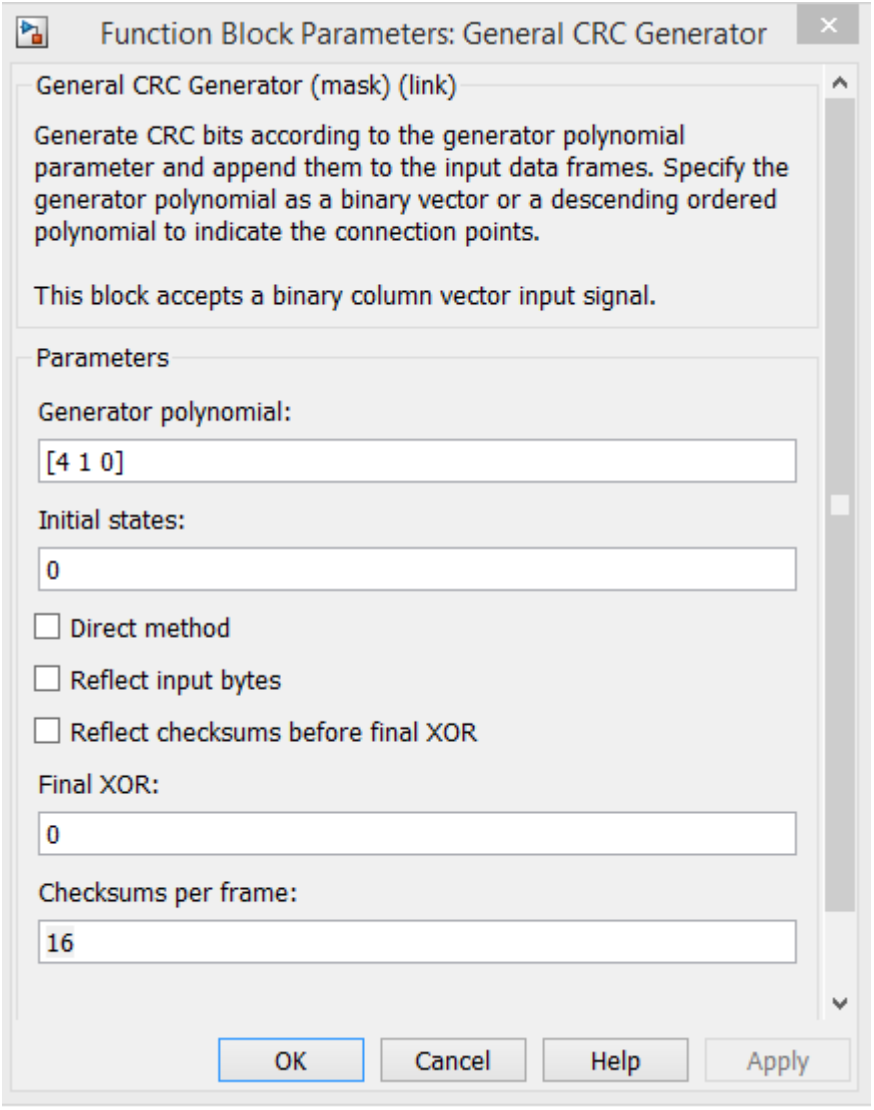


Рис. 3.60. Параметры блока «General CRC Generator»

«Initial states» - начальное состояние сдвиговых регистров.

«Direct method» - включение прямого метода вычисления CRC, иначе работает по табличному методу.

«Reflect input bytes» - инвертировать входной поток.

«Reflect checksums before final XOR» - инвертировать контрольные суммы перед конечной операцией XOR.

«Final XOR» - Выполнить операцию XOR в конце кодирования.

«Checksums per frame» - количество контрольных сумм во фрейме.

BPSK Modulator Baseband – BPSK модулятор.

BPSK Demodulator Baseband – BPSK демодулятор.

AWGN Channel (Канал связи) – добавляет «белый» гауссовский шум в канале (рисунок 3.61).

«SNR» - задаёт отношение сигнал/шум в канале.

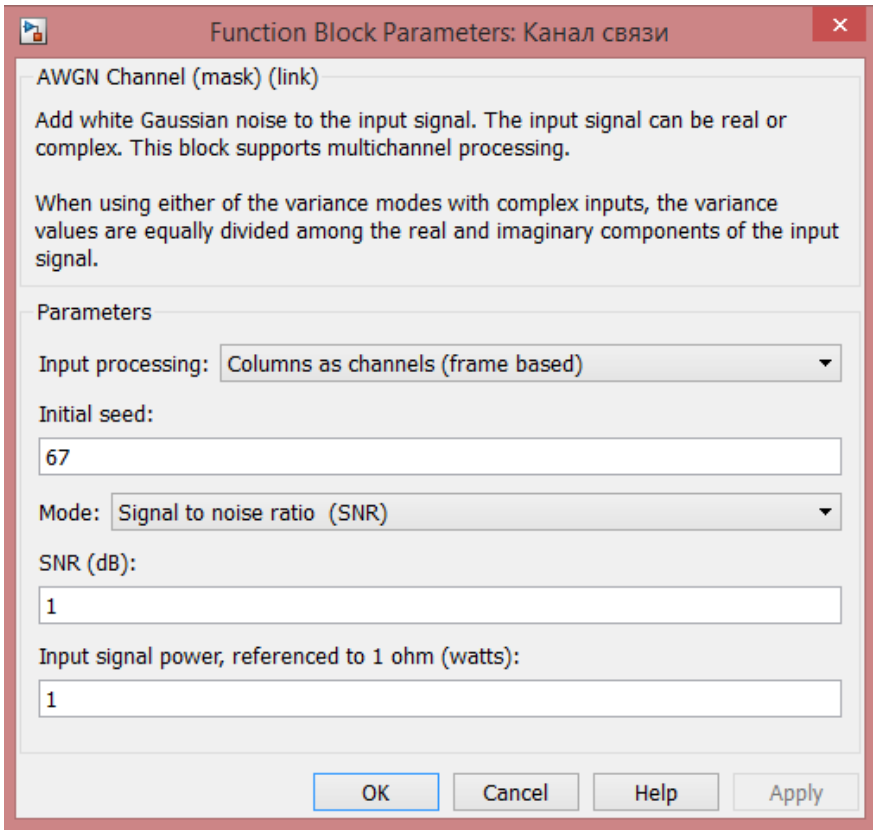


Рис. 3.61. Параметры блока «AWGN»

General CRC Syndrome Detector - циклический избыточный декодер. Все параметры декодера задаются аналогично параметрам блока «General CRC Generator» (рисунок 3.3).

Error Rate Calculation – вычислитель ошибок между переданной и принятой последовательностью.

Buffer – буфер. Переводит последовательность бит в один блок.

Add (сумматор) – суммирует ошибки от CRC-декодера.

Display - дисплей, отражающий ошибки.

Результаты моделирования

Исследование циклического избыточного кода

Модель циклического избыточного кода (crc), разработанная представленная на рисунке 3.58, позволяет исследовать обнаруживающую способность CRC кодов с различными полиномами.

Задаём одинаковый генераторный полином в блоки CRC-кодер и CRC-декодер (рисунок 3.62):

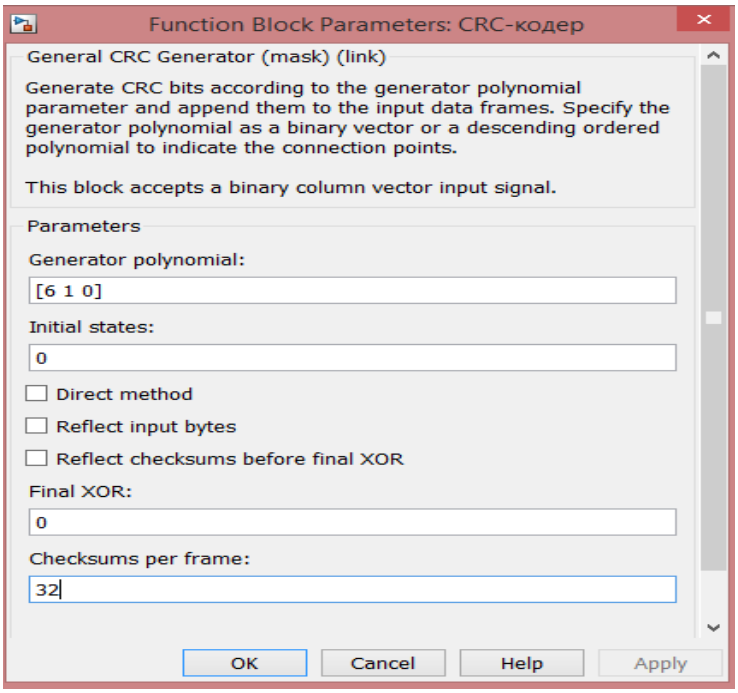


Рис. 3.62. Параметры блока CRC-кодер

Общее число передаваемых символов составляет 8192. Количество контрольных сумм изменяется от 2 до 8192, с увеличением каждого предыдущего значения в 2 раза (2, 4, 8, 16…8192).

Значение SNR в блоке «Канал связи» установлено в 1 дБ. Таким образом, битовая вероятность ошибки (BER) составит 0,05786.

На рисунке 4.2 представлен график зависимости числа обнаруженных ошибок от числа контрольных сумм для различных полиномов CRC-кода.

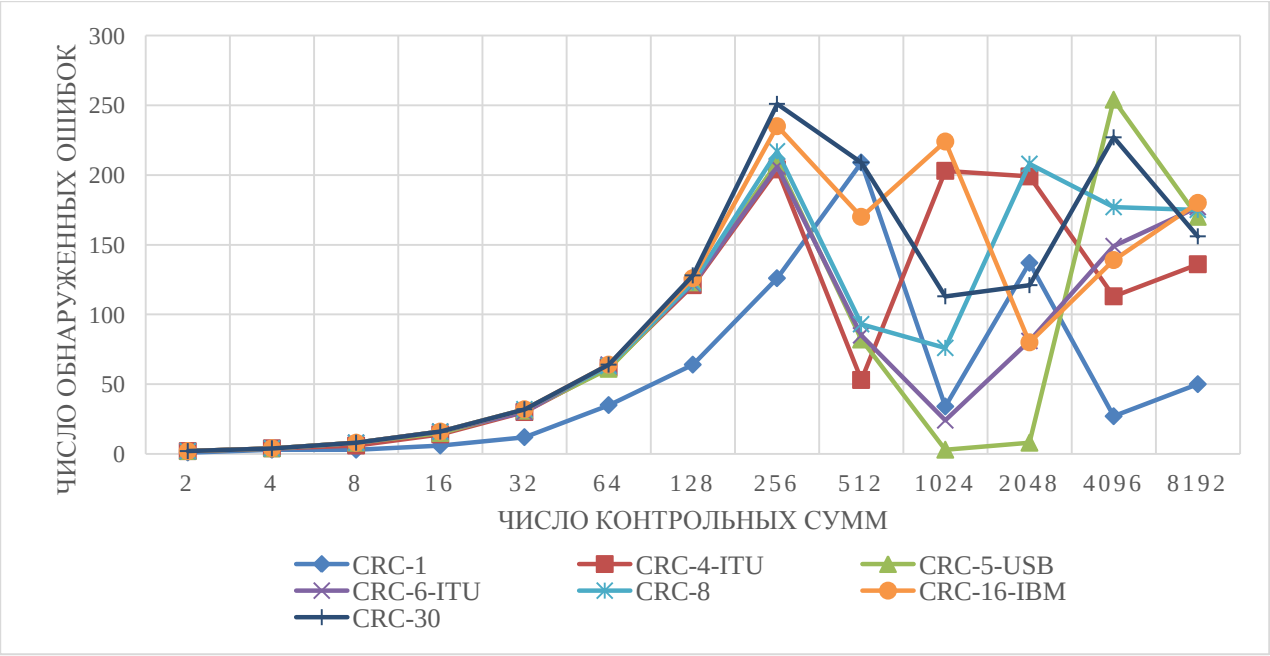


Рис. 3.63. График зависимости числа обнаруженных ошибок от числа контрольных сумм для различных полиномов CRC-кода

В данном разделе проведено исследование модели циклического избыточного кода (CRC).

Модель позволяет исследовать CRC-коды с возможностью задания любого генераторного полинома и изменении количества контрольных сумм во фрейме.

Получены следующие результаты и сделаны следующие выводы:

1) чем выше степень полинома, тем лучше его обнаруживающая способность;

2) для каждого полинома есть такое число контрольных сумм в блоке, при котором его обнаруживающая способность максимальна, причём у всех полиномов эти точки различны.

Однако, при выборе полинома CRC-кода также необходимо учитывать и другие факторы:

1) увеличение степени полинома приводит к усложнению реализации кодера и декодера;

2) чем выше частота вычисления контрольных сумм, т.е. чем больше контрольных сумм добавляется в блок данных, тем меньше пропускная способность канала;

3) CRC-коды используют для обнаружения ошибок, что означает наличие канала переспроса. При выборе между кодом CRC/каналом переспроса и помехоустойчивым кодированием, необходимо учитывать характеристики канала связи. При большом числе ошибок передача данных будет невозможна.

4) Выбор полинома зависит от размера передаваемого блока данных, чем больше блок – тем выше степень полинома необходимо подбирать. Таким образом, существует ограничение на размер блока данных, иначе в любом блоке на приёмном конце будет обнаруживаться ошибка.

3.3. Сверточные коды. Декодирование сверточных кодов

Современная теория кодов достаточно развита и содержит детальную классификацию. Все применяемые коды можно разбить на две большие группы: блоковые, в которых кодирование и декодирование производится в пределах определенного участка кодовой последовательности- блока, и древовидные, в которых обработка символов производится непрерывно, без разделения на блоки. Часть кодов относится к разряду линейных, в которых кодовые последовательности представлены как элементы линейного векторного пространства. Можно применить также разбиение на коды, исправляющие независимые случайные ошибки, и коды, исправляющие пакетные ошибки.

В отличие от блоковых, свёрточные коды обладают следующими преимуществами:

- сверточные коды позволяют производить кодирование и декодирование потоков данных непрерывно во времени;

- сверточные коды не нуждаются в блоковой синхронизации;

- применение сверточных кодов позволяет достичь очень высокой надежности передаваемой информации.

Сверточные коды используются при низком отношении сигнал-шум, когда исправляющей способности блоковых кодов при разумной длине блока оказывается недостаточно.

Сверточное кодирование, удобнее всего описывать, характеризуя действие соответствующего кодирующего устройства. *Сверточный кодер* представляет собой устройство воспринимающее за каждый такт работы в общем случае *к* входных информационных символов и выдающее на выход за тот же такт *n* выходных символов, подлежащих передаче по каналу связи. Параметром сверточного кода, характеризующим его помехоустойчивость, является минимальное свободное расстояние — d_c, определяемое как минимальное расстояние по Хэммингу между последовательностями сверточного кода на длине кодовых ограничений по выходу. Кодовое ограничение по выходу – это число символов на выходе кодера, в формировании которых участвует один входной бит. Эффективность сверточного кода определяется в основном тем, каким образом соединены сумматоры с ячейками регистра сдвига.

Отношение $R = k/n$ называют относительной скоростью кода. Выходные символы, создаваемые кодером на данном такте, зависят от *k* информационных символов, поступивших на этом и предыдущем тактах. Таким образом, выходные символы сверточного кодера однозначно определяются его входным сигналом и состоянием, зависящим от *m-k* предыдущих информационных символов.

Основными элементами сверточного кодера являются: регистр сдвига, сумматоры по модулю 2 и коммутатор.

Регистр сдвига является динамическим запоминающим устройством (рисунок 3.55), в котором хранятся двоичные символы 0 или 1. Число триггерных ячеек *m* в регистре сдвига и определяет память кода. В момент поступления на вход регистра нового информационного символа символ, хранящийся в крайнем правом разряде, выводится из регистра и сбрасывается. Каждый из остальных, хранящихся в регистре символов перемещается на один разряд вправо, освобождая тем самым крайний левый разряд, куда и поступает новый информационный символ. [6]

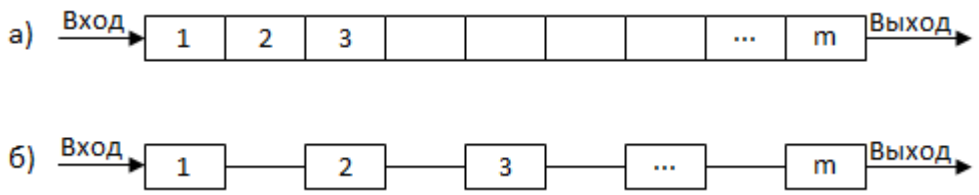


Рис. 3.64. Регистр сдвига

Используются два различных изображения регистров сдвига: с состыкованными впритык ячейками (рисунок 3.64, а) и с непосредственными последовательными связями между ячейками (рисунок 3.64, б), что дает возможность на схемах встраивать между соответствующими ячейками добавочные элементы (схемы суммирования по модулю 2).

Сумматор по модулю 2 осуществляет сложение поступающих на его входы символов 0 и 1. Правило сложения по модулю 2 следующее: сумма двоичных символов равна 0, если число единиц среди поступающих на входы символов четно, и равна 1, если это число нечетно.

Коммутатор осуществляет последовательное считывание поступающих на его входы (контакты) символов и устанавливает на выходе очередность посылки кодовых символов в канал связи.

По аналогии с блоковыми кодами, сверточные коды можно классифицировать на *систематические* и *несистематические*.

Систематическим сверточным кодом является такой код, для которого в выходной последовательности кодовых символов содержится без изменения породившая ее последовательность информационных символов. В противном случае сверточный код является несистематическим.

Свёрточный код создаётся прохождением передаваемой информационной последовательности через линейный сдвиговый регистр с конечным числом состояний. В общем, регистр сдвига состоит из M (m-битовых) ячеек и линейного преобразователя, состоящего из n функциональных генераторов и выполняющего алгебраические функции, как показано на рисунке 3.65.

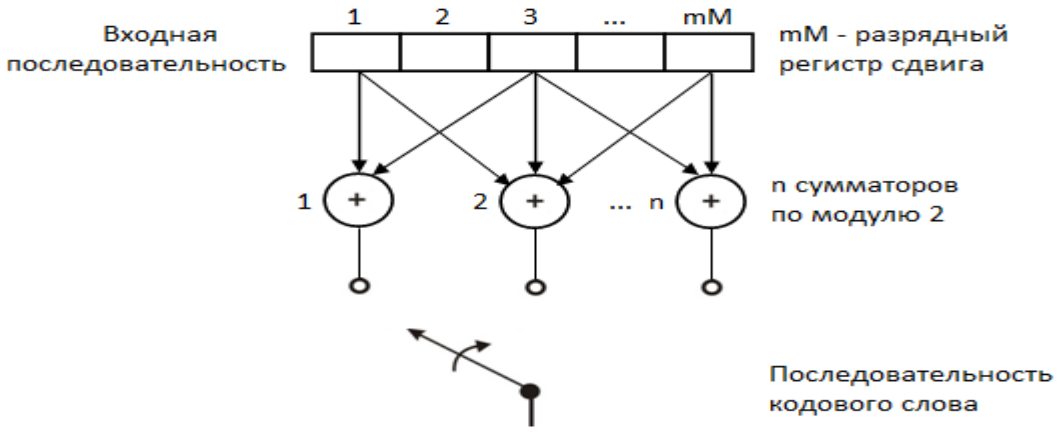


Рис. 3.65. Общий вид сверточного кодера

Входные данные к кодеру, которые считаются двоичными, продвигаются вдоль регистра сдвига по k бит за раз. Число выходных битов для каждой k-битовой входной последовательности равно n. Следовательно, кодовая скорость, определённая как $R = k/n$, согласуется с определением скорости блокового кода.

Представление сверточных кодов

Графическое представление

Рассмотрим свёрточный кодер со скоростью кода 1/2, его графическое представление показано на рисунке 3.57. В этом кодере каждый раз, информационный бит поступает на вход регистров сдвига, а на выходе генерируется два бита.

В этом кодере каждый раз, информационный бит поступает на вход регистров сдвига, а на выходе генерируется два бита.

В качестве примера рассмотрена ситуация, когда на вход кодера подаётся некая последовательность информационных символов *V*= 1 1 0 1 0 1 на выходе имеем последовательность *U* = 11 10 00 10 00 01. Процесс образования выходных символов легко воспроизвести в уме глядя на рисунок.

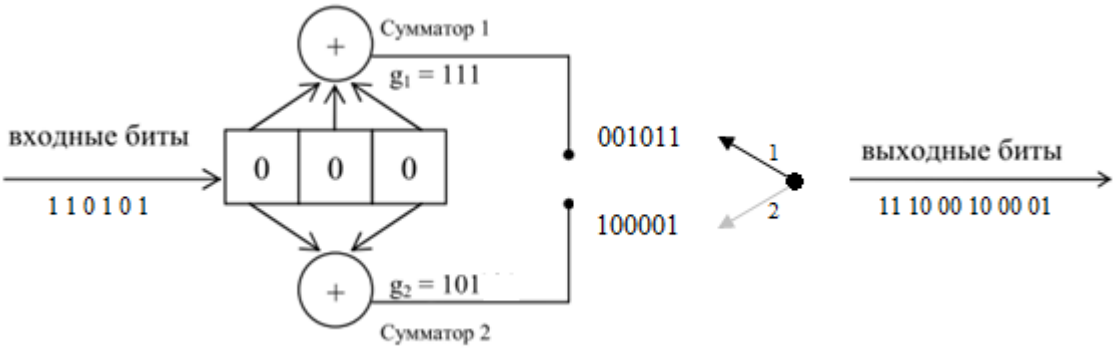


Рис. 3.66. Свёрточный кодер с K= 3, k= 1, n= 2

Старшие разряды входных информационных бит для удобства располагаются справа и поступают в регистр сдвига первыми. При поступлении на вход регистра одного бита, на выходах сумматора 1 и сумматора 2 образуется по одному биту, которые в дальнейшем поочередно считываются коммутатором и образуют выходную последовательность.

Более подробно весь процесс кодирования информационного потока 00...101 приведен в таблице 3.3.

Таблица 3.3. Процесс кодирования информационного потока

№ такта	Входной информационный бит	Состояние регистра сдвига	Сумматор 1	Сумматор 2	Выходные кодовые комбинации
0	-	0000			-
1	1	1000	1⊕0⊕0=1	1⊕ 0⊕0=1	11

2	0	0100	0⊕0⊕0=0	0⊕ 1⊕0=1	01
3	1	1010	1⊕1⊕0=0	1⊕ 0⊕0=1	01
4	1	1101	1⊕0⊕1=0	1⊕ 1⊕1=1	01
5	0	0110	0⊕1⊕0=1	0⊕ 1⊕0=1	11
6	1	1011	1⊕1⊕1=1	1⊕ 0⊕1=0	10

В данной таблице видно, как меняется состояние регистра сдвига с приходом нового информационного бита, показан и процесс образования выходных кодовых комбинаций образуемых в результате считывания коммутатором символов с обоих сумматоров.

Полиномиальное представление

Иногда связи кодера описываются с помощью *полиномиального генератора*, аналогичного используемому для описания реализации обратной связи регистра сдвига циклических кодов. Сверточный кодер можно представить в виде набора из *n* полиномиальных генераторов, по одному для каждого из *n* сумматоров по модулю 2. Каждый полином имеет порядок *K* - 1 или меньше и описывает связь кодирующего регистра сдвига с соответствующим сумматором по модулю 2, почти так же как и вектор связи. Коэффициенты возле каждого слагаемого полинома порядка *(K* - 1) равны либо 1, либо 0, в зависимости от того, имеется ли связь между регистром сдвига и сумматором по модулю 2. [7].

Для кодера на рисунке 3.57, можно записать полиномиальный генератор $g_1(X)$ для верхних связей и $g_2(X)$ - для нижних.

$g_1(X) = 1 + X + X^2$, $g_2(X) = 1 + X^2$

Здесь слагаемое самого нижнего порядка в полиноме соответствует входному разряду регистра. Выходная последовательность находится следующим образом:$U(X) = m(X)g_1(X)$ чередуется с $m_1(X)g_2(X)$.

Прежде всего, необходимо выразить вектор некого сообщения m = 1 0 1 в виде полинома, т.е. $m(X) = 1 + X^2$. Для очистки регистра предполагается использование нулей, следующих за битами сообщения. Тогда выходящий полином U(X), или выходящая

последовательность U кодера (рисунок 3.57) для входного сообщения m может быть найдена следующим образом:

$m(X)g_1(X) = (1+X^2)(1+X+X^2) = 1 + X + X^3 + X^4$

$m(X)g_2(X)=(1+X^2)(1+X^2)=1+X^4$

$m(X)g_1(X) =1+X+0X^2+X^3+X^4$

$m(X)g_2(X)=1+0X+0X^2+0X^3+X^4$

$U(X)= (1,1)+(1,0)X+(0,0)X^2 +(1,0)X^3+(1,1)X^4$

U= 11 10 00 10 11

В этом примере кодер был представлен в виде *полиномиальных генераторов*, с помощью которых описываются циклические коды [7]. *Полиномиальные генераторы кодера изображенного на рисунке 2.57 так же можно представить в виде векторов:*

$g_1 = [111]$ и $g_2 = [101]$.

Представление в виде кодовых древ, решеток и диаграмм состояний

Помимо графического и полиномиального представления имеются три альтернативных метода, которые часто используются для описания свёрточного кода. Это древовидная диаграмма, решетчатая диаграмма и диаграмма состояний. Для примера, древовидная диаграмма для свёрточного кодера, показанного на рисунке 3.57 иллюстрируется на рисунке 3.58.

Рассмотрим свёрточный код со скоростью 1/2, K=3, описанный ранее и показанный на рисунке 3.57. Первый входной бит может быть 0 или 1. Соответствующие выходные биты – 00 или 11. Когда следующий бит входит в кодер, первый бит передвигается в следующую ячейку. Соответствующие выходные биты зависят от бита, переместившегося во вторую ячейку и нового входного бита. Следовательно, древовидная диаграмма для этого кода, показанная на рисунке 3.58, имеет две ветви на узел, соответствующие двум возможным входным символам. Жирной линией на рисунке 3.58 обозначен путь, по которому бы шел кодер, если бы входной комбинацией была последовательность 101.

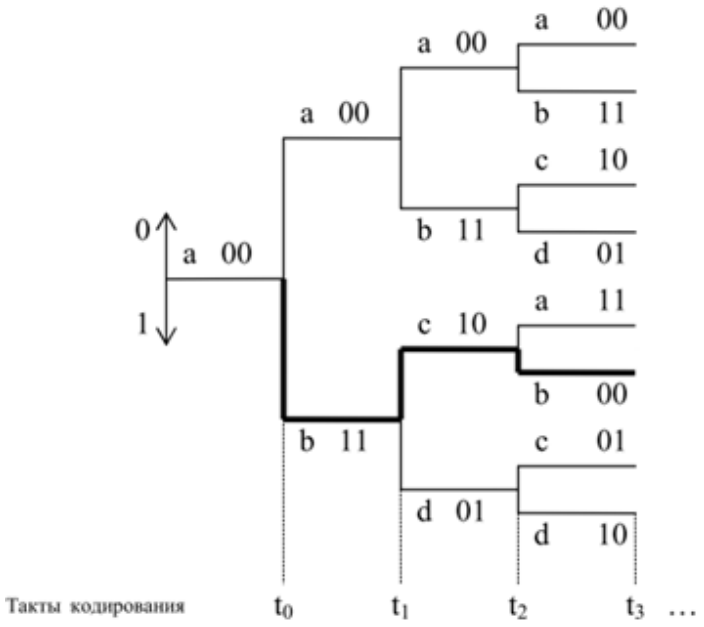


Рис. 3.67. Древовидная диаграмма для свёрточного кода, имеющего параметры K= 3, k= 1, n= 2

Поскольку кодовое ограничение по входу кодера K= 3, а значит дерево начнёт повторяться после третьего шага. Как показано на рисунке 3.58, все ветви, исходящие из узла, обозначенного *a* (состояния *a*) дают идентичные выходы. Путем слияния узлов, имеющих одинаковое название, мы получаем решётку, показанную на рисунке 3.68. На нём жирной линией обозначен путь по которому бы шел кодер, если бы входной комбинацией была последовательность 1100, в таком случае на выходе имелась бы комбинация 11 01 01 11.

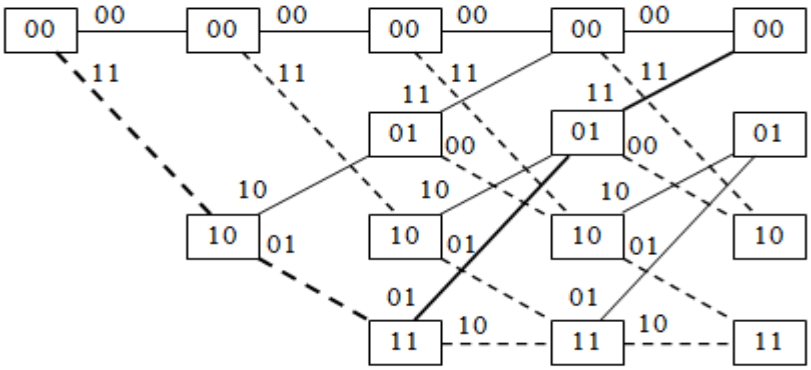


Рис. 3.68. Решётчатая диаграмма для сверточного кода, имеющего параметры K= 3, k=1, n= 2

Поскольку выход кодера определяется входом и состоянием кодера, ещё более компактной, чем решётка, является диаграмма состояний. Диаграмма состояний — это граф возможных состояний кодера и возможных переходов из одного состояния в другое. Итоговая диаграмма состояний для этого кодера показана на рисунке 3.60.

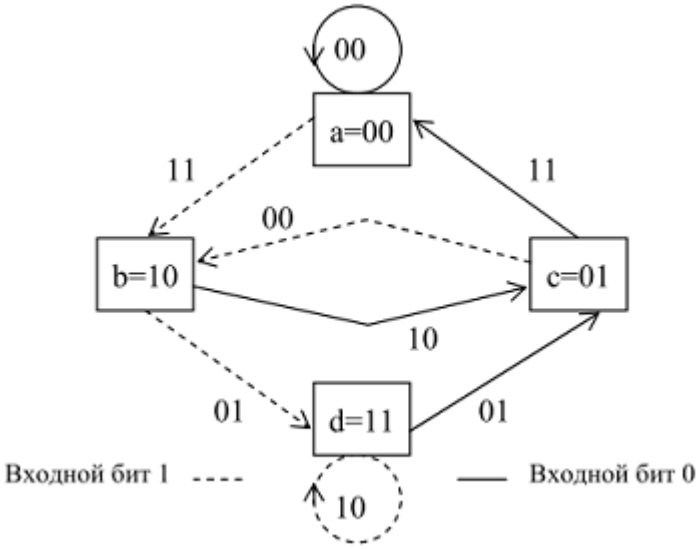


Рис. 3.69. Диаграмма состояний для свёрточного кода, имеющего параметры K= 3, k= 1, n= 2

Для обобщения стоит отметить, что свёрточный код со скоростью *k/n* и кодовым ограничением K характеризуется 2^k ветвями, уходящими от каждого узла на древовидной диаграмме. Решетка и диаграмма состояний имеют (каждая из них) $2^{k(K-1)}$ возможных состояний. Имеются 2^k ветвей, входящих в каждое состояние, и 2^k ветвей, покидающих каждое состояние (для решётки и дерева это верно после наступления установившегося режима) [1, 10].

Методы декодирования свёрточных кодов

Метод порогового декодирования

При пороговом декодировании сверточных кодов вычисляются синдромы (признаки места ошибочных символов), затем эти синдромы или последовательности, полученные посредством линейного преобразования синдромов, подаются на входы порогового элемента, где путем “голосования” (мажоритарный метод) и сравнения его результатов с порогом выносится решение о значении декодируемого символа. Основное достоинство этого метода декодирования – простота реализации. Однако он не полностью реализует потенциальные корректирующие способности сверточного кода. Кроме того, не все сверточные коды могут быть декодированы этим методом. Пороговое декодирование, как правило, применяется для систематических кодов. [1]. Общая схема декодера для сверточного кода (R = 1/2) представлена на рисунке 3.70.

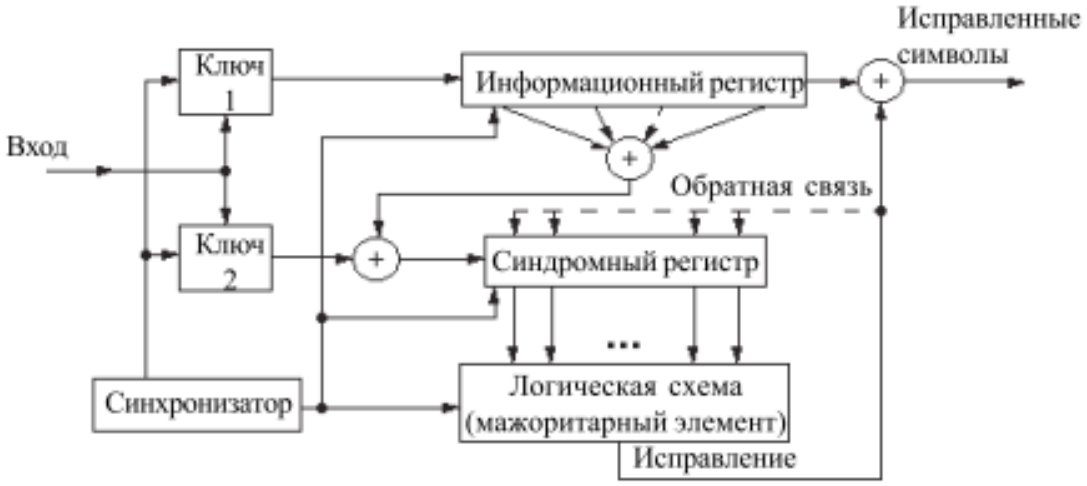


Рис. 3.70. Общая схема декодера для сверточного кода

Декодер содержит аналог кодера, в котором по принимаемым информационным символам в сдвигающем регистре формируется копия проверочной последовательности. С этой целью синхронизатор декодера с помощью ключей 1 и 2 “расфасовывает” входную последовательность символов на 2 потока – информационный и проверочный, синхронизатор управляет работой всего декодера.

В формирователе синдрома (сумматоре по модулю 2) образуется последовательность синдромов S, которая поступает на вход синдромного регистра. В отсутствие в канале ошибок последовательности на входах формирователя синдрома всегда совпадают, и синдромная последовательность состоит из одних нулей. Различным наборам ошибок соответствуют определенные конфигурации синдромных последовательностей, в которых на определенных позициях появляются единичные символы. Закон формирования проверочных символов выбирается таким образом, чтобы по структуре синдромной последовательности можно было определить искаженные символы.

Логическая схема определяет по синдрому правильность записанного в информационном регистре блока информационных символов. Если имеется комбинация ошибок, которая может быть исправлена, то логическая схема исправляет ошибки в этом блоке путем подачи единичных символов на выходную схему суммирования по модулю 2 в моменты выхода из информационного регистра искаженных символов.

Ошибки, исправляемые в очередном информационном блоке, могут влиять на символы синдромов, соответствующих последующим блокам, поскольку сверточные коды непрерывны. Для того чтобы декодер смог полностью реализовать свои корректирующие возможности, следует исключить влияние этих ошибок. Это может быть достигнуто за счет обратной связи, которая на схеме (рисунок 3.62) представлена пунктирной линией.

Обратная связь преобразует синдромный регистр прямого действия в нелинейный регистр сдвига с обратной связью. Это может привести к явлению, называемому

размножением ошибок. Неисправимые ошибки в канале могут вызвать переход синдромного регистра в такое состояние, что и при отсутствии аддитивных ошибок в канале декодер будет продолжать всегда декодировать неправильно. Причина этого состоит в том, что выход нелинейного регистра сдвига с обратной связью, когда на его вход поступает нулевая последовательность, а начальное состояние – ненулевое, может быть периодическим [6].

Последовательный алгоритм декодирования

При рассмотрении алгоритмов последовательного декодирования удобно представлять сверточный код в виде кодового дерева. Как уже отмечалось ранее, исходному нулевому состоянию сдвигающего регистра кодера соответствует начальный узел дерева. Если входной информационный символ, поступающий в регистр равен 1, то ему приписывается линия (ребро дерева), идущая, как принято на этом рисунке, вниз, а если информационный символ равен 0, – то вверх. Тем самым получаем два новых узла, соответствующие следующему такту работы кодера, для каждого из которых дерево строится далее аналогичным образом, и так далее. Над каждым ребром дерева записываются кодовые символы, получаемые при этом на выходе кодера. Совокупность нескольких последовательных ребер, соединяющих какие-либо два узла, составляет ветвь дерева.

Каждая последовательность кодируемых информационных символов порождает определенный путь по кодовому дереву. Очевидно, задача декодера заключается в отыскании истинного (правильного) пути, то есть того пути, который в действительности был порожден кодером [1].

Таким образом, при алгоритмах последовательного декодирования декодер определяет наиболее правдоподобный путь по решетке, что позволяет исключить из анализа большую часть остальных путей, имеющих меньшее правдоподобие.

Для этого, сначала необходимо передвигаться поочередно вдоль каждого пути кодовой решетки, сравнивая принятую последовательность со значением, соответствующим пути, по которому происходит движение. Если при этом удается обнаружить некоторый путь, значение которого, совпадает с принятой последовательностью, то естественно считать, что эта последовательность и передавалась.

Если быть точнее, происходит сравнение принятой комбинации с комбинациями возможными для данного шага декодирования. Путь считается наиболее правдоподобным, если метрика между принятой последовательностью и последовательностью соответствующей данному переходу минимальна.

Резюмируем: на каждом шаге декодирования мы имеем два исходящих из узла пути. Сначала мы вычисляем расстояние Хемминга между принятой комбинацией и комбинацией соответствующей одному исходящему пути, затем вычисляем расстояние Хемминга между

принятой комбинацией и комбинацией соответствующей второму пути. После чего выбираем тот путь, значение расстояния Хемминга которого меньше. Данный процесс показан на рисунке 3.71. В данном случае переход вниз по решетке будет считаться наиболее правдоподобным, так как расстояние Хемминга между значением соответствующему этому переходу и принятой комбинацией является наименьшим.

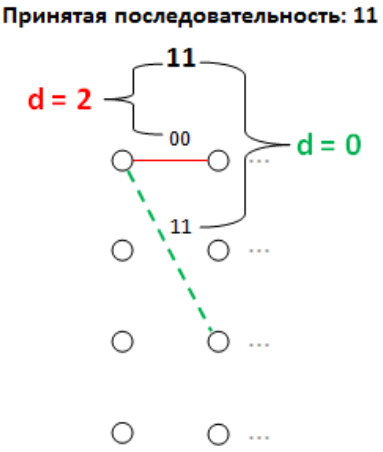


Рис. 3.71. Выбор пути с наименьшим расстоянием Хэмминга

Декодер выполняет операцию выбора пути с наименьшей метрикой до тех пор пока не столкнётся с неоднозначностью, при которой принять решение о том какой путь является кратчайшим не представляется возможным из-за того что расстояния Хемминга между принятой последовательностью и значениями двух возможных и путей одинаковы. В этой ситуации декодер вынужден проверить оба возможных пути, и по следующим метрикам принять решение касаемо неопределённости.

В нашем же случае, если расстояния Хемминга для обоих путей одинаковы, то прибегаем к жесткому методу принятия решений и пользуемся следующим правилом:

- Если мы находимся в узле 00 или 11, и пришла комбинация 00 или 01 – считаем что передавался ноль, 10 или 11 – единица.
- Если мы находимся в узле 01 или 10, и пришла комбинация 00 или 01 –считаем что передавалась единица, 10 или 11 – ноль.

Аналогичным образом декодер продолжает углубляться в решетку и принимать решения, касающиеся информационных битов, устраняя все пути, кроме одного.

На рисунке 3.72 изображена ситуация когда на вход декодера поступает последовательность, в которой биты выделенные цветом заведомо искажены так, как это бывает в реальных каналах связи.

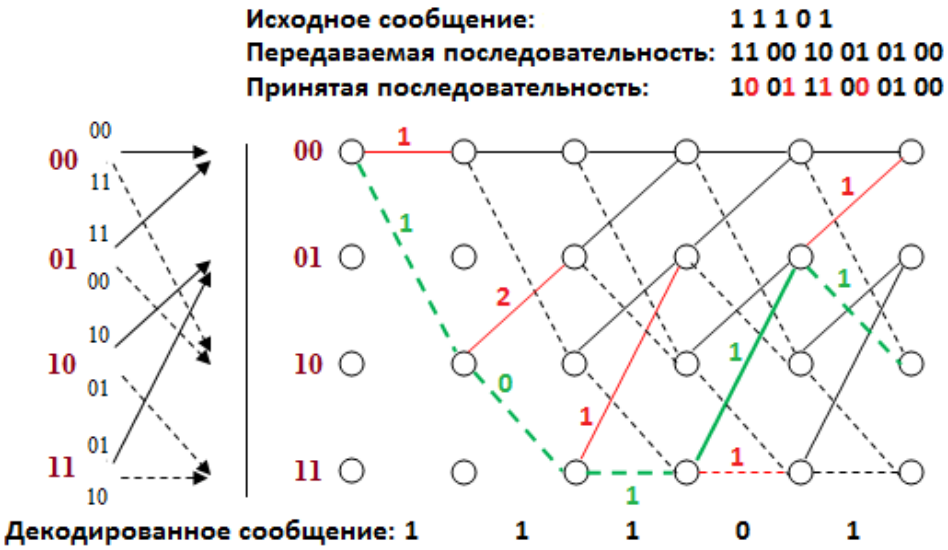


Рис. 3.72. Решетчатая диаграмма последовательного декодера

Сравнение алгоритма Витерби с последовательным алгоритмом

Принципиальное отличие алгоритма Витерби от последовательного алгоритма декодирования заключается в том, что алгоритм Витерби принимает решение по суммарным метрикам путей, причём в узлах, где эти пути сходятся. Этот шаг в процессе декодирования показан на рисунке 3.65. (метрики обозначены данным образом для наглядности).

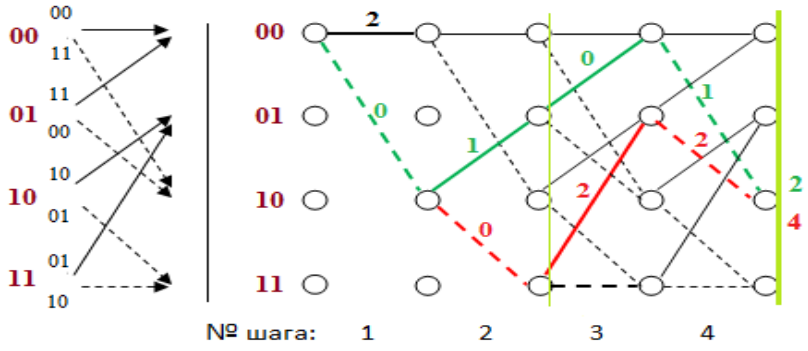


Рис. 3.73. Решетчатая диаграмма декодера Витерби

Таким образом, если бы в ситуации, представленной на рисунке 3.64, мы принимали решение на основе последовательного алгоритма, то на первом и втором шаге декодирования мы приняли бы решение идти вниз по решетке, выбирая кратчайшие метрики Хемминга. Однако в таком случае мы бы не учли метрики последующих ветвей нашего, на первый взгляд кратчайшего пути, которые на шаге три и четыре сильно возрастают, а значит, выбрали бы не самый короткий путь при декодировании. Поэтому, алгоритм Витерби является оптимальным. Однако он сложнее последовательного как в понимании, так и в реализации. В качестве метрики выступает расстояние Хемминга.

Программная реализация сверточного кодера

Подготовка данных к кодированию

Сначала, символы, введенные с клавиатуры ассоциируются со своими десятичными значениями в таблице ASCII. Затем, ASCII-код символа переводится в восьмеричную систему исчисления.

Ранее был объявлен массив «Bit_Array()», который имеет 8 ячеек, в которых содержится двоичное представление чисел от 0 до 7. Так как восьмеричная система имеет всего восемь цифр (0-7) то перебирается массив в цикле от 0 до 7, и находится равенство между индексом массива (счетчиком цикла) и восьмеричной цифрой ASCII-кода, после чего вместо нее подставляется двоичное представление из ячейки массива.

К примеру, цифра 0, будет иметь десятичный ASCII код - 48, в восьмеричный - 60. Что в двоичной системе исчисления будет 110000. Если ASCII-код - двузначное число, впереди дописывается 00, чтобы двоичные числа имели одинаковую длину. Формируется строка бит сообщения 00110000.

После того как символы исходного текста ассоциированы с кодами таблицы кодов ASCII и переведены в двоичный вид, осуществляется свёрточное кодирование.

Функция свёрточного кодирования

Разрабатываемый свёрточный кодер является несистематическим, без обратной связи и изображен на рисунке 3.74. Функциональную схему кодера (см. Приложении Б).

Кодер называется систематическим, если в каждом кодовом подблоке длины c имеется b символов, которые совпадают с символами соответствующего информационного подблока. В противном случае кодер - несистематический.

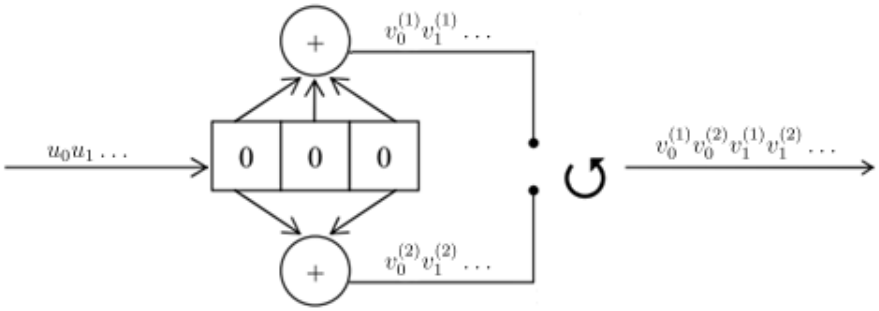


Рис. 3.74. Двоичный сверточный кодер скорости R = 1/2.

Бесконечная последовательность информационных символов $u = u\ 0\ u\ 1\ \ldots$.поступает в регистр сдвига (в нашем примере памяти m = 3, то есть с двумя элементами задержки). Кодер генерирует две выходные последовательности $v^{(1)} = v_0{}^{(1)}\ v_1{}^{(1)}$... и $v^{(2)} = v0^{(2)} v_1{}^{(2)}$.., которые поступают в коммутатор, формирующий выходную (кодовую) последовательность $v = v_0{}^{(1)}\ v_0{}^{(2)}\ v_1{}^{(1)}\ v_1{}^{(2)}$.., передаваемую по каналу. Поскольку на каждый

входящий в кодер информационный символ приходится два кодовых символа, кодовая скорость равна R = 1/2. Если одна из кодовых последовательностей $v^{(1)}$ или $v^{(2)}$ совпадает с информационной последовательностью u, то, как отмечалось ранее, кодер называется систематическим. В противном случае кодер называется несистематическим [9].

Генераторы свёрточного кодера изображенного на рисунке 3.75. можно описать при помощи полиномов:

$G_1 = 1+X+X^2$

$G_2 = 1+X^2$

Таким образом, порождающая матрица этого сверточного кодера , в терминах оператора задержки имеет вид:

$G(X) = (1+X+X^2\ 1+X^2)$.

Как уже отмечалось ранее, кодер – это автомат. В данном случае имеется классический алгоритм автомата Мили — конечный автомат, выходная последовательность которого (в отличие от автомата Мура) зависит от состояния автомата и входных сигналов. Это означает, что в графе состояний каждому ребру соответствует некоторое значение (выходной символ). В вершины графа автомата Мили записываются выходящие сигналы, а дугам графа приписывают условие перехода из одного состояния в другое, а также входящие сигналы. Диаграмма состояний свёрточного кодера представленного на рисунке 3.66 изображена на рисунке 3.75.

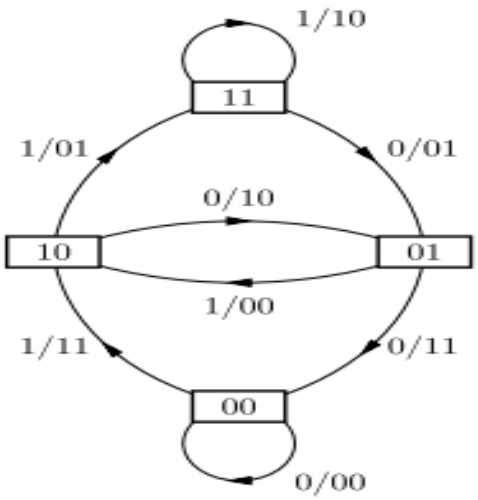


Рис. 3.75. Диаграмма состояний сверточного кодера скорости R = 1/2

Свёрточное кодирование кодером со скоростью R= 1/2 в программном комплексе выполняет функция Fun_BitCoder. После объявления массива и его целочисленных переменных обнуляются регистры сдвига, описывается цикл, осуществляющий побитное чтение информационных символов, их сдвиг в регистрах и суммирование по модулю два

согласно установленным правилам. Элемент исходного кода программы, отвечающий за реализацию свёрточного кодирования, представлен на рисунке 3.76.

```
            R1 = 0
            R2 = 0
        For i = 0 To length_array 'цикл обработки сообщения автоматом Мили
            X = S_mess_chr(i).ToString
            Y1 = X Xor R1 'сумма по модулю 2(исключающее или)
            Y2 = X Xor R2 '
            R2 = R1 'сдвиг битов в регистрах
            R1 = X '
            Y1 = Y1 Xor R2 'сумма по модулю 2(исключающее или)
            s_res = s_res + Y1.ToString + Y2.ToString 'результат кодирования
        Next
        Return s_res
    End Function
```

Рис. 3.76. Элемент программного кода функции свёрточного кодирования

В роли сумматоров выступают переменные Y1 и Y2, а в роли ячеек регистра переменные R1 и R2. Коммутатором является переменная s_res. Для осуществления поочередной коммутации бит с выходов кодера символы переводятся в строковый тип.

Функция имитации канала связи

По мере прохождения по каналу связи сигнал подвергается воздействию помех, именно этот процесс имитирует данная функция. *Помехи* - случайные воздействия, искажающие передаваемый сигнал. Воздействие помехи на сигнал может быть двояким. Если помеха $\xi(t)$ складывается с сигналом $s(t)$ и на вход приёмника действует их сумма $x(t) = s(t) + \xi(t)$ такую помеху называют *аддитивной*. Если результирующий сигнал равен произведению помехи и передаваемого сигнала $x(t) = s(t) \cdot \xi(t)$, то помеху называют *мультипликативной*.

В канале с *мультипликативной* помехой наблюдаются всплески искажений, причинами которых могут быть, например, коммутационные помехи, быстрые замирания радиосигнала. Выражается *мультипликативная* помеха в изменении характеристик линии связи (сопротивление линии связи (ЛС), частота среза ЛС, нелинейность характеристик ЛС).

Чтобы приблизить модель симметричного канала к действительности, вводится понятие *пакета ошибок*.

Пакетом ошибок длины L называется вектор ошибок E, все ненулевые компоненты которого расположены на отрезке из подряд следующих позиций, причем в начале и конце отрезка расположены ненулевые компоненты.

Аддитивная помеха не зависит от ЛС и определяется внешними воздействиями на среду передачи сигналов. В качестве модели аддитивной помехи обычно принимают белый шум, то есть стационарный гауссовский случайный процесс, имеющий нулевое математическое

ожидание и равномерный и бесконечно широкий спектр плотности мощности. При воздействии помех на передаваемый сигнал, имеющих аддитивный характер, L=1 имеем пакет длиной в один символ, то есть однократную ошибку.

В разрабатываемом программном комплексе имеется возможность пронаблюдать эффективность работы последовательного декодера свёрточных кодов как при наличии однократных ошибок, так и пачек длиною L > 1.

На рисунке 3.77 приведена блок-схема программно реализованного алгоритма имитации канала связи с белым шумом и канала с многолучевым распространением.

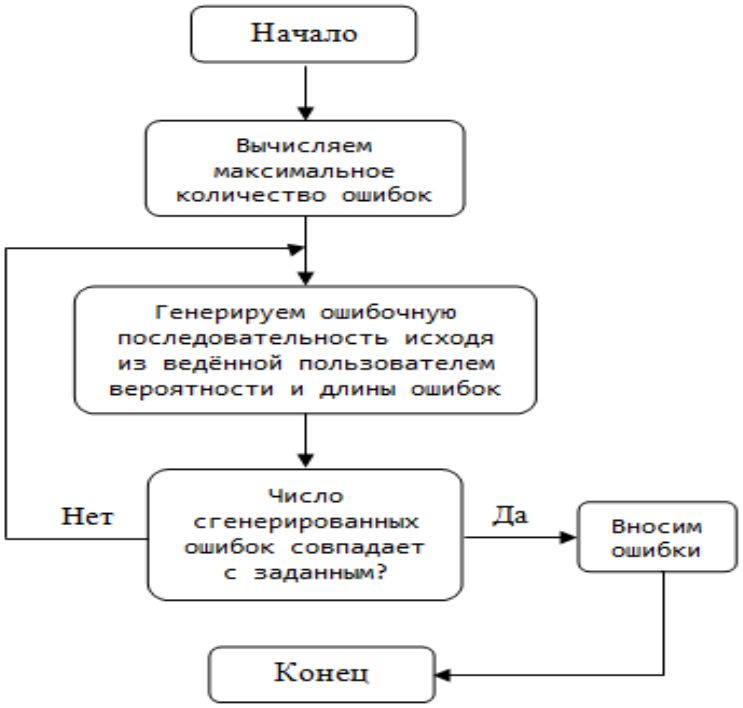


Рис. 3.77. Блок-схема для алгоритма имитации канала связи

За имитацию канала связи отвечает функция «Fun_SimulatorChannelCom». В ней, первым делом, определяется максимальное количество генерируемых ошибок:

max_mistake_dou = length_array * (CInt(probability_mistake) / 100),

где: max_mistake_dou – максимальное количество ошибок

length_array – длина сообщения

probability_mistake – вероятность ошибки

Например, если длина сообщения равна 80 бит, а битовая вероятность ошибки – 0.1 то генератор сгенерирует от 0 до 8 ошибок.

Далее, активируется обычный генератор случайных чисел в диапазоне от 1 до 100: Dim value As Integer = CInt(Int((100 * Rnd()) + 1))

Функция CInt приводит выражение в тип Integer и округляет его до ближайшего целого. После чего число сгенерированных ошибок сверяется с заданным числом ошибок,

если ошибок меньше чем задано, цикл продолжается. Проверяется также заданная вероятность с сгенерированным числом, если оно меньше или равно, то в сообщение вносится ошибка. Внесение ошибки осуществляется путем инвертирования 0 и 1. Элемент исходного кода программы, отвечающий за реализацию функции имитации канала связи представлен на рисунке 3.79.

```
If (max_mistake_int > 0) And (i <= (length_array_1 - 1)) And (remainder > 0) Then ' пока
кол-во введеных ошибок не достигло максимума
                If (value <= CInt(probability_mistake)) Then 'если случайное число или
равно, заданной вероятности ошибки
                    j = 1  ' выставляем флаг ошибки

If (S_mess_chr(f).ToString = "0") Then    ' заменяем биты в сообщении
                        S_mess_chr(f) = "1"
                    Else
                        S_mess_chr(f) = "0"
                    End If
```

Рис. 3.78. Элемент программного кода функции имитации канала связи

Если были заданы двойные ошибки, инвертируются по два соседних бита.

В конечном итоге, после того как подсчёт ошибок осуществлен, и их количество и кратность соответствует заданным параметрам, сообщение с ошибками выводится на блок индикации, где искаженные биты выделяются цветом.

Функция вычисления метрик путей

Важным этапом разработки алгоритма декодирования является выбор алгоритма вычисления метрик путей. В качестве такой метрики будет принято расстояние Хэмминга. Расстояние Хэмминга — число позиций, в которых соответствующие символы двух слов одинаковой длины различны. Согласно выбранному алгоритму вычисления метрик путей, биты сообщения суммируются по модулю два со значениями возможных путей и полученному результату подставляется его десятичное значение, соответствующее величине расстояния Хемминга между слагаемыми. За это отвечает функция «Fun_Metric», программная реализация которой приведена на рисунке 3.70, где:

s_1 – биты сообщения;

s_2 – возможные значения путей решетки;

res – результат суммирования по модулю два бит сообщения со значениями возможных путей решетки;

res_1 - десятичное значение, соответствующее величине расстояния Хемминга между слагаемыми.

```
Public Function Fun_Metric(ByVal f_s_mess_1 As String, ByVal f_s_mess_2 As String)
        Dim s_1, s_2 As Integer
        Dim res, res_1 As Integer
        s_1 = CInt(f_s_mess_1)
        s_2 = CInt(f_s_mess_2)
        res = s_1 Xor s_2 ' сумма по модулю два
        Select Case res
            Case "00"
                res_1 = 0
            Case "01"
                res_1 = 1
            Case "10"
                res_1 = 1
            Case "11"
                res_1 = 2
        End Select
        Return res_1
    End Function
```

Рис. 3.79. Программный код функции вычисления метрик путей

Из таблицы 3.4 видно, что значение 00 может стать результатом суммирования только двух слов, биты которых не отличаются ни в одной из позиций, а это значит, что расстояние Хемминга между такими словами рано 0. Значения res = 01 и res = 10 возможны лишь в том случае, если по модулю 2 суммировались слова, соответствующие биты которых отличаются друг от друга в одной позиции, то есть расстояние Хемминга между ними равно 1. Что же касается значения res = 11, то оно может получиться только в результате сложения двух слов, биты которых в обеих позициях отличаются, и расстояние Хемминга между этими словами равно 2.

Таблица 3.4. Вычисление метрик путей

№ такта	Входной информационный бит	Состояние регистра сдвига	Сумматор 1	Сумматор 2	Выходные кодовые комбинации
0	-	0000			-
1	1	1000	1⊕0⊕0=1	1⊕ 0⊕0=1	11
2	0	0100	0⊕0⊕0=0	0⊕ 1⊕0=1	01
3	1	1010	1⊕1⊕0=0	1⊕ 0⊕0=1	01
4	1	1101	1⊕0⊕1=0	1⊕	01

				1⊕1=1	
5	0	0110	0⊕1⊕0=1	0⊕ 1⊕0=1	11
6	1	1011	1⊕1⊕1=1	1⊕ 0⊕1=0	10

Функция декодирования последовательным алгоритмом

Учитывая топологию решетки, первый шаг декодирования был реализован отдельно. В элементе исходного кода приведенного на рисунке 3.71 происходит проверка и выбор возможного пути, зависимости от битов сообщения (проверяются биты массива S_str_Array()).

Как мы уже знаем из теории, на первом шаге декодирования состояние декодера – 00, возможны только два пути, к узлам 00 и 10, которымсоответствуют кодовые символы с выхода кодера 00 и 11.

Однако на вход декодера могут придти целых четыре комбинации бит сообщения, поэтому, на первом шаге декодирования, было решено связать приход кодовой комбинации 00 и 01 к переходу в узел 00, а кодовые комбинации 10 и 11 к переходу в узел 10. Так мы избавляемся от неопределённостей, заставляя декодер принять решение, даже если пришла не разрешенная комбинация.

```
If ((S_str_Array(i) = "00") Or (S_str_Array(i) = "01")) Then
    state_decoder = "00"
    state_metric = Fun_Metric(state_decoder, S_str_Array(i))
    s_mess = "0"
    f_x1 = Arr_XY(0, f)
    f_y1 = 284

...


ElseIf ((S_str_Array(i) = "10") Or (S_str_Array(i) = "11")) Then

    state_decoder = "10"
    state_metric = Fun_Metric(state_decoder, S_str_Array(i))
    s_mess = "1"
    f_x1 = Arr_XY(2, f)
    f_y1 = 384

...

End If
GoTo End_1_step
```

Рис. 3.80. Элемент программного кода функции декодирования последовательным алгоритмом

После того как путь выбран, в переменную state_metric записывается значение метрики пути, вычисленное описанной функцией Fun_Metric. Далее ставится метка 0 или 1,

указывающая на то какой символ передавался, после чего задаются конечные координаты по горизонтали и вертикали отрезка для выбранного пути.

На этом заканчивается первый шаг декодирования и осуществляется безусловный переход на метку End_1_step. Сюда же переходит программа и после отработки всех остальных шагов.

Далее происходит аналогичные операции, только количество возможных путей увеличивается до 4-х (00, 01, 10, 11), осуществляется переход ко второму шагу декодирования. Граф автомата, каковым является наш декодер, принимает вид показанный на рисунке 3.82

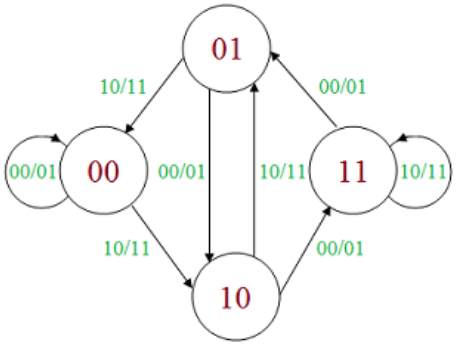


Рис. 3.81. Граф реализуемого последовательного декодера

Стратегия представленная на рисунке 3.82 реализуется при помощи алгоритма подобному тому, что был использован на первом шаге декодирования, теперь декодер так же выбирает путь с кратчайшей метрикой, однако в случае если расстояния Хемминга для обоих путей одинаковы, то прибегает жесткому методу принятия решений:

- Если мы находимся в узле 00 или 11, и пришла комбинация 00 или 01 – считаем что передавался ноль (двигаемся вверх), 10 или 11 – передавалась единица (двигаемся вниз).
- Если мы находимся в узле 01 или 10, и пришла комбинация 00 или 01 –считаем что передавалась единица (двигаемся вниз)., 10 или 11 – передавался ноль (двигаемся вверх).

Процесс происходит до тех пор пока не закончится сообщение. Полный исходный код функции декодирования приведен в приложении (см. Приложение И).

Подготовка к выводу результатов декодирования

Для того чтобы в конечном итоге получить символьный результат организуем цикл реверсирования массива символов сообщения, иначе мы получим строку в зеркальном отражении. Организуем цикл перебора массива символов сообщения, разделяем их на пачки по 8 бит и каждую пачку преобразуем в десятичное число по принципу: $2^7 + 2^6 + 2^5 + 2^4 +$

$2^3+ 2^2+ 2^1+ 2^0$ а затем получаем символ из его ASCII-кода. Повторяем операции пока не закончится сообщение, после чего результаты выводятся в соответствующих окнах.

Сверточное декодирование. Практическая часть

ЗАДАНИЕ №1

Вариант 1

Сверточный код: $g_1(X) = 1+ X^2 + X^3$, $g_2(X) = 1 + X + X^3$.

Входная последовательность: V = 10110.

На рисунке 3.73 представлен сверточный кодер.

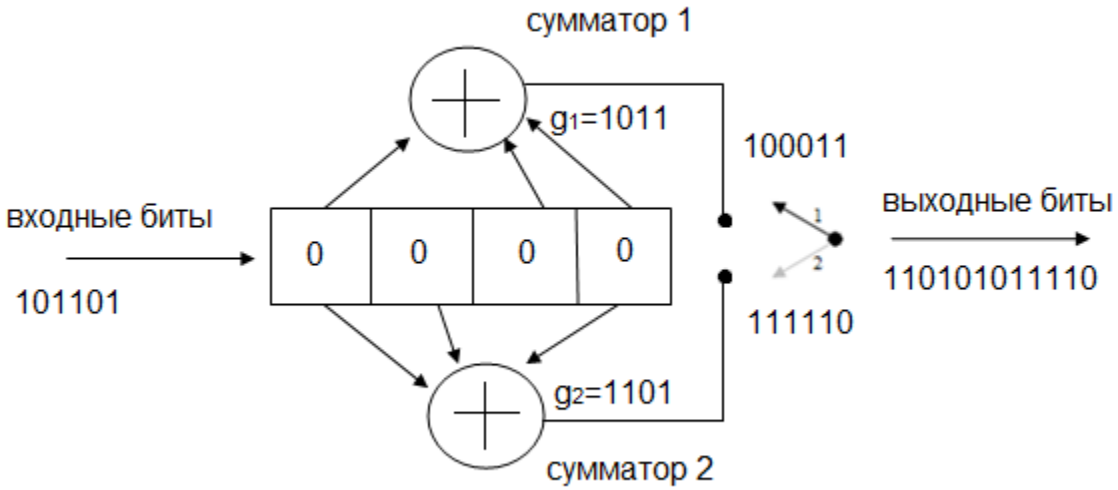


Рис. 3.82. Свёрточный кодер

Произвели кодирование входной последовательности V данным свёрточным кодером и нарисовали таблицу кодирования информационного потока. Данные представлены в таблице 3.5.

Таблица 3.5. Процесс кодирования информационного потока

№ такта	Входной информационный бит	Состояние регистра сдвига	Сумматор 1	Сумматор 2	Выходные кодовые комбинации
0	-	0000			-
1	1	1000	1⊕0⊕0=1	1⊕ 0⊕0=1	11
2	0	0100	0⊕0⊕0=0	0⊕	01

				1⊕0=1	
3	1	1010	1⊕1⊕0=0	1⊕ 0⊕0=1	01
4	1	1101	1⊕0⊕1=0	1⊕ 1⊕1=1	01
5	0	0110	0⊕1⊕0=1	0⊕ 1⊕0=1	11
6	1	1011	1⊕1⊕1=1	1⊕ 0⊕1=0	10

Построили решетку кодера:

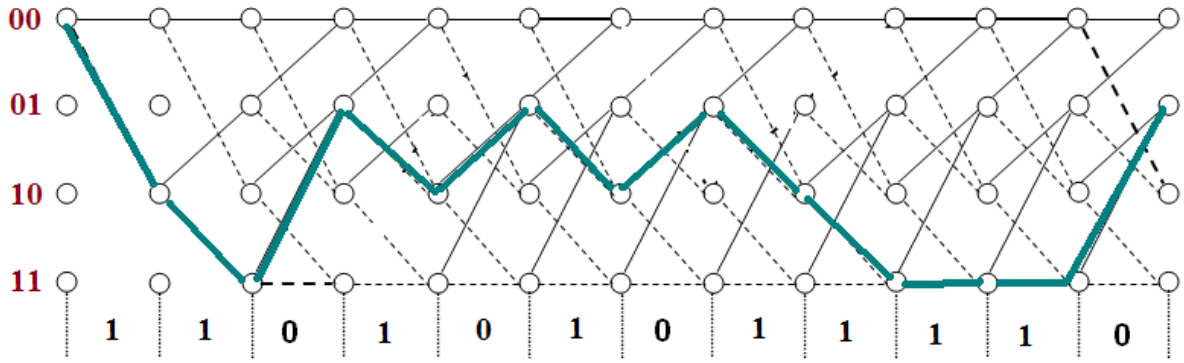


Рис. 3.83. Решетка кодера

Изобразили решетку декодера и произвели декодирование:

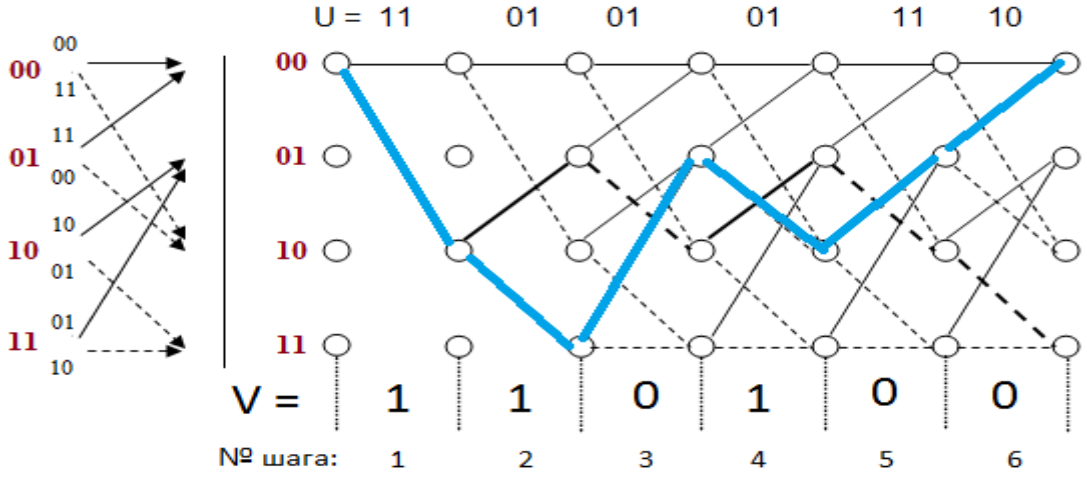


Рис. 3.84. Решетка декодера

В результате были выявлены 6 ошибок: После декодера был получен результат:

11 10 11 01 10 01. Подчеркиванием выделены места, где были совершены ошибки.

Вариант 2

Сверточный код: $g_1(X) = X + X^2 + X^3$, $g_2(X) = 1 + X + X^2$.

Входная последовательность: V = 010100.

На рисунке 3.85 представлен сверточный кодер.

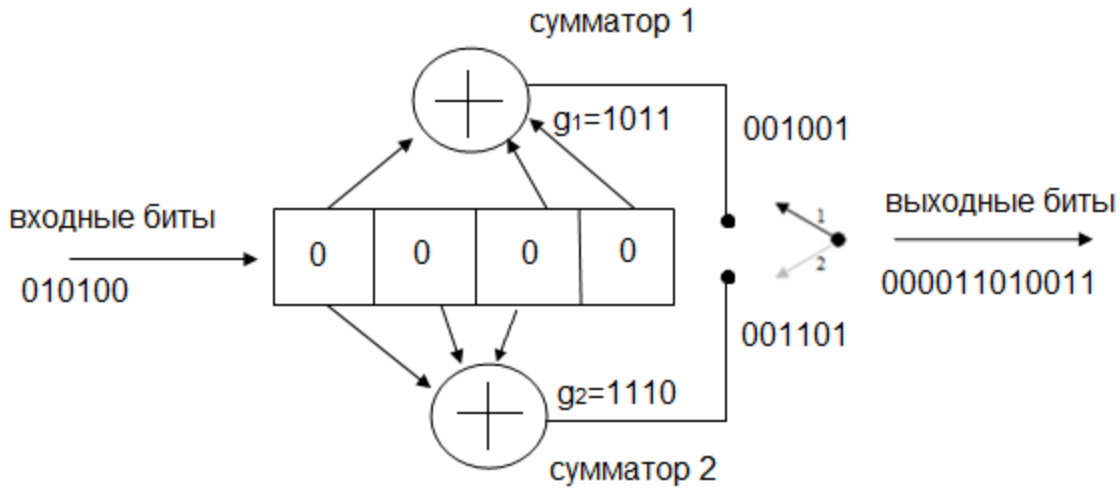


Рис. 3.85. Свёрточный кодер

Произвели кодирование входной последовательности V данным свёрточным кодером и нарисовали таблицу кодирования информационного потока. Данные представлены в таблице 3.6.

Таблица 3.6. Процесс кодирования информационного потока

№ такта	Входной информационный бит	Состояние регистра сдвига	Сумматор 1	Сумматор 2	Выходные кодовые комбинации
0	-	0000			-
1	0	0000	0⊕0⊕0=0	0⊕ 0⊕0=0	00
2	0	0000	0⊕0⊕0=0	0⊕ 0⊕0=0	00
3	1	1000	1⊕0⊕0=1	1⊕	11

				0⊕0=1	
4	0	0100	0⊕0⊕0=0	0⊕ 1⊕0=1	01
5	1	1010	1⊕1⊕0=0	1⊕ 0⊕1=0	00
6	0	0101	0⊕0⊕1=1	0⊕ 1⊕0=1	11

Построили решетку кодера:

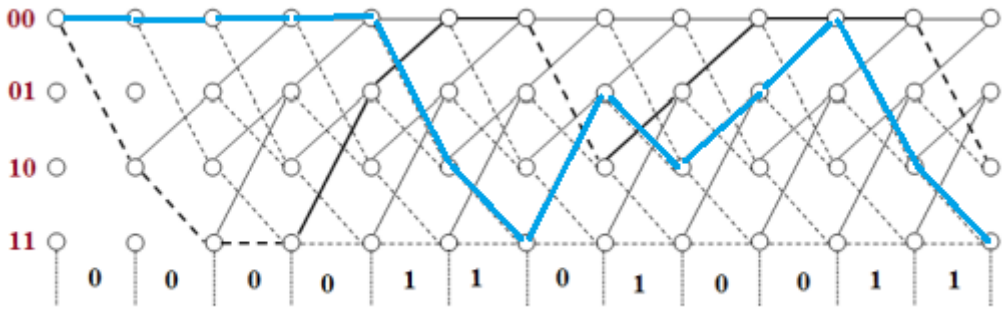


Рис. 3.86. Решетка кодера

Изобразили решетку декодера и произвели декодирование:

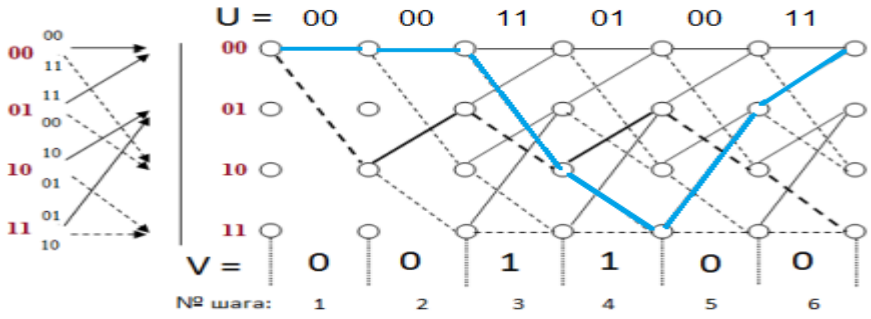


Рис. 3.87. Решетка декодера

В результате были выявлены 5 ошибок: После декодера был получен результат:

00 00 1<u>0</u> <u>1</u>1 0<u>1</u> <u>00</u>. Подчеркиванием выделены места, где были совершены ошибки.

ЗАДАНИЕ №2

Построить графики зависимости числа ошибок от вероятности ошибок для одиночной и двойной длины ошибок.

Рис. 3.88. Процесс кодирования и декодирования

График зависимости числа ошибок от вероятности ошибок для одиночной длины ошибок представлен на рисунке 3.89, а для двойной длины ошибок на рисунке 3.90.

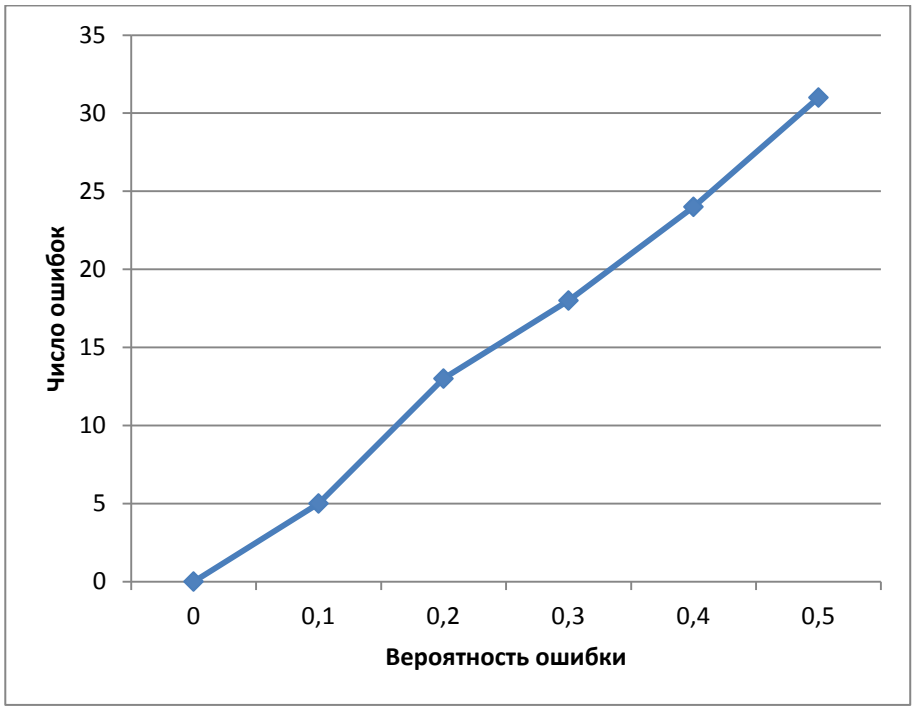


Рис. 3.89. График зависимости числа ошибок от вероятности ошибок для одиночной длины ошибок

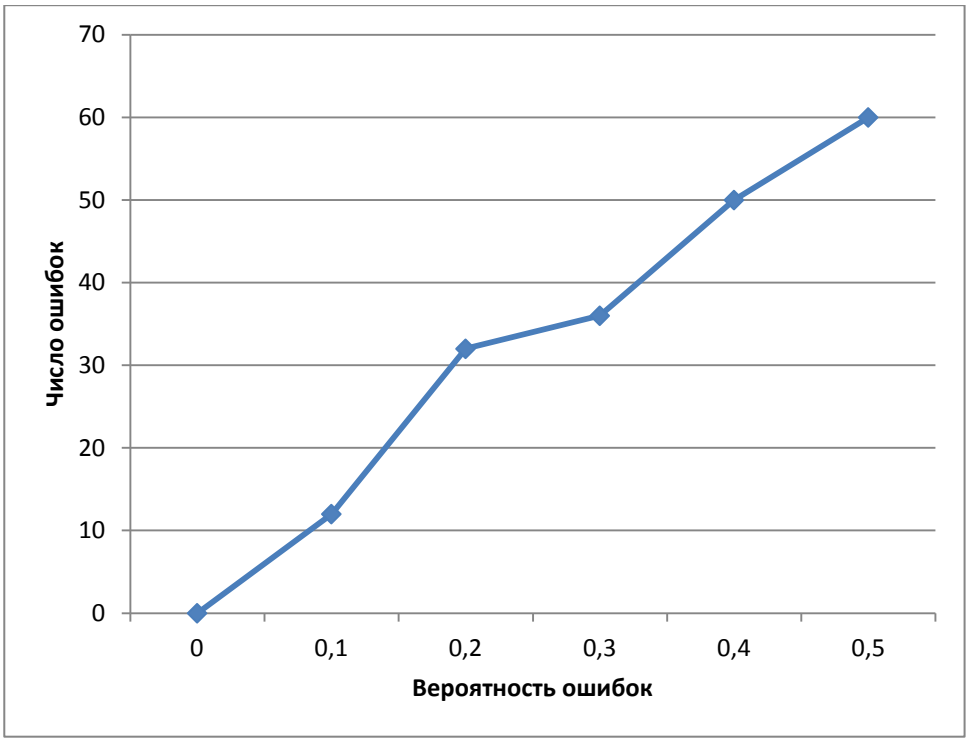


Рис. 3.90. График зависимости числа ошибок от вероятности ошибок для двойной длины ошибок

В процессе выполнения индивидуальной работы была осуществлена работа по созданию программного комплекса для визуализации и исследования метода свёрточного декодирования на основе последовательного алгоритма. Программный комплекс выполнен в в полном соответствии с техническим заданием, и отвечает всем заявленным в нём требованиям.

Были произведены лабораторные испытания программного комплекса, в ходе которых была отмечена высокая эффективность его использования в качестве макета наглядно демонстрирующего процессы свёрточного кодирования и последовательного алгоритма декодирования.

3.4. Декодирование сверточных кодов по методу Витерби с использованием ПО MATLAB

Алгоритм сверточного декодирования Витерби [10]

В 1967 году Витерби разработал и проанализировал алгоритм, в котором, по сути, реализуется декодирование, основанное на принципе максимального правдоподобия; однако в нем уменьшается вычислительная нагрузка за счет использования особенностей структуры конкретной решетки кода. Преимущество декодирования Витерби, по сравнению с декодированием по методу "грубой силы", заключается в том, что сложность декодера Витерби не является функцией количества символов в последовательности кодовых слов.

Алгоритм включает в себя вычисление *меры подобия* (или *расстояния),* между сигналом, полученным в момент времени *t,,* и всеми путями решетки, входящими в каждое состояние в момент времени *ti*. В алгоритме Витерби не рассматриваются те пути решетки, которые, согласно принципу максимального правдоподобия, заведомо не могут быть оптимальными. Если в одно и то же состояние входят два пути, выбирается тот, который имеет лучшую метрику; такой путь называется *выживающим*. Отбор выживающих путей выполняется для каждого состояния. Таким образом, декодер углубляется в решетку, принимая решения путем исключения менее вероятных путей. Предварительный отказ от маловероятных путей упрощает процесс декодирования. В 1969 году Омура (Omura) показал, что основу алгоритма Витерби составляет оценка максимума правдоподобия. Отметим, что задачу отбора оптимальных путей можно выразить как выбор кодового слова с *максимальной метрикой правдоподобия* или *минимальной метрикой расстояния.*

Алгоритм Витерби является достаточно мощным алгоритмом декодирования, при этом сложность его аппаратурной реализации невысока. В настоящее время алгоритм Витерби применяется во многих стандартах беспроводной связи, таких как IEEE 802.11a/g, WiMAX, DAB/DVB, WCDMA,GSM.

Наиболее подходящей элементной базой для аппаратной реализации декодера Витерби являются программируемые логические интегральные схемы (ПЛИС).

Работа декодера Витерби

Алгоритм Витерби находит широкое применение и реализует *поиск максимально правдоподобного пути* на кодовой решетке с отбрасыванием части наименее правдоподобных вариантов путей на каждом шаге декодирования.

Алгоритм Витерби характеризуется постоянством вычислительной работы, однако *сложность* декодера Витерби растет, как при всех переборных алгоритмах, *по экспоненциальному закону* от длины кодового ограничения сверточного кода. Поэтому алгоритм Витерби используется для декодирования коротких сверточныхкодов.

Алгоритм заключается в *повторении одного основного шага*. На каждой из последующих диаграмм рис. 3.91 этот шаг изображен подробно.

Метрика пути (МП) есть сумма метрик ветвей, образующих некоторый путь на решетчатой диаграмме. Путь конечной длины оканчивается в определенном состоянии. Метрика состояния (МС) равна метрике пути, который заканчивается в данномсостоянии.

Шаг декодирования состоит в обработке декодером принимаемых из канала данных в интервале между двумя соседнимиотсчетами.

На рисункепоказано развитие процесса декодирования символов СК со скоростью $R_{код}$ = 1/2 и длиной кодирующего регистра K = 3. На вход декодера поступают пары символов из канала: (…11,10,00,11,01...) (декодирование с жестким решением). Цифрами около ветвей обозначены метрики ветвей, цифры в кружках обозначают метрикисостояний.

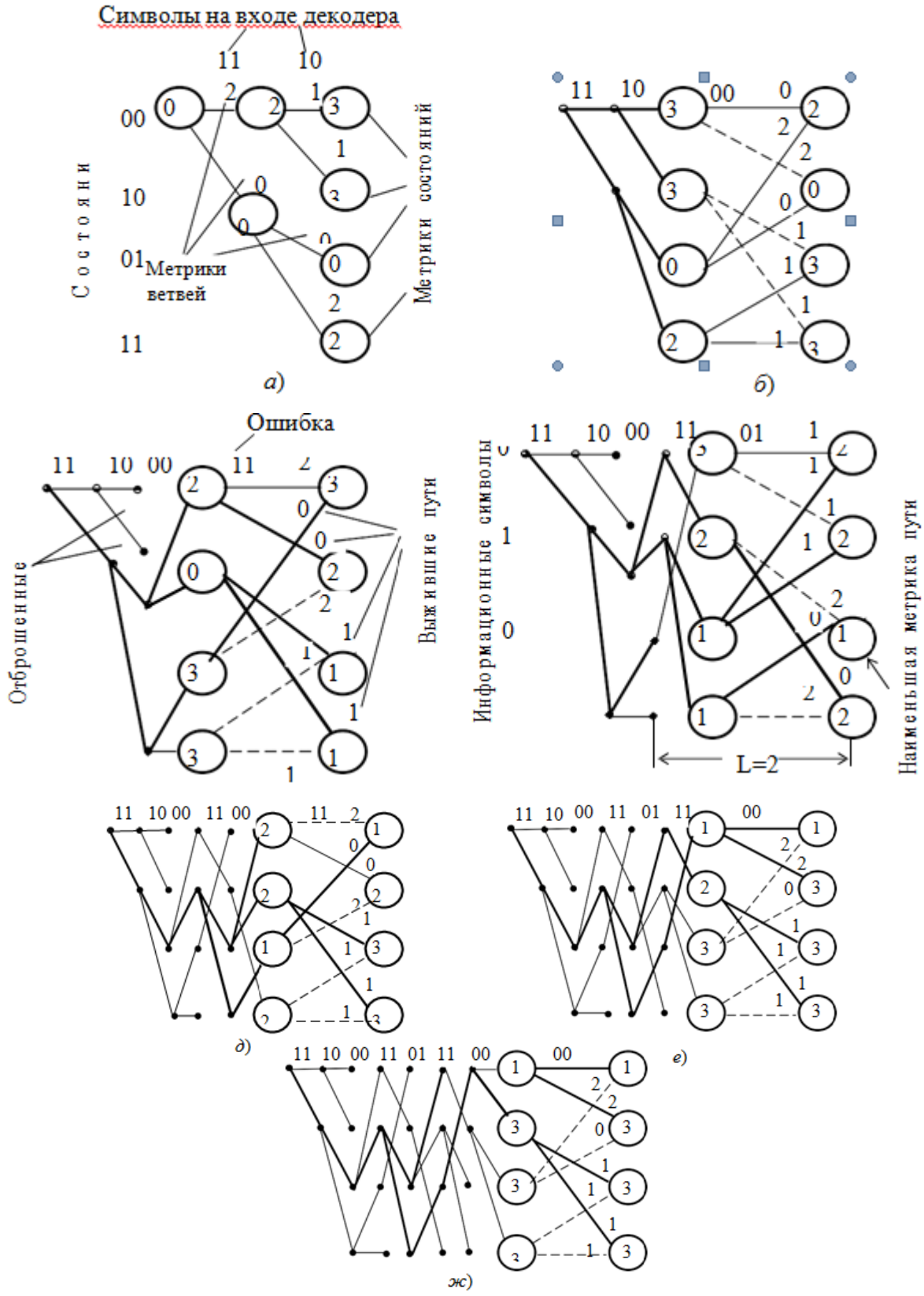


Рис. 3.91. Декодирование кода(7,5)

К каждому новому состоянию ведут два пути. К примеру, к состоянию 00 ведут пути из предыдущих состояний 00 и 01. На *i*-м шаге декодирования декодер вычисляет метрики путей как суммы метрик предыдущих состояний и метрик входящихветвей.

Далее производится попарное сравнение метрик путей, входящих в каждое из состояний (пары показаны фигурными скобками). В результате сравнения *выбирается меньшая метрика,* и она считается метрикой данного состояния для последующего шага декодирования. *Путь*, входящий в данное состояние с меньшей метрикой, считается *выжившим.* На рис. 3.91 отрезки выживших путей показаны сплошной линией. Пути, входящие в состояния с большими метриками, считаются отмершими (оборванными). Они показаны на решет-чатой диаграммепунктиром.

Таким образом, на каждом шаге декодирования в соответствии с алго- ритмом Витерби, *в каждом из состояний* решетчатой диаграммы производят-ся*однотипныеоперации*:

1) *Сложение метрик* предыдущих состояний с метриками соответст-вующихветвей.

2) *Сравнение метрик* входящих путей.

3) *Выбор путей* с наименьшими метриками, величины которых исполь-зуются как метрики состояний на последующем шаге декодирования. Если метрики сравниваемых путей одинаковы, то выбор одного из двух путей производится случайным образом.

На каждом шаге декодирования половина возможных продолжений путей отбрасывается. Другая половина образует продолжения путей для следующего шага декодирования, на котором вновь появляются два варианта продолжения каждого пути. Это обеспечивает *постоянство количества вычислений*, производимых на каждом шаге. *Декодер прослеживает* по кодовой решетке *путь*, имеющий *минимальное расстояние* от пути, который порождает кодер.

Таким образом, декодер, выбирающий на решетчатой диаграмме путь с наименьшей метрикой, *минимизирует вероятность ошибки*. Поскольку при декодировании анализу подвергаются последовательности конечной длины *L,* алгоритм не является строго оптимальным. Результаты расчетов и моделирова-ния показывают, что при соответствующем выборе величины $L > (6\ldots7)\nu$ мож- но получить результаты декодирования, достаточно близкие к оптимальным. *Сложность реализации алгоритма Витерби*для декодирования СК можно оценить по количеству ветвей кодовой решетки,

обрабатываемых декодером на длине декодирования *L,* с учетом сложности каждого шага решетки.

Декодер состоит из АЦП в каналах *X* и *Y*, вычислителя метрик ветвей, процессора, в котором производятся операции *сложения, сравнения ивыбора*,устройства памяти путей, которые выжили, и мажоритарного элемента МЭ, в котором выбирается путь с наибольшей метрикой. Оптимальное значение шага квантования зависит от отношения сигнал/шум на входе АЦП. При восьми уро-внях квантования минимум потерь обеспечивается при отношении размаха си- гнала к шагу квантования, равном (4,5...5,5).

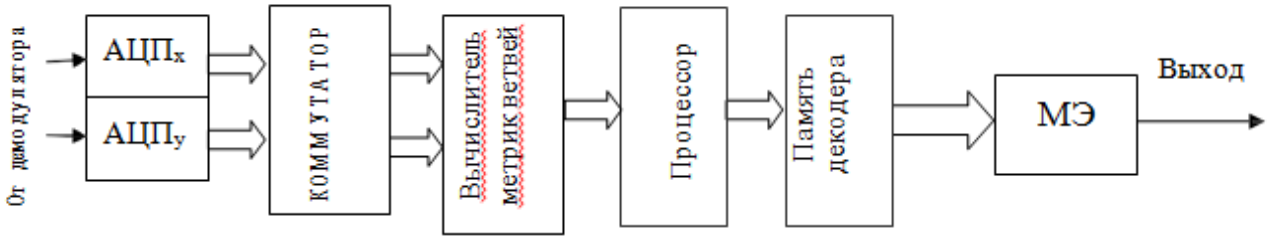


Рис. 3.92. Структурная схема декодера Витерби

Разработка программного интерфейса для исследования характеристик и визуализации основных преобразований.

На данном этапе необходимо реализовать и проанализировать описанный выше алгоритмы декодирования Витерби на практике в программной среде Matlab 2015. Необходимо построить соответствующие схемы, которые будут в себя включать источник сообщения, кодер, декодер, канал передачи, анализатор ошибок и соответственно устройства визуализации полученных результатов.

Индивидуальности сверточного кодировки. Библиотека Communications Blockset поддерживает разомкнутые либо бинарные замкнутые коды, которые могут быть описаны структурой сетки либо рядом порождающих полиномов. Это употребляет метод Viterbi, чтоб выполнить твердое и мягкое декодирование. Библиотека также включает декодер, максимизирующий апостериорную возможность, который может употребляться для мягкого декодирования сверточных кодов.

Характеристики сверточного кодировки. Процессы сверточного кодировки употребляют блоки Convolutional Encoder, Viterbi Decoder, и/либо APP Decoder из библиотеки Convolutional. Если параметр маски требуется и в кодирующем устройстве и в декодере, используйте одно и то же значение в обоих блоках. Блоки в подбиблиотеке Convolutional подразумевают, что используется один из 2-ух разных представлений сверточного кодера: если проектируется кодирующее устройство, используя диаграмму со

сдвиговыми регистрами и сумматорами по модулю 2 ,можно вычислить порождающую полиномиальную матрицу кода и потом применять функцию poly2trellis(из Communications Toolbox), чтоб произвести подобающую решетчатую (trellis) диаграмму маски параметра автоматом. Если проектируется кодирующее устройство, используя решетчатую диаграмму, можно выстроить решетчатую (trellis) диаграмму в MATLAB и применять его в качестве параметра маски.

Внедрение полиномиального описания в блоках. Для использования полиномиального описания в блоках Convolutional Encoder, Viterbi Decoder, либо APP Decoder используется сервисная функция poly2trellisиз Communications Toolbox. Эта функция воспринимает полиномиальное описание и преобразовывает это в решетчатое описание. К примеру, последующая команда вычисляет решетчатое описание кодирующего устройства, длина ограничения которого равна 5 и чьи порождающие полиномы 35 и 31: trellis = poly2trellis (5, [35 31])

Определение характеристик кодировки. Блоки Convolutional Encoder и Viterbi Decoder могут выполнить этот код, если у их характеристик установлены надлежащие значения. Длина ограничения кодирующего устройства - вектор длины 2, потому кодирующее устройство имеет два входа. Элементы этого вектора указывают на число сохраненных битов в каждом сдвиговом регистре, включая текущие входные биты. Подсчет ячеек памяти в каждом сдвиговом регистре на диаграмме и добавлении того для текущих входов приводят к длине ограничения, равного [5 4]. Определяя порождающую матрицу кода как матрицу размера 2 X 3 восьмеричных чисел, используйтся элемент в i – ой строке и j – ом столбце, чтоб указать, как i - ый вход повлияет на j – ый выход. К примеру, чтоб вычислить элемент во 2-ой строке и 3-ем столбце, заметьте, что последнее левое и два самых правых элемента во 2-м сдвиговом регистре диаграммы влияют на сумму, что сформировывает 3-ий выход. Эта информация в двоичном коде равна 1011, что эквивалентно восьмеричному числу 13. Полная порождающая матрица кода равна [27 33 0; 0 5 13]. Для использования длины кодового ограничения и характеристик порождающей матрицы кода в блоках Convolutional Encoder and Viterbi Decoder, примените функцию poly2trellis ,чтоб преобразовать эти характеристики в решеточную диаграмму.

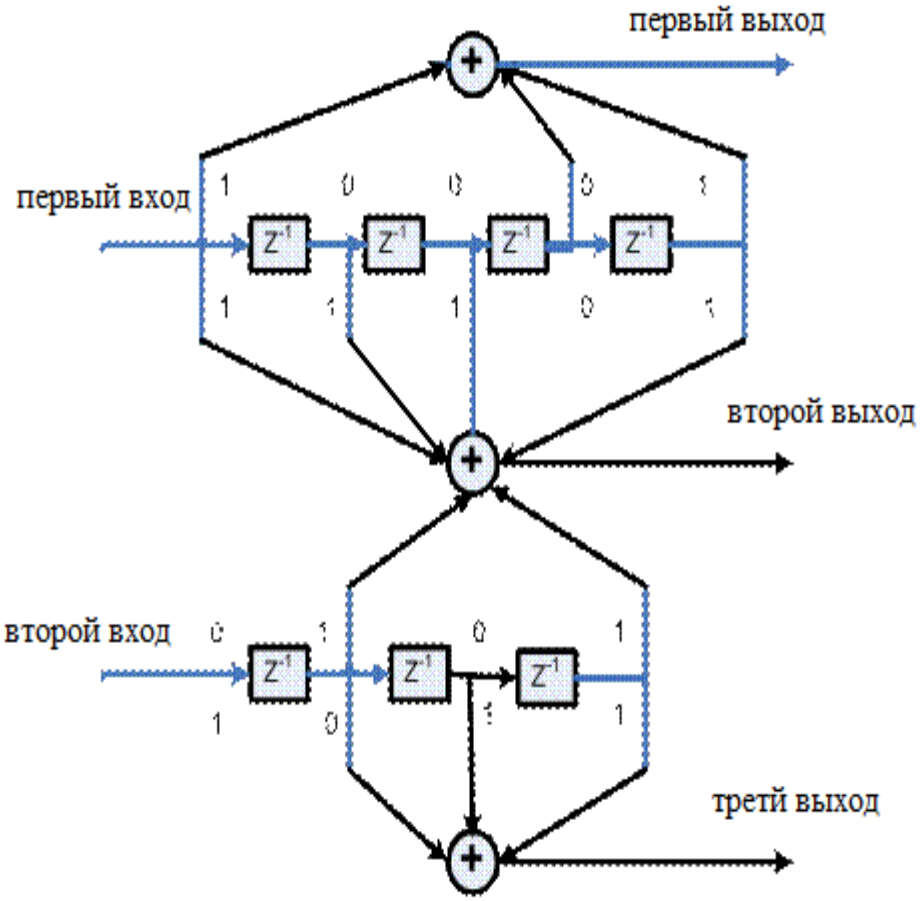


Рис. 3.93. Модель кодирующего устройства

Моделирование кодера Витерби

На рисунке представлена модель кодирующего устройства.

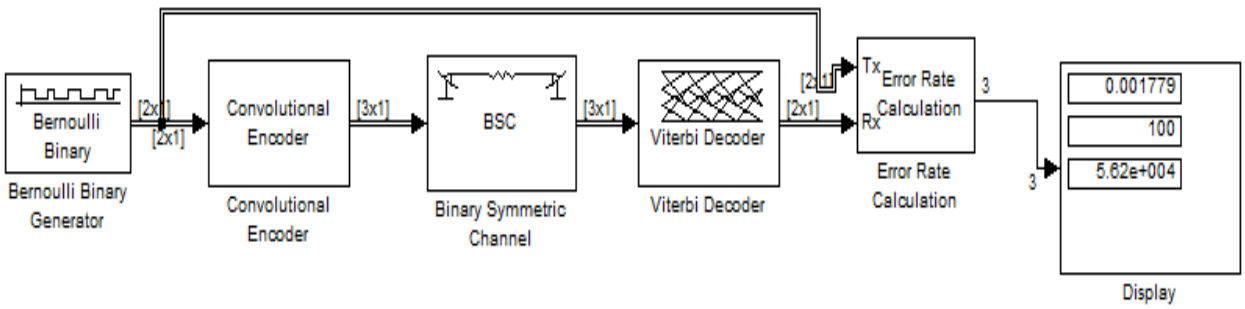
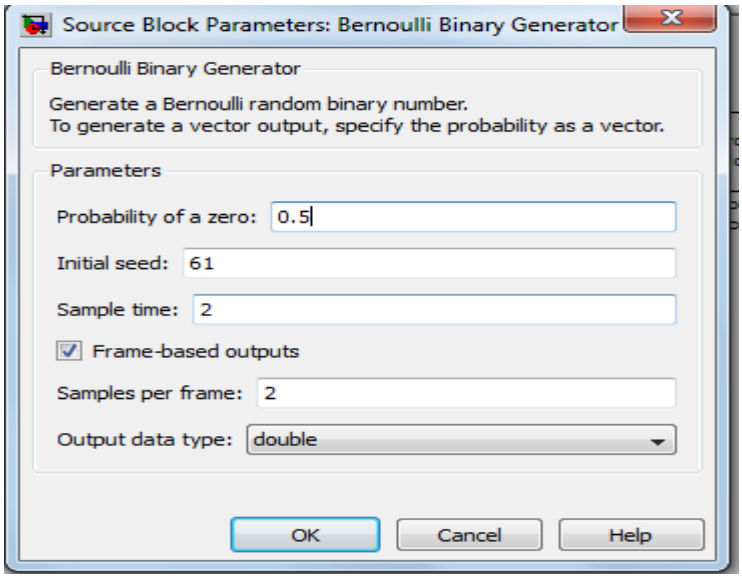


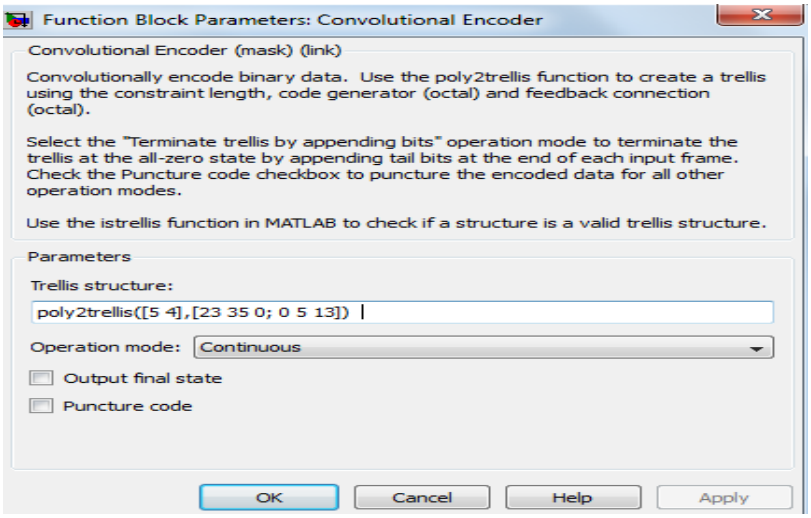
Рис. 3.94. Линия передачи с применением декодера Витерби

Чтоб выстроить модель, соберите и конфигурируйте блоки.

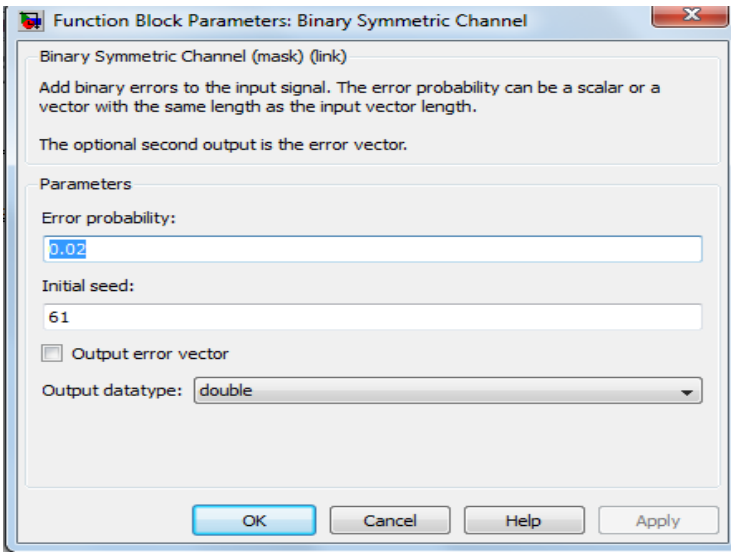
1. Блок Bernoulli Binary Generator из библиотеки Comm Sources.



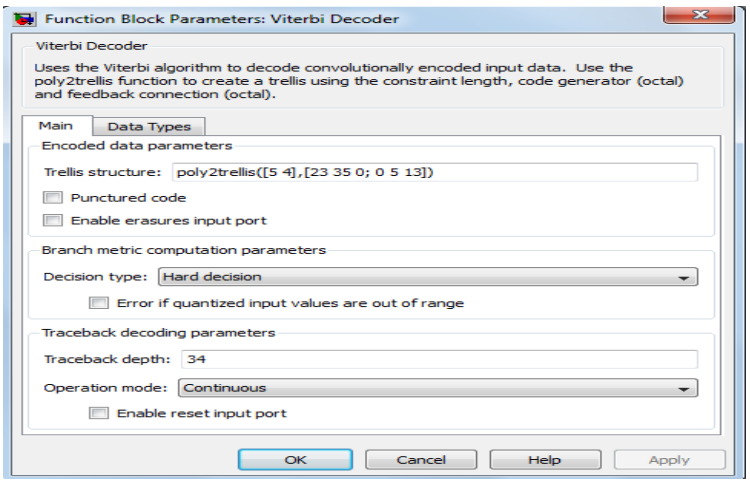
2. Блок Convolutional Encoder



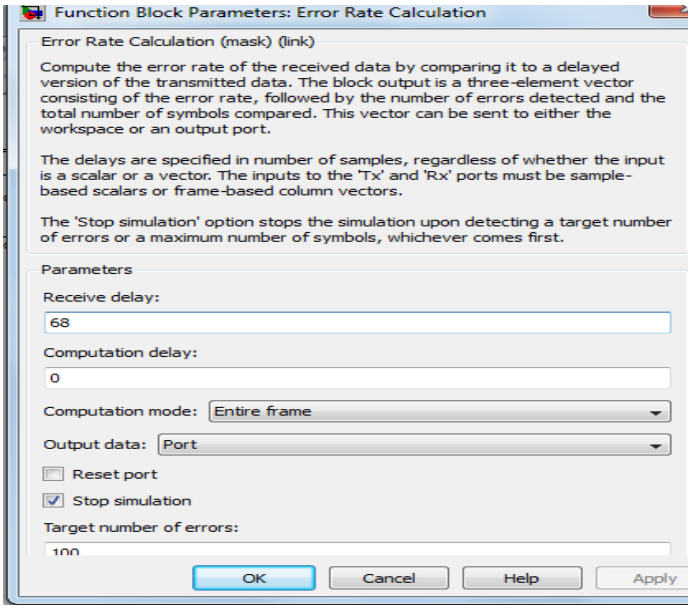
3. Блок Binary Symmetric Channel из библиотеки Channels



4. Блок Viterbi Decoder



5. Блок Error Rate Calculation из библиотеки Comm Sinks.



6. Блок Display из библиотеки Simulink Sinks. Потяните базисный край знака, чтоб сделать показ, довольно огромным для 3-х записей. Соедините блоки как на рисунке. Из меню Simulation, выберите Configuration parameters. В диалоговом окне установите параметр Stop time, равным inf.

Инструкции по размерам матрицы возникают на соединительных линиях, лишь если выбрать параметр Signal Dimensions подменю Port/signal displays из меню Format модели.

Кодирующее устройство воспринимает вектор кадра размера 2 x 1 и производит вектор кадра размера 3 x 1 , в то время как декодер делает обратное. Параметр Samples per frame parameter в блоке Bernoulli Binary Generator принят равным 2, так как блок должен произвести слово сообщения длины 2. Параметр Receive delay в блоке Error Rate Calculation принят равным 68, который является длиной вектора (2) восстановленного сообщения значения Traceback depth (34) в блоке Viterbi Decoder. Если исследуется переданные и

приобретенные сигналы, как матрицы в рабочем пространстве MATLAB, Вы видите, что 1-ые 34 строчки восстановленного сообщения состоят из нулей, в то время как следующие ряды - расшифрованные сообщения. Таковым образом, задержка приобретенного сигнала - 34 вектора длины 2, либо 68 отсчетов. Пуск модели производит выход монитора, состоящий из 3-х чисел: скорость роста ошибок, общее количество ошибок и общее количество ошибок сравнений, которые делает блок Error Rate Calculation во время моделирования. (1-ые два числа меняются в зависимости от Ваших значений Initial seed в блоках Bernoulli Binary Generator и Binary Symmetric Channel). Остановка моделирования после 100 ошибок происходит, так как параметр Target number of errors установлен равным 100, в блоке Error Rate Calculation. Скорость роста ошибки обязана быть меньше 0,02, как Error probability в блоке Binary Symmetric Channel.

Мягкое декодирование. Моделирование делает случайный сигнал двоичного сообщения, кодирует сообщение в сверточный код, модулирует код, используя двоичную манипуляцию со смещением фазы, (BPSK) и добавляет белый Гауссовский шум к модулированному сигналу, чтоб моделировать шум канала. Тогда, моделирование готовит полученную информацию для блока декодирования. В конце концов, моделирование ассоциирует декодированную информацию с сигналом начального сообщения, чтоб вычислить скорость роста ошибок. Конец моделирования наступает после обработки 100 битов ошибок либо 107 битов сообщения, после прибытия первого.

Длина ограничения кодирующего устройства - скаляр, потому что у кодирующего устройства есть один вход. Значение длины ограничения - число битов, сохраненных в сдвиговом регистре, включая текущий вход. Есть 6 регистров памяти, и текущий вход составляет один бит. Таковым образом, длина ограничения кода - 7. Кодовый генератор является матрицей размера 1 x 2 матрица восьмеричных чисел, так как кодирующее устройство имеет вход и два выхода, 1-ый элемент в матрице показывает на значение входа, влияющего на 1-ый выход, и 2-ой элемент в матрице показывает, на значение входа, влияющего на 2-ой выход.

К примеру, 1-ый выход в диаграмме кодирующего устройства – сумма по модулю 2 самого правого и 4 последних левых частей в массиве входных значений диаграммы. Двоичное число с семью цифрами 1111001 отражает эту информацию и эквивалентно восьмеричному числу 171. Восьмеричное число 171, таковым образом, становится первым входом порождающей кодовой матрицы. Тут, любая тройка употребляет последний левый бит как самый значимый бит. 2-ой выход соответствует двоичному числу 1011011, который эквивалентен восьмеричному номеру 133. Порождающий генератор кода потому [171 133].

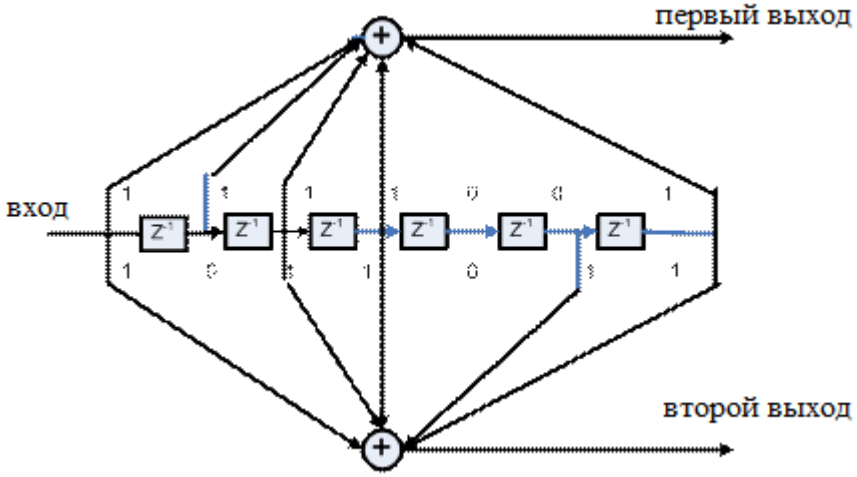


Рис.3.95. Модель кодирующего устройства

Параметр Trellis structure в блоке Convolutional Encoder говорит о том, какой блок, кодирует, когда поступают данные. В этом вариант, функция poly2trellis, в Communications Toolbox, преобразовывает длину кодового ограничения и пару восьмеричных чисел в решетчатую диаграмму. В то время как данные о сообщении, входящие в блок Convolutional Encoder, являются скалярным потоком битов, закодированные данные являются потоком бинарных векторов длиной 2.

Карта приобретенных данных. Приобретенные данные, другими словами, выход блока Convolutional Encoder, состоят из комплексных чисел, которые находятся в спектре меж -1 и 1. Чтоб вернуть оригинальное двоичное сообщение, приемная часть модели обязана декодировать сверточный код. Блок Viterbi Decoder в данной модели ждет, что ее входные данные будут целые числа меж 0 и 7. Демодулятор, рядовая подсистема в данной модели, преобразовывает приобретенные данные в формат, который блок Viterbi Decoder может интерпретировать подабающим образом. Наиболее точно, подсистема демодулятора: Конвертирует приобретенные данные в настоящий сигнал, удаляя его мнимую часть. Уместно представить, что мнимая часть приобретенных данных не содержит существенную информацию, так как мнимая часть переданных данных – есть ноль (игнорирование малых ошибок округления) и так как шум канала не чрезвычайно мощнейший. Восстанавливает приобретенные данные, деля на его текущее обычное отклонение и потом умножая на -1. Квантует нормализованные данные, используя три бита. Композиция данной карты и решения карты блока Viterbi Decoder на сто процентов изменяет модуляцию BPSK, чтоб блок BPSK Modulator Baseband выступил на передающей стороне данной модели. Декодирование сверточного кода. После получения данных подабающим образом нанесенных на карту векторов длиной 23-битовых значений решения, блок Viterbi Decoder

декодирует его. Блок употребляет метод мягенького декодирования с 23 разными входами поэтому, что параметр Decision type - есть Soft Decision и параметр Number of soft decision bits равен 3. Интерпретация данных мягенького решения. Когда характеристики Decision type установлен в состояние Soft Decision, блок Viterbi Decoder просит входных значений меж 0 и 2b-1, где b – есть параметр Number of soft decision bits. Блок интерпретирует 0 как самое достоверное решение того, что бит кодового слова, является 0 и интерпретирует 2b-1 как самое достоверное решение того, что бит кодового слова, является 1. Значения меж этими крайностями представляют наименее достоверные решения.

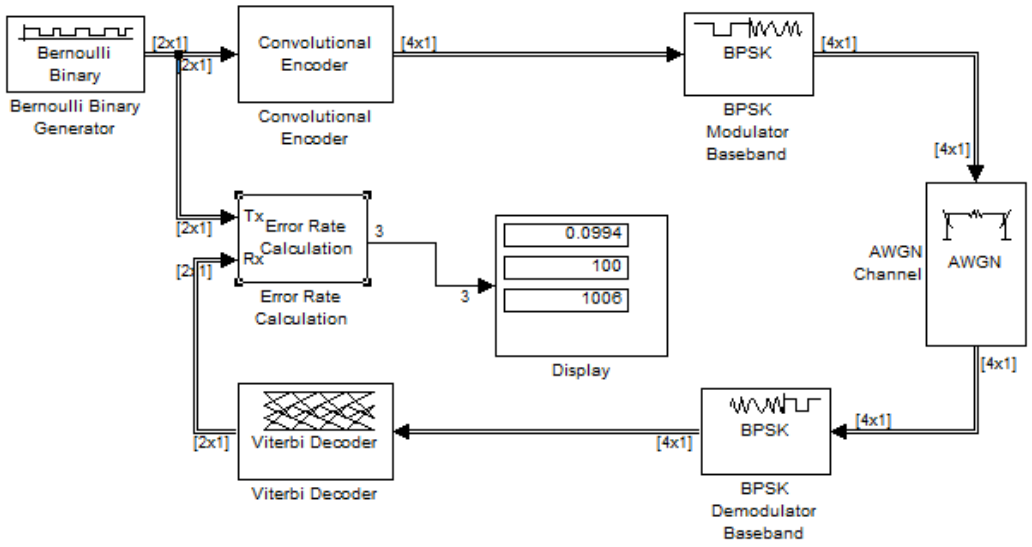


Рис. 3.96. Схема декодера Витерби в MATLAB

	Вер-ть ошибки 0,02	Вер-ть ошибки 0,03
Код (5, [35 31])	скорость роста ошибок: 0.5435 общее количество ошибок: 100 общее количество ошибок сравнений: 184	скорость роста ошибок: 0.5435 общее количество ошибок: 100 общее количество ошибок сравнений: 184
Код ([5 4],[23 35 0; 0 5 13])	скорость роста ошибок: 0,001779 общее количество ошибок: 100 общее количество ошибок сравнений: 5.62e+0.04	скорость роста ошибок: 0,01489 общее количество ошибок: 100 общее количество ошибок сравнений: 6714

Блоки программы декодера Витерби, реализованного в Matlab [21]

В блоке формирования узлов сетки выполняются следующие функции:

1. Формируется матрица фрагмента решетчатой диаграммы размера n∗m, в которой количество столбцов соответствует числу состояний кодера (узлов), а количество строк равно числу путей, подходящих к каждому узлу. Каждому элементу матрицы соответствуют ответвляемые слова при переходах между состояниями. Переходы между состояниями повторяются на протяжении всей решетки.

2. Формируется матрица переходов между состояниями.

3. Каждому переходу к конкретному узлу соответствует определенный входной бит (единица, либо нуль). Формируем матрицу размера (n∗m) состоящую только из нулей и единиц .

4. Для каждого входного дибита (слова) декодера вычисляются все возможные значения метрик по фрагменту решетчатой диаграммы кодера, сформированной в предыдущем блоке. В каждый момент времени происходит операция сложения по модулю два входного дибита с соответствующим данному моменту времени ответвляемым словом. Для исключения нижней части решетчатой диаграммы в начальные моменты времени мы искусственно увеличиваем значение метрик переходов нижней части решетки.

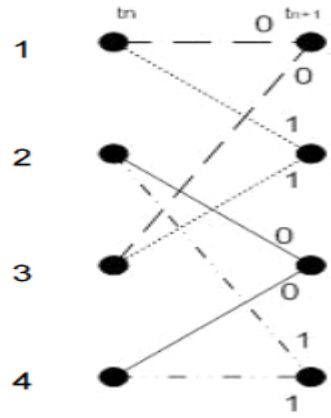


Рис. 3.97. Фрагмент решетчатой диаграммы декодера

Блок удаления путей:

1. В каждом узле, сравнивая метрики входных путей, обнуляем путь с наибольшей метрикой, тем самым убирая его из рассмотрения, но при этом могут возникнуть «тупиковые» пути.

2. Для удаления «тупиковых» путей, поочередно перебирая возможные m состояний в данный момент времени ищем отсутствие какого-либо элемента матрицы переходов

состояний, если элемент отсутствует, то пути в предыдущих фрагментах решетки (матрицы) удаляются.

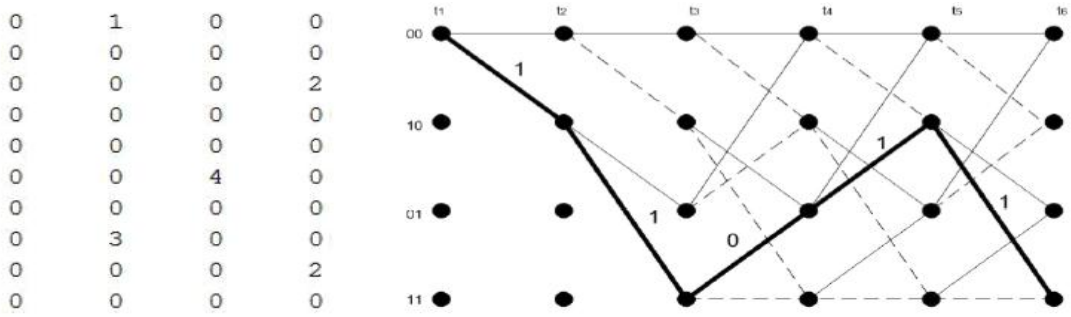


Рис. 3.98. Декодирование по выделенному пути

Блок декодирования:

Последовательно проверяем количество ненулевых переходов в *n* строках. Если такой переход единственный, то находим позицию этого элемента, и определяем какой бит был подан на вход кодера

Вопрос 1. Исследование методов построения кодеров непрерывных кодов.

В рабочем поле необходимо собрать схему для работы сверточного кода (5, [35 31]). Схема представлена на рисунке 3.99

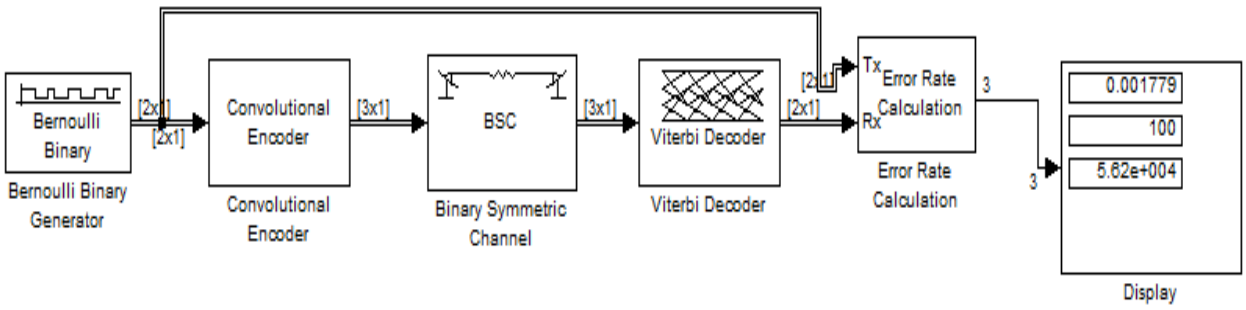


Рис. 3.99. Линия передачи с применением декодера Витерби.

В состав линии с кодированием входят:

1. BernoulliBinaryGenerator
2. Convolutional Encoder
3. Binary Symmetric Channel (каналпередачи)
4. Viterbi Decoder
5. Error RateCalculation (анализаторошибок)
6. Display

Устанавливаем характеристики блоков для кода (5, [35 31])

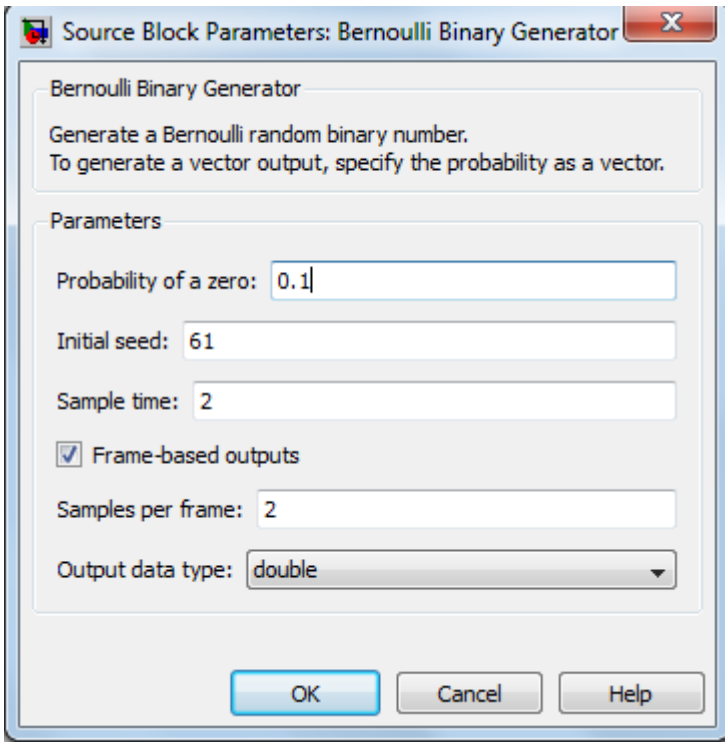


Рис. 3.102. Параметры Bernoulli Binary Generator

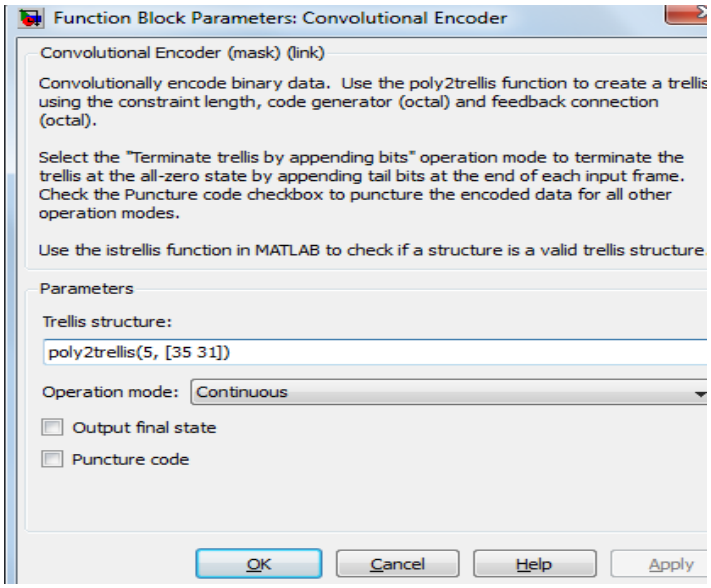


Рис. 3.103. Convolutional Encoder

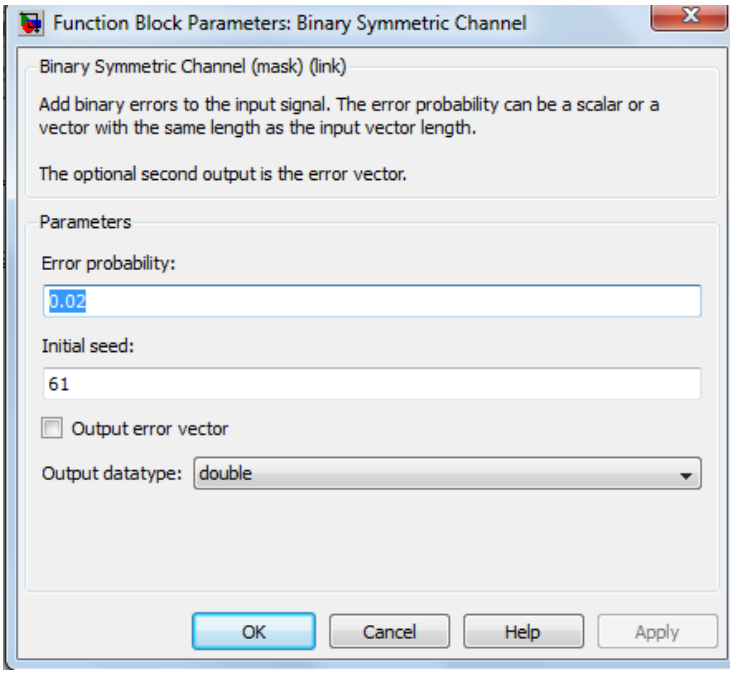


Рис. 3.104. Binary Symmetric Channel

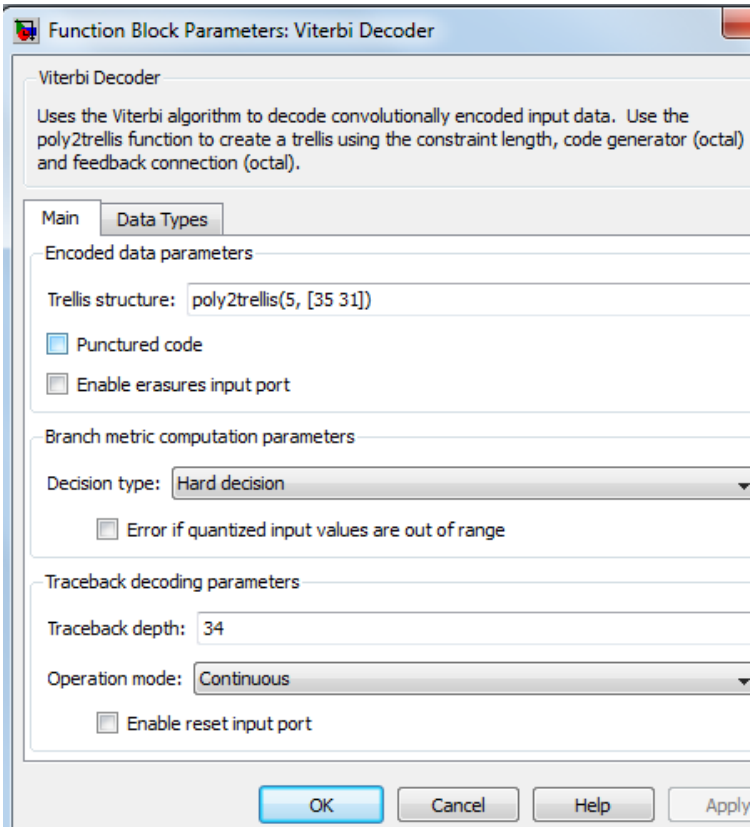


Рис. 3.100 Viterbi Decoder

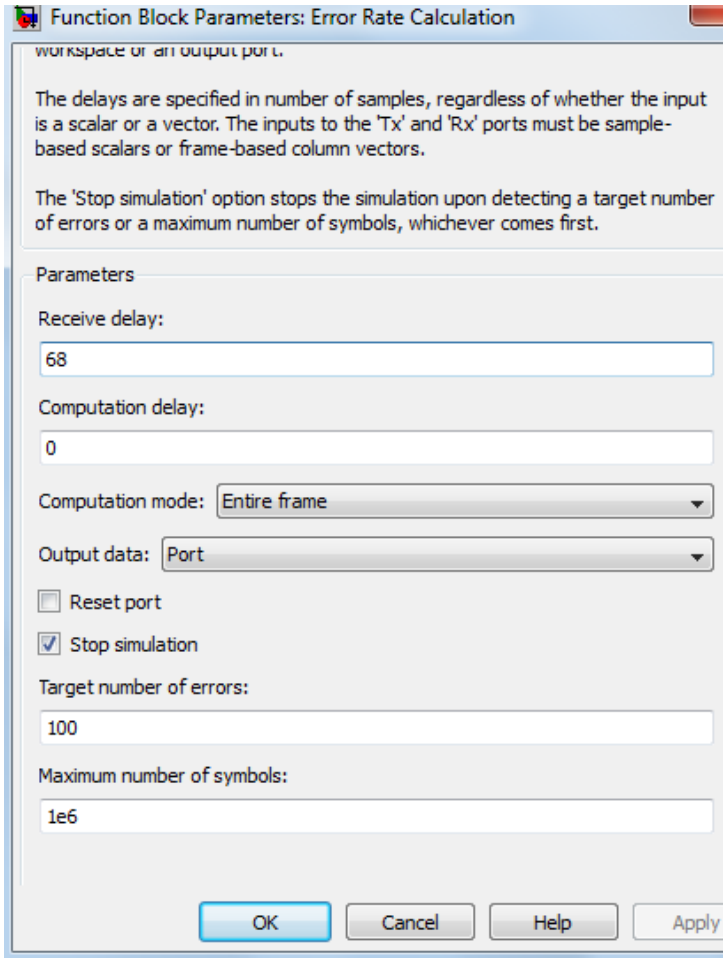


Рис. 3.101. Параметры Error Rate Calculation

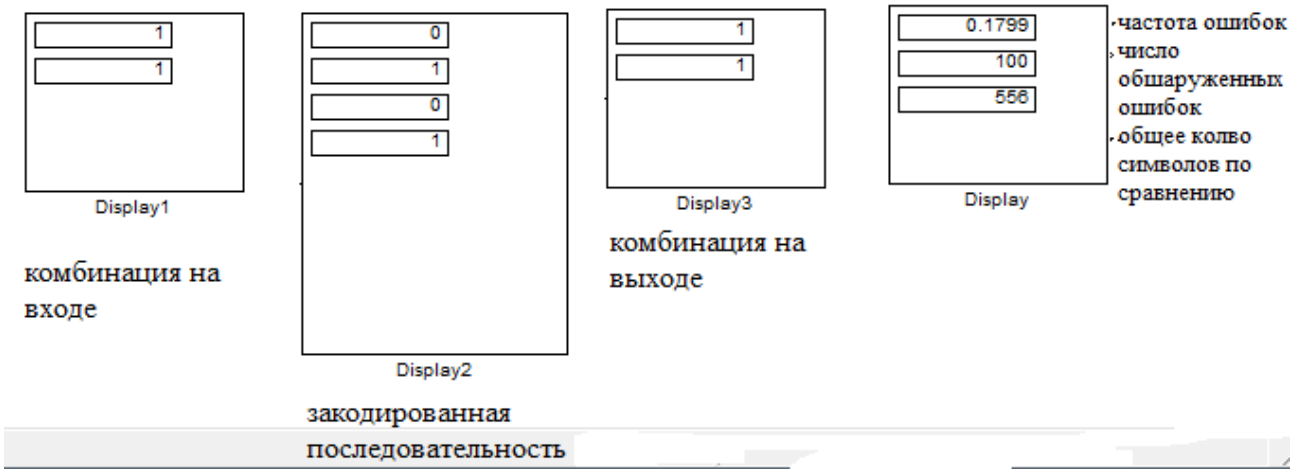


Рис. 3.102. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,02)

Анализируя рисунок выше, можно сделать вывод, что комбинация на входе совпадает с комбинацией на выходе, таким образом, передача осуществилась удачно. Что касается

ошибок, то их частота равна 0,1799, число ошибок равно 100, общее количество символов по сравнению равно 556.

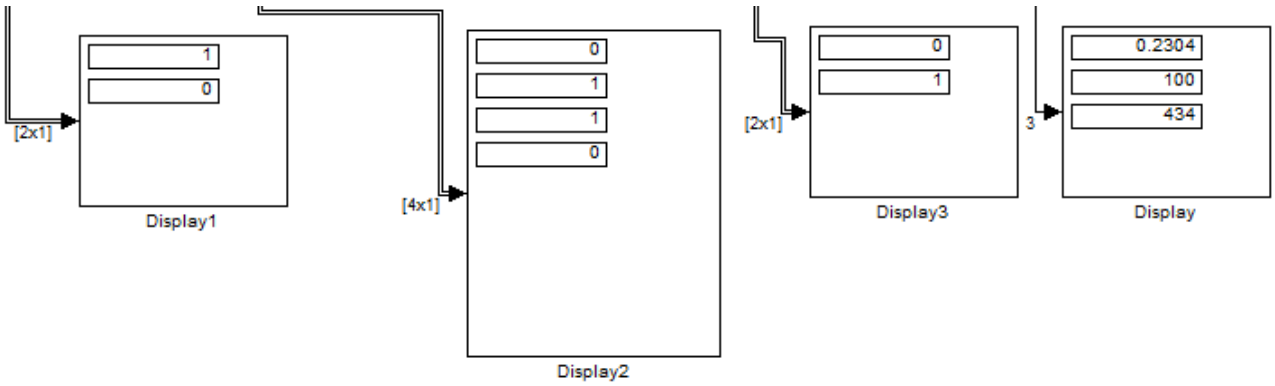


Рис. 3.103. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,08)

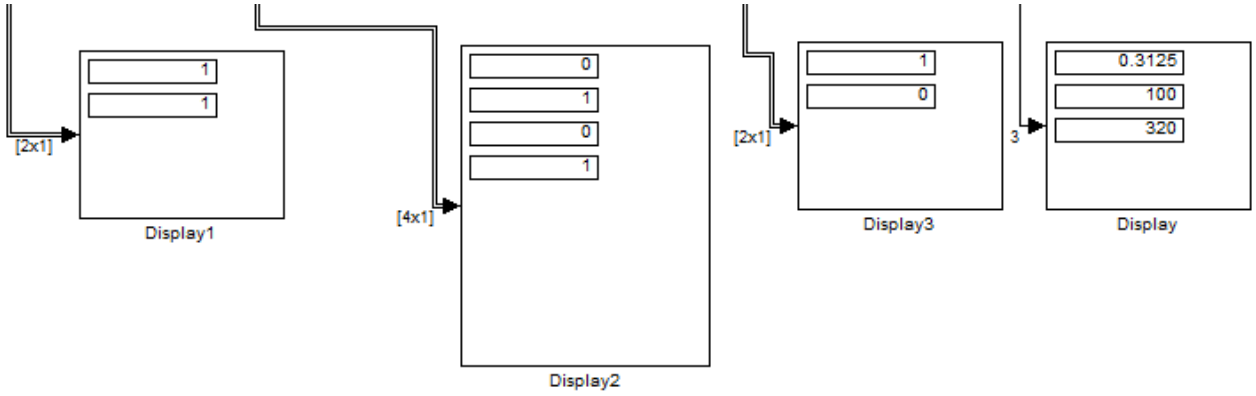


Рис. 3.104. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,12)

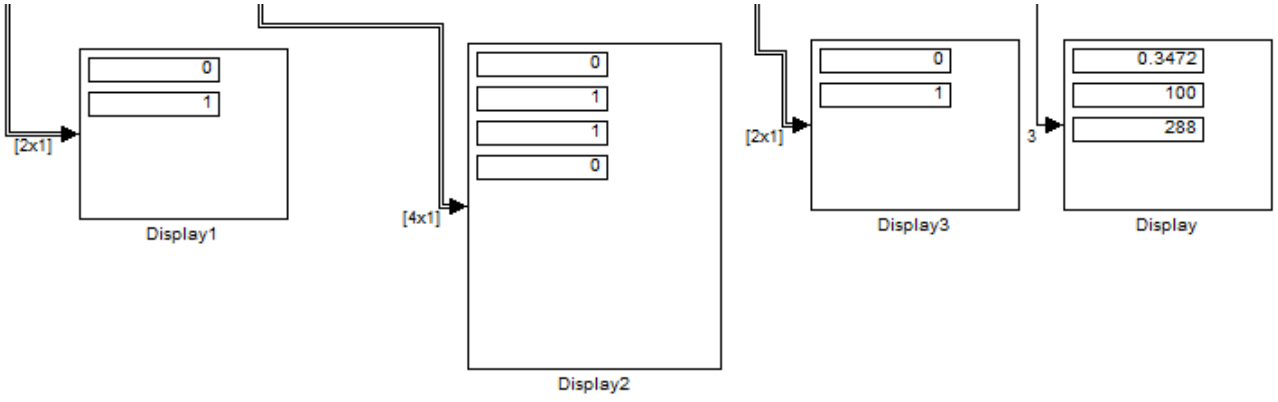


Рис. 3.105. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,14)

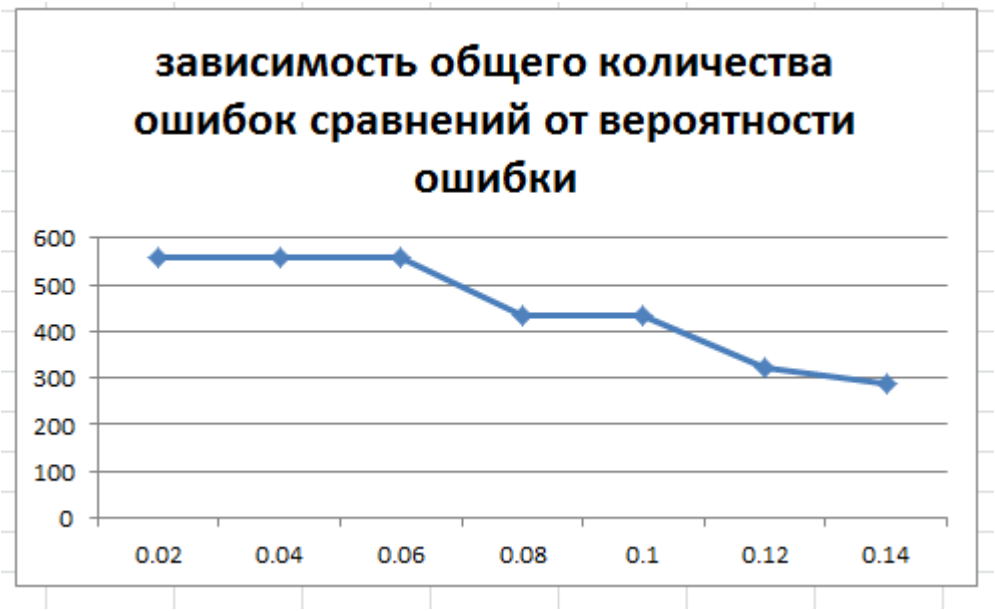


Рис. 3.106. График зависимости общего количества ошибок сравнений от вероятности ошибки для кода (5, [35 31])

Устанавливаем характеристики блоков для кода ([5 4],[27 33 0; 0 5 13])

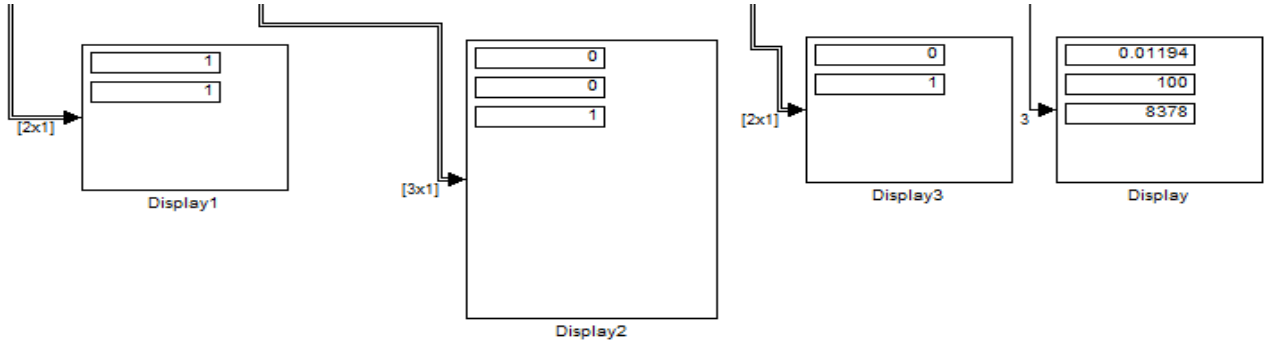


Рис. 3.107. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,02)

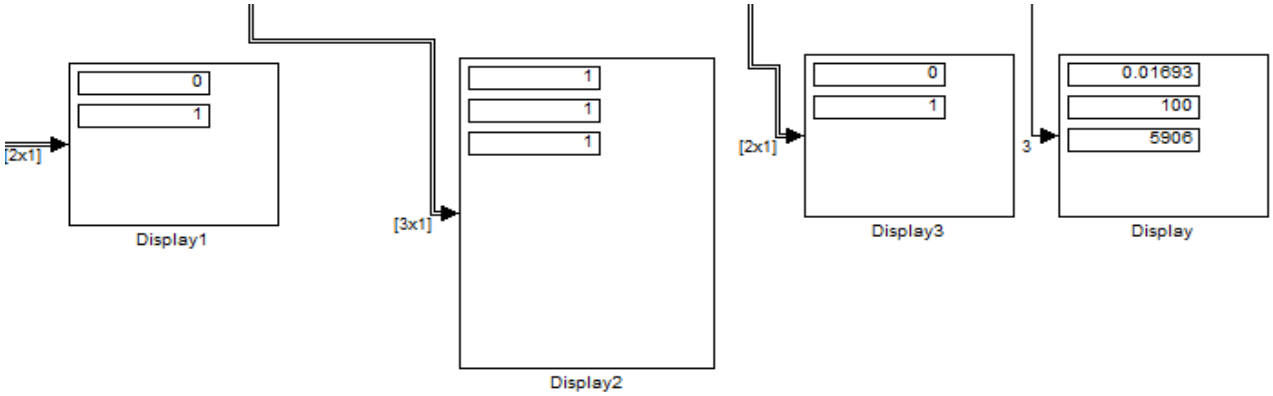


Рис. 3.108. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,025)

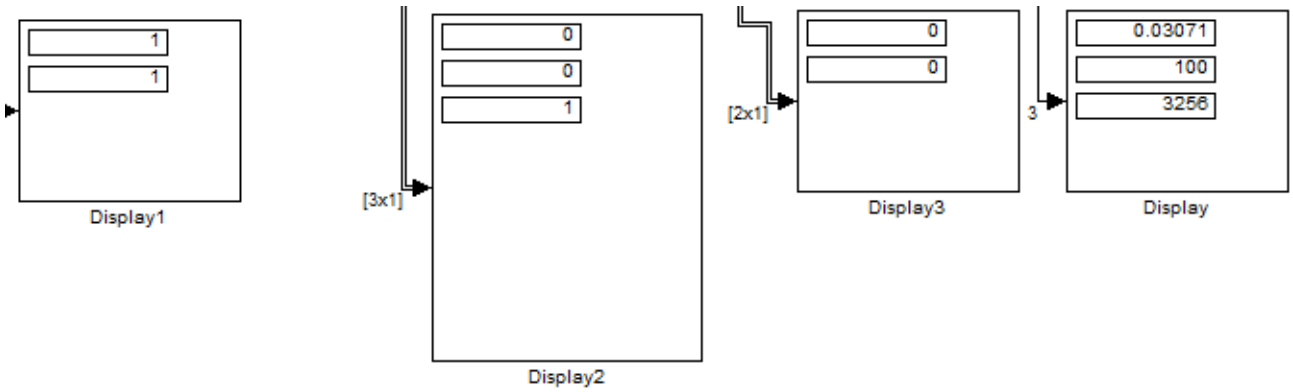


Рис. 3.109. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,03)

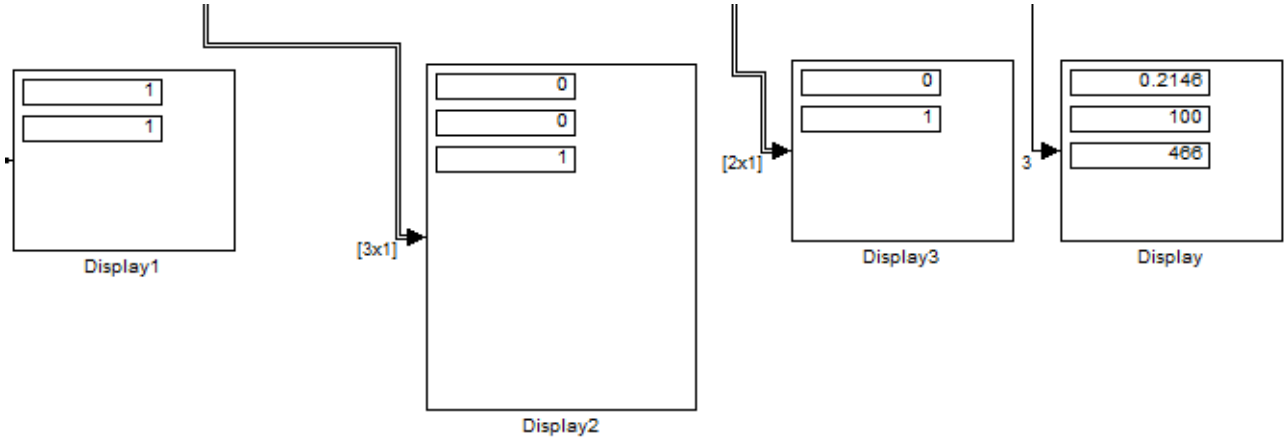


Рис. 3.110. Комбинация на входе, Закодированная последовательность, Комбинация на выходе, Ошибки (вероятность ошибок равна 0,035)

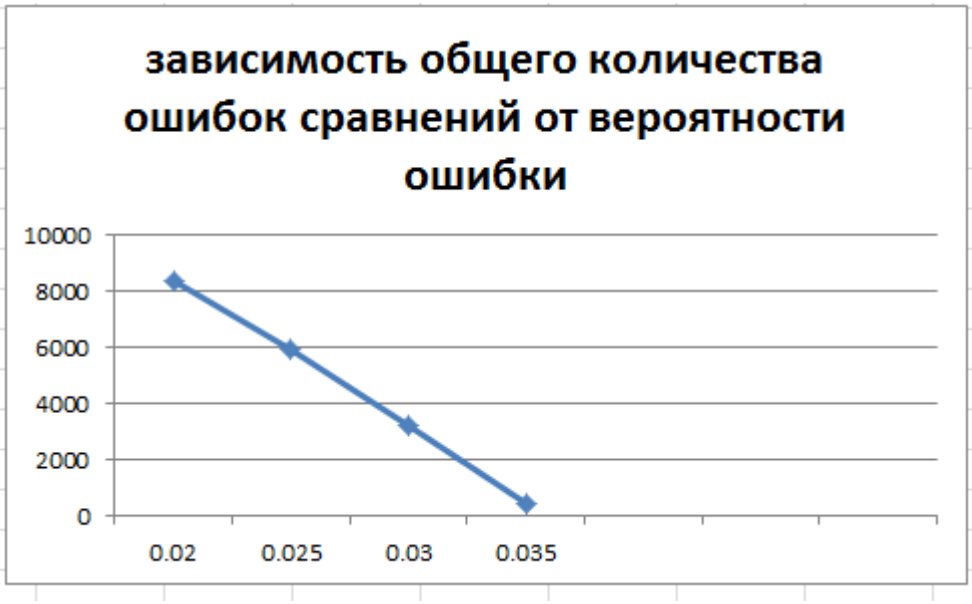


Рис. 3.111. График зависимости числа ошибок от вероятности ошибки для ([5 4], [27 33 0; 0 5 13])

Вопрос 2. Построение схемы декодирования непрерывных кодов в канале с Гауссовскими помехами.

В рабочем поле необходимо собрать схему для работы сверточного кода (7, [171 133]). Схема представлена на рисунке 3.112

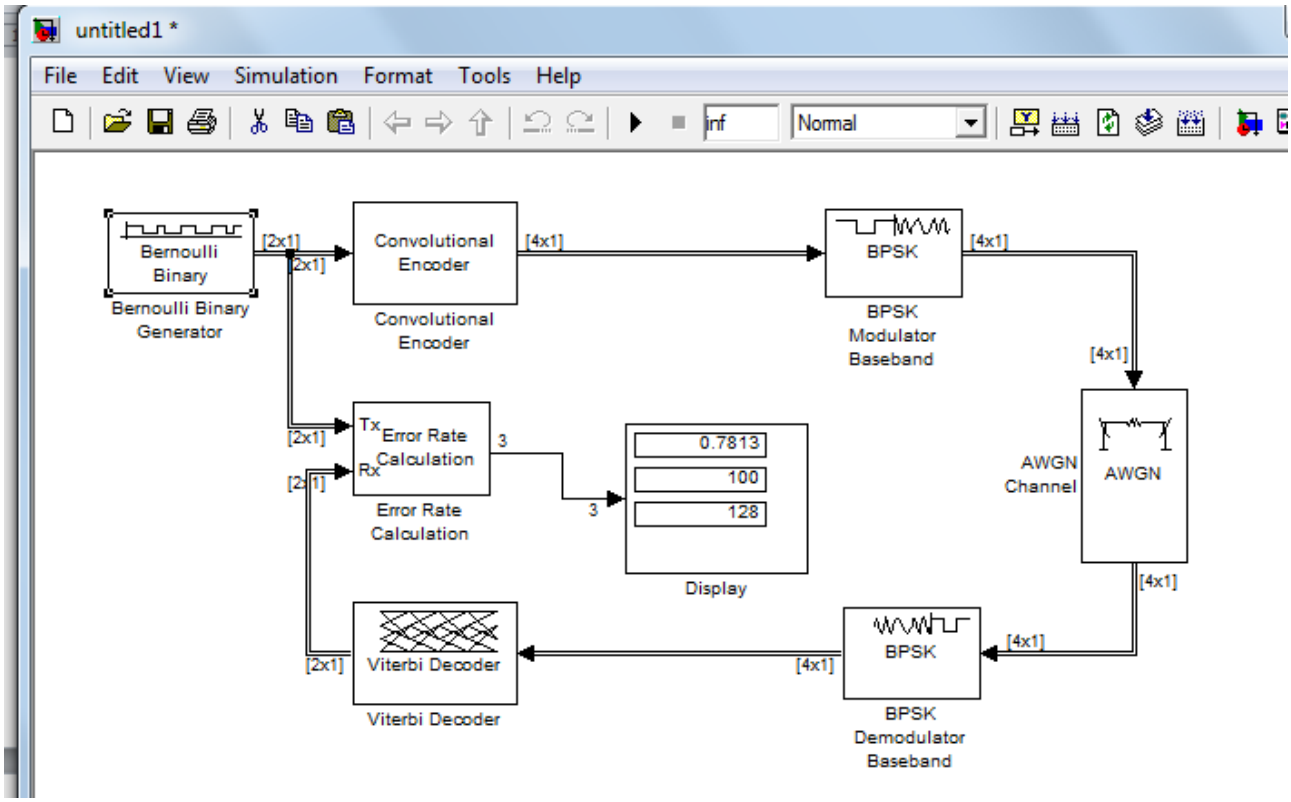


Рис. 3.112. Линия передачи с применением декодера Витерби

В результате практической работы построена схема линии передачи с декодированием Витерби в среде Simulink. Построены графики зависимостей общего количества ошибок сравнений на выходе декодера от вероятности ошибки в канале связи для кодов(5, [35 31]) и ([5 4],[27 33 0; 0 5 13]). Из графиков видно, что число ошибок сравнений уменьшается с ростом вероятности ошибок.

3.5. Турбокодирование. Обобщенная схема турбокодера ТСС с параллельным каскадированием. Сверточные турбокоды. Декодирование турбокодов. Характеристики помехоустойчивости сверточных турбокодов ТСС. Исследование турбокодов с использованием ПО MATLAB.

Принципы построения турбо-кодов

В данном разделе приведены основные теоретические данные одного из важнейших классов итеративно декодируемых кодов – турбо-кодов (turbo codes) и с их обобщением –

многократными турбо-кодов (multiple turbo codes). Турбо-кодирование основано на двух фундаментальных идеях: построение кодов с кодовыми словами, обладающими квазислучайными свойствами и построение декодеров, основанных на легко реализуемых алгоритмах итеративного декодирования [1]. Турбо-коды обладают хорошими характеристиками, особенно при больших длинах кодов и умеренных требованиях к вероятности ошибки на бит.

Параллельное каскадирование двух сверточных кодов

Турбо-код можно определить как параллельное каскадирование (parallel concatenation) двух сверточных кодов. Кодер состоит из двух параллельных систематических кодеров скорости R =1/2 и памяти m = 2 (компонентные кодеры (constituent encoders)) с обратной связью (recursive encoders) имеющих рациональные порождающие подматрицы:

$$G = (1 + \frac{1 + D^2}{1 + D + D^2}) \qquad (3.1)$$

и простого блокового перемежителя (ПБП). Входом турбо-кодера служит двоичная информационная последовательность длины K

$$u = u_0 u_1 \ldots u_{K-1} \qquad (3.2)$$

Входом первого составного кодера и перемежителя, ассоциированного с вторым составным кодером, служит последовательность u длины $K = K + m$,

$$u = u_0 u_1 \ldots u_{K-1} \qquad (3.3)$$

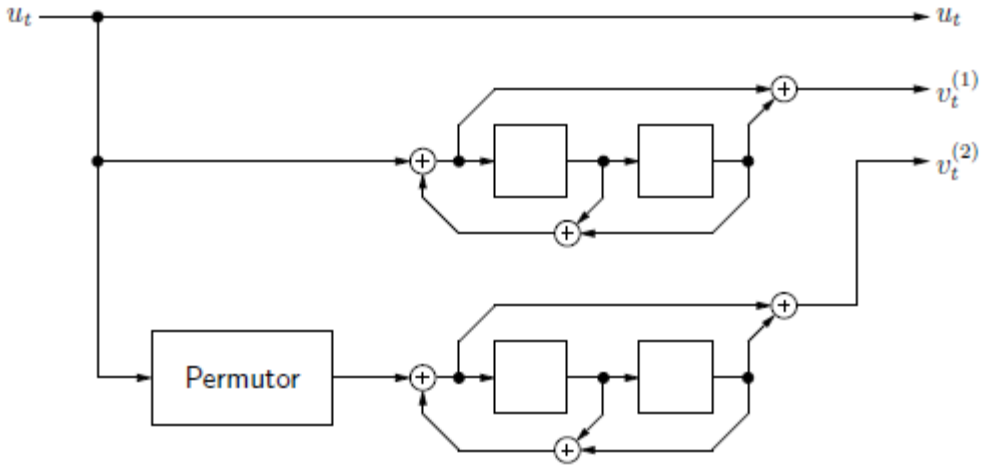


Рис. 3.113. Систематический сверточный кодер скорости R = 1/3, основанный на параллельном каскадировании двух сверточных кодеров

где m память составного кодера. Первые K символов этой последовательности – это символы информационной последовательности u, последние m символов образуют хвост,

приводящий первый компонентный кодер в нулевое состояние. Поскольку кодирование систематическое, входная последовательность (включая хвост) имеет вид:

$$v(0) = v_0{}^{(0)}\, v_1{}^{(0)} \ldots v_{K-1}{}^{(0)} = u. \qquad (3.4)$$

Первый компонентный кодер генерирует проверочную последовательность:

$$v(1) = v_0{}^{(1)}\, v_1{}^{(1)} \ldots v_{K-1}{}^{(1)}. \qquad (3.5)$$

Поскольку хвост заставляет первый компонентный кодер вернуться в нулевое состояние, мы можем использовать БКВР алгоритм апостериорно вероятностного декодирования кодовой последовательности первого компонентного кодера.

Параллельно последовательность u поступает в ПБП, определенный перестановочной матрицей $P = (pik)$, $i = 0, 1, \ldots, K-1$, $k = 0, 1, \ldots, K-1$ размеров (K x K). Выходная последовательность $u(2) = uP$ ПБП поступает во второй компонентный кодер, генерирующий проверочную последовательность

$$v(2) = v_0{}^{(2)}\, v_1{}^{(2)} \ldots v_{K-1}{}^{(2)} \qquad (3.6)$$

Вообще говоря, второй кодер не переходит в конце кодирования в нулевое состояние. В этом случае нельзя использовать для декодирования БКДР алгоритм, но можно использовать однопутевой алгоритм апостериорно вероятностного декодирования. При больших длинах K это не сильно отражается на характеристиках декодирования и вызывает лишь их небольшую деградацию.

ПБП, использованный во втором кодере, перестановочной матрицей P. С помощью введения вектора перестановки индексов (indexpermutation vector) длины K можно дать альтернативное определение ПБП:

$$\pi = (\pi_0\, \pi_1 \ldots \pi_{K-1}) \qquad (3.7)$$

где $0 \le \pi_i{}' \le K-1$, $i = 0, 1, \ldots, K-1$, $\pi' \not= \pi''$ если $i \not= i'$ и π' таково, что $p_{\pi' ii} = 1$, то есть $p_{\pi' ii}$ – это единственный ненулевой элемент в -ом столбце перестановочной матрицы P.

Комбинация трех последовательностей, $v^{(0)}$, $v^{(1)}$, и $v^{(2)}$ дает полную передаваемую последовательность (кодовое слово)

$$v = v_0 v_1 \ldots v_{K-1} \qquad (3.8)$$

где $v_n = v_n{}^{(0)}\, v_n{}^{(1)}\, v_n{}^{(2)}$, $n = 0, 1, \ldots, K-1$, такая что общая длина блока кода $N = 3K$. Поскольку, как правило $K \gg m$, скорость кода $R \approx 1/3$.

Как было отмечено выше, второй кодер, вообще говоря, не возвращается в нулевое состояние. Важным исключением является турбо-кодер, у которого оба компонентных кодера – это рекурсивные сверточные кодеры памяти $m = 1$ с порождающей матрицей.

$$G = \left(1 + \frac{1}{1+D}\right) \qquad (3.9)$$

В этом случае хвост состоит из одного символа u_{K-1}, равному 0, если вес информационной последовательности четный, и равен 1, если вес нечетный. Такая входная последовательность приводит оба компонентных сверточных кодера памяти $m = 1$ в нулевое состояние. Поэтому решетки компонентных сверточных кодеров начинаются и заканчиваются в нулевом состоянии и мы можем использовать БКДР алгоритм для декодирования обоих составных кодов.

Важной особенностью обычных турбо-кодов является слабый рост минимального расстояния $d$min с ростом длины кода.

Параллельное каскадирование трех и более сверточных кодов

Обычные турбо-коды имеют относительно плохое минимальное расстояние. При относительно высоких отношениях сигнал-шум это вызывает эффект “зависания” вероятности ошибки декодирования, при котором вероятность ошибки очень медленно убывает с ростом отношения сигнал-шум. Этот участок графика называется “полом” (floor).

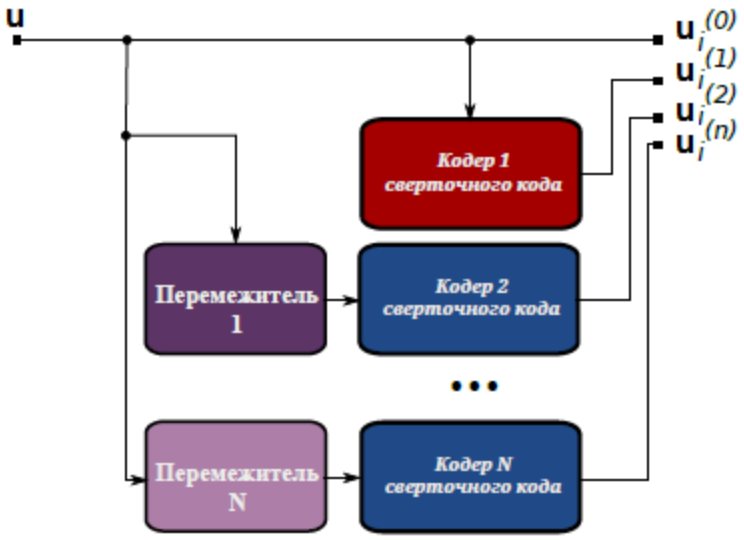


Рис. 3.114. Структура -размерного сверточного турбокодера

Хотя минимальное расстояние турбо-кода может быть улучшено выбором хорошего перемежителя, оно не может расти быстрее, чем логарифмически с ростом длины блока. Поведение минимальное расстояния МТК существенно лучше.

Кодер -мерного МТК состоит из J, $J \geq 3$, параллельных компонентных сверточных кодеров и J единичных блоковых перемежителей $P(j)$, j =1, 2, . . . , J, размеров K Ч K, как показано на рис. 7.2. (Определим формально нулевой перемежитель $P(0)$ как единичную матрицу размеров K x K.) Предполагается, что все компонентные кодеры – это

систематические рекурсивные кодеры скорости R = 1/2 и памяти m. Двоичная последовательность длины K

$$u = u_0 u_1 \ldots u_{K-1} \tag{3.10}$$

это входная последовательность МТК. Это может быть информационная последовательность; в этом случае все компонентные кодеры – это коды с нейтрализацией хвоста. Аналогично обычным турбо-кодам следует рассмотреть случай, когда u состоит из K информационных символов плюс хвост из m двоичных символов, так что $K = K+ m$. В частности, если все компонентные кодеры – это систематические рекурсивные сверточные кодеры памяти 1, хвост состоит из 1 символа.

Последовательность u поступает на входы J ПБП, описываемым перестановочными матрицами $P(j) = (p(j)_{ik})$, где $j = 1, 2, \ldots, J$, $i, k = 0, 1, \ldots, K-1$, размеров KxK. За перемежителями следуют компонентные кодеры, такие что -ая переставленная версия u, то есть $u(j) = uP(j)$, $j = 1, 2, \ldots, J$, поступает на вход j-ого компонентного кодера.

Выходная последовательность -ого компонентного кодера (проверочная последовательность) – это последовательность:

$$v^{(j)} = v_0{}^{(j)}\, v_1{}^{(j)} \ldots v_{K-1}{}^{(j)},\ j = 1, 2, \ldots, J. \tag{3.11}$$

Входная последовательность МТК,

$$u = v^{(0)} = v_0{}^{(0)}\, v_1{}^{(0)} \ldots v_{K-1}{}^{(0)},$$

перемешивается с выходными последовательностями J компонентных кодеров и передается по каналу. Кодовое слово многократного турбо-кода,

$$v = v_0\ v_1\ \ldots v_{K-1} \tag{3.12}$$

где $v_n = v_n{}^{(0)}\, v_n{}^{(1)} \ldots v_n{}^{(J)}$, $n = 0, 1, \ldots, K-1$, так что длина блока кода равна $N = (J+1)K$ и результирующая скорость равна $R = 1/(J+1)$. Кодовая скорость может быть увеличена выкалыванием символов.

Комбинация тождественного перемежителя и остальных J перемежителей называется (J +1)-мерным перемежителем. Он может быть представлен в виде (J + 1)-мерной таблицы, сконструированной так, что проекция ненулевого элемента на плоскость, образованную 0-ой и -ой, $j = 1, 2, \ldots, J$, осями давало быматрицу перестановок $P(j)$, $j = 1, 2, \ldots, J$.

Итеративное декодирование турбо-кодов

В данном разделе описано декодирование турбо-кодов. Предположим, что кодовая последовательность $v = v_0 v_1 \ldots v_{K-1}$, сконструированная из трех последовательностей, $v^{(0)} = u$, $v^{(1)}$, and $v^{(2)}$, передается по дискретному каналу без памяти. Пусть

$$r = r_0\, r_1 \ldots r_{K-1} \tag{3.13}$$

где $r_n = r_n{}^{(0)}\, r^{(1)}{}_n\, r_n{}^{(2)}$, $n = 0, 1, \ldots, K-1$, означает принятую последовательность и пусть

$$\pi_n^{(0)}(0) = P(u_n = 0) \tag{3.14}$$

означает априорную вероятность, что информационный символ $u_n = 0$, n =0, 1, . . . ,$K - 1$. На практике обычно $P(u_n = 0) = 1/2$.

Итеративный алгоритм декодирования турбо-кодов похож на итеративный алгоритм распространения доверия кодирования МППЧ кодов. Декодер состоит из двух компонентных апостериорно-вероятностных декодеров. Процесс декодирования состоит из I итераций; каждая итерация, в свою очередь, состоит из двух фаз. В результате -ой фазы, e = 1, 2, i-ой итерации i = 1, 2, . . . , I, вычисляется последовательность:

$$\pi^{(e)}(i) = \pi_0^{(e)}(i)\,\pi_1^{(e)}(i) \ldots \pi_{K-1}^{(e)}(i) \tag{3.15}$$

где $\pi(e)\ n\ (i)$ апостериорная вероятность 2, что информационный символ $un = 0$, n = 0, 1, . . . ,$K - 1$. Удобно вместо $\pi(e)n\ (i)$ оперировать с соответствующими логарифмическими отношениями правдоподобия, то есть,

$$z_n^{(e)} = \log \frac{\pi_n^{(e)}}{1-\pi_n^{(e)}} \tag{3.16}$$

и вместо $\pi(e)(i)$ оперировать с последовательностью:

$$z^{(e)}(i) = z_0^{(e)}(i)\, z_1^{(e)}(i) \ldots z_{K-1}^{(e)}(i) \tag{3.17}$$

Можно использовать обозначение:

$$z_n^{(0)}(0) = \log \frac{P(\mathrm{u}_n = 0 \,|\, r_n^{(0)})}{P(\mathrm{u}_n = 1 \,|\, r_n^{(0)})} = \log \frac{\pi_n^{(0)}}{1-\pi_n^{(0)}} + \log \frac{P(\mathrm{u}_n = 0 \,|\, r_n^{(0)})}{P(\mathrm{u}_n = 1 \,|\, r_n^{(0)})} \tag{3.18}$$

для логарифмического отношения апостериорных вероятностей значений символа un, при условии, что принят символ rn. Оно называется внутренней информацией (intrinsic information) об информационном символе $un = v(0)n$. Предполагается, что оба компонентных декодера хранят в памяти последовательность $z^{(0)}(0) = z_0^{(0)}(0)\, z_1^{(0)}(0) \ldots z_{K-1}(0)\,(0)$ и используют ее на каждом шаге итеративного процесса декодирования.

Пусть $r^{(0)}$, $r^{(1)}$, and $r^{(2)}$ означают последовательности принятых символов, соответствующих переданным информационной последовательности $v^{(0)} = u$, первой проверочной последовательности $v^{(1)}$, и второй проверочной последовательности $v^{(2)}$, соответственно. Пусть, далее, $r`_n^{(0)}$ означает последовательность принятых символов, соответствующих последовательности u за исключением принятого символа $r_n^{(0)}$, то есть,

$$r`_{\mathrm{n}}^{(0)} = r_0^{(0)}\, r_1^{(0)} \ldots r_{n-1}^{(0)}\, r_{n+1}^{(0)} \ldots \tag{3.19}$$

Аналогично, пусть $\pi(e)/n\ (i)$ означает последовательность апостериорных вероятностей для символов последовательности u за исключением $\pi(e)\ n\ (i)$ после -ой фазы i-ой итерации, то есть

$$\pi_{n}^{(e)}(i) = \pi_0^{(e)}(i)\pi_1^{(e)}(i) \ldots \pi_{n-1}^{(e)}(i)\pi_{n+1}^{(e)}(i) \ldots \quad (3.20)$$

На первой фазе первой итерации первый компонентный декодер, используя $r(0)n$, $r(1)$ и

$$\pi_{n}^{(0)}(0) = \pi_0^{(0)}(0)\pi_1^{(0)}(0) \ldots \pi_{n-1}^{(0)}(0)\pi_{n+1}^{(0)}(0) \ldots \quad (3.21)$$

вычисляет логарифмическое отношение правдоподобия

$$y_n^{(1)}(1) = \log \frac{P(\mathrm{r}_n^{(0)}, \mathrm{r}^{(1)} \left| r_n^{(0)} \mathrm{u}_n = 0\right.)}{P(\mathrm{r}_n^{(0)}, \mathrm{r}^{(1)} \left| \mathrm{u}_n = 1\right.)} \quad (3.22)$$

для информационных символов $u_n = v_n^{(0)}$, $n = 0, 1, \ldots, K-1$.

Для вычисления $y(1)n$ (1) декодер использует БКВР алгоритм апостериорновероятностного декодирования описанный а параграфе 4.5. Величина $y(1)n$ (1) называется внешней информацией (extrinsic information) первого компонентного декодера об информационном символе un при первой итерации. Первый компонентный декодер вычисляет последовательность:

$$z^{(1)}(1) = z_0^{(1)}(1)z_1^{(1)}(1) \ldots z_{K-1}^{(1)}(1) \quad (3.23)$$

используя формулу

$$z_n^{(1)}(1) = z_n^{(0)}(0) + y_n^{(1)}(1) \quad (3.24)$$

и посылает ее второму компонентному декодеру.

На второй фазе первой итерации второй компонентный декодер, зная последовательность $z_n^{(1)}(1)$, вычисляет апостериорные вероятности $\pi_n^{(1)}(1)$, $n = 0, 1, \ldots, K-1$, и, следовательно, $\pi_n^{(1)}(1)$. Используя $r_n^{(0)}$, $r^{(2)}$, и $\pi_n^{(1)}(1)$, декодер, вычисляет внешнюю информацию$y_n^{(2)}(1)$,

$$y_n^{(2)}(1) = \log \frac{P(\mathrm{r}_n^{(0)}, \mathrm{r}^{(2)} \left| r_n^{(0)} \mathrm{u}_n = 0\right.)}{P(\mathrm{r}_n^{(0)}, \mathrm{r}^{(2)} \left| \mathrm{u}_n = 1\right.)} \quad (3.25)$$

Затем декодер вычисляет новое логарифмическое отношение правдоподобия и посылает:

$$z^{(2)}(1) = z_0^{(2)}(1)\, z_1^{(2)}(1) \ldots z_{K-1}^{(2)}(1) \quad (3.26)$$

первому компонентному декодеру. Это завершает первую итерацию.

Рассмотрим теперь -ую итерацию, $i = 2, 3, \ldots, I$. На первой фазе i-ой итерации первый компонентный декодер, зная последовательность логарифмических отношений правдоподоподобий $z^{(2)}(i-1)$, вычисляет последовательность апостериорных вероятностей $\pi^{(2)}(i-1)$ и, следовательно, $\pi_n^{(2)}(i-1)$. Используя $r_n^{(0)}$, $r^{(1)}$, и $\pi_n^{(2)}(i-1)$, первый компонентный декодер вычисляет $y_n^{(1)}(i)$,$n = 0, 1, \ldots, K-1$, и $z_n^{(1)}(i)$, $n = 0, 1, \ldots, K-1$. Затем он посылает $z^{(1)}(i)$ второму компонентному декодеру и т.д.. Идея состоит в том, что на каждой фазе итеративного процесса декодирования активный компонентный декодер использует (апроксимированные) апостериорные вероятности, вычисленные другим

компонентным декодером на предыдущем шаге декодирования в качестве априорных вероятностей и вычисляет новую апроксимацию апостериорных вероятностей.

Декодер чередует эти фазы декодирования в течение I итераций. Решение u_n, n = 0, 1, . . . ,K − 1, получается сравнением финального логарифмического отношения $z_n^{(2)}(I)$ с порогом 0:

$$u_n = \begin{cases} 0, \text{если} & z_n^{(2)}(\mathrm{I}) \succ 0; \\ 1, \text{если} & z_n^{(2)}(\mathrm{I}) \prec 0. \end{cases} \qquad (3.27)$$

(Если $z_n^{(I)} = 0$, тогда декодер использует правило бросания монеты и принимает решение $u_n = 0$ или $u_n = 1$ с вероятностью 1/2.) Итеративный декодер иллюстрируется на рис. 2.3.1, где $u = u_0 u_1 \ . \ . \ . u_{K-1}$. Итеративное декодирование многократных турбо-кодов аналогично итеративному декодированию обычных турбо-кодов. Декодирование многократных турбокодов с J компонентными кодами состоит из J итераций, каждая итерация аналогична итерациям декодирования обычных турбо-кодов.

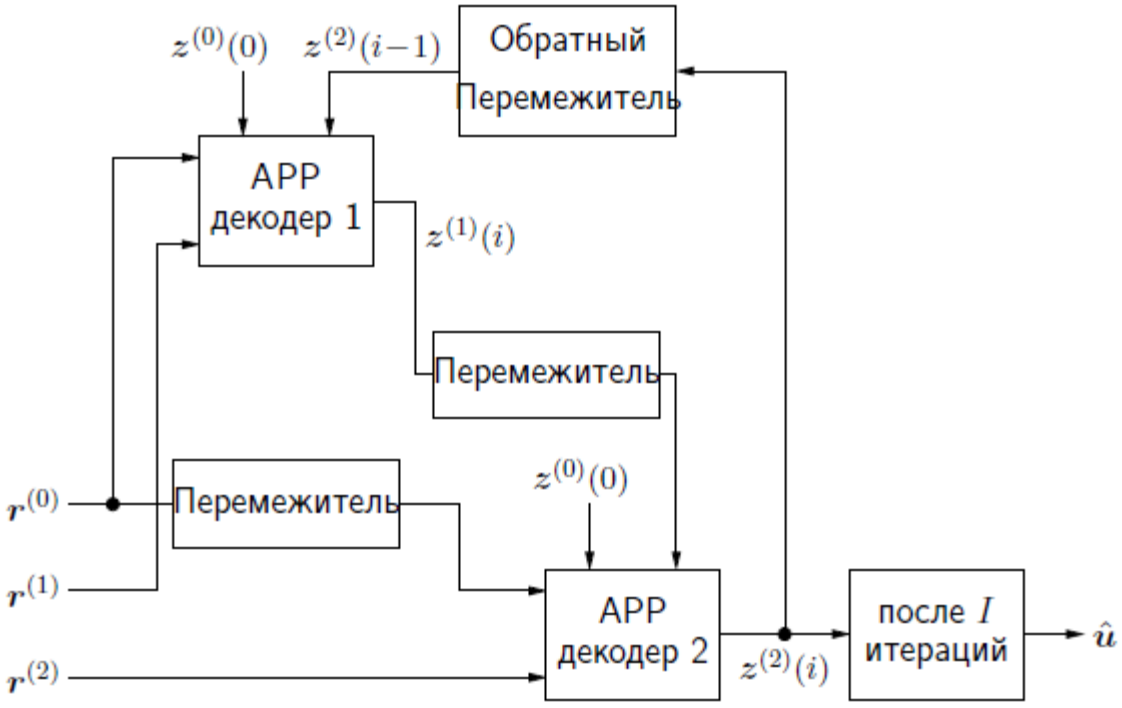


Рис. 3.115. Итеративный декодер для турбо-кодов

Разработка модели системы связи с использованием турбо-кодов

В диалоговом окне приложения Simulink необходимо собрать схему, представленную на рисунке 3.116.

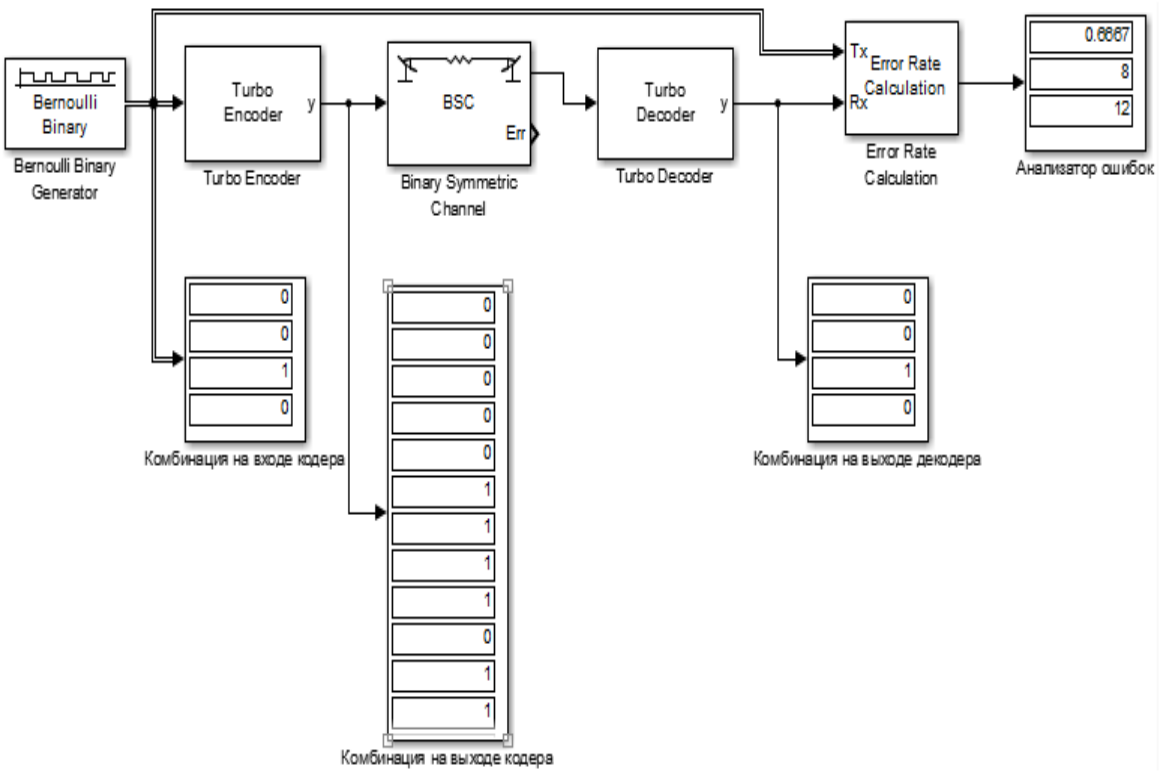


Рис. 3.116. Модель системы связи с использованием турбо-кодов

В состав проектируемой системы связи входят следующие элементы:

1. Bernoulli Binary Generator;
2. Turbo Encoder;
3. Binary Symmetric Channel (канал передачи);
4. Turbo Decoder;
5. Error Rate Calculation (анализатор ошибок);
6. Дисплей, отображающий комбинацию на входе кодера;
7. Дисплей, отображающий комбинацию на выходе кодера;
8. Дисплей, отображающий комбинацию на выходе декодера;
9. Дисплей анализатора ошибок, отображающий частоту ошибок, число обнаруженных ошибок, количество символов по сравнению.

На следующем этапе моделирования необходимо произвести настройку функциональных элементов исследуемой системы связи.

Сначала необходимо выполнить настройку генератора бинарных последовательностей Бернулли (Bernoulli Binary Generator) (рис. 3.110):

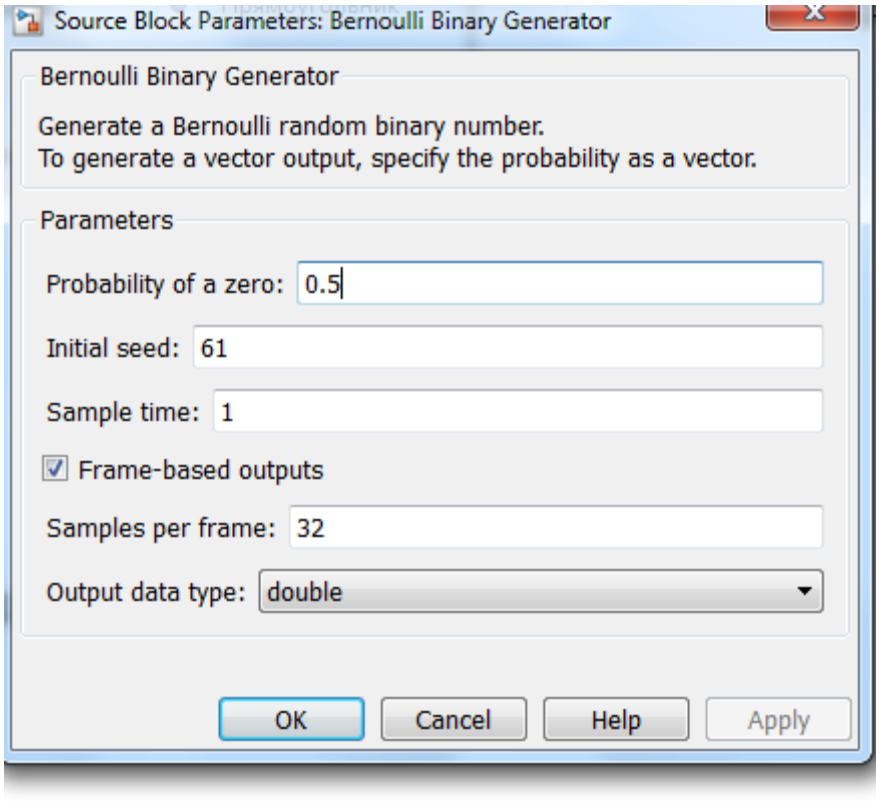


Рисунок 3.110. Настройка Bernoulli Binary Generator для 32 бит

В представленном случае будет использоваться входная последовательность, состоящая из 32 бит

Далее необходимо выполнить настройку турбо-кодера и декодера (рис. 3.111 и рис. 3.112):

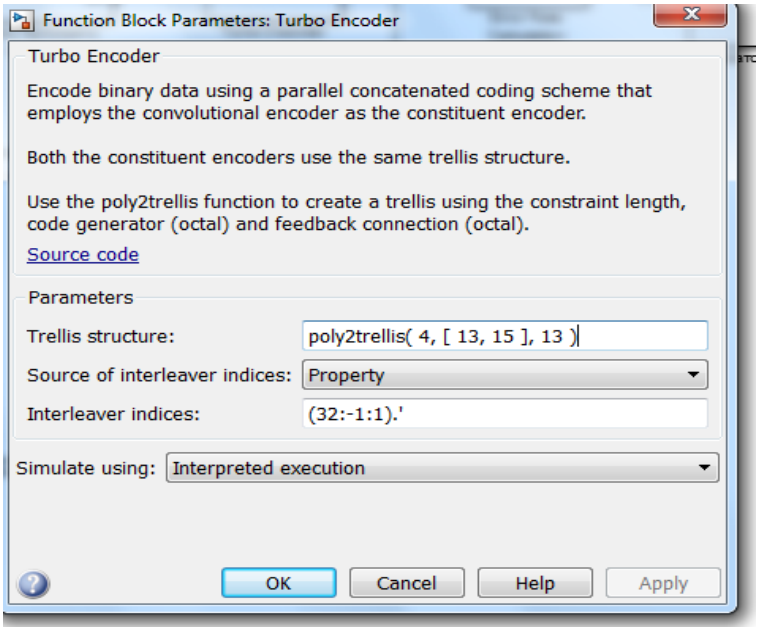


Рис. 3.111. Настройка Turbo Encoder

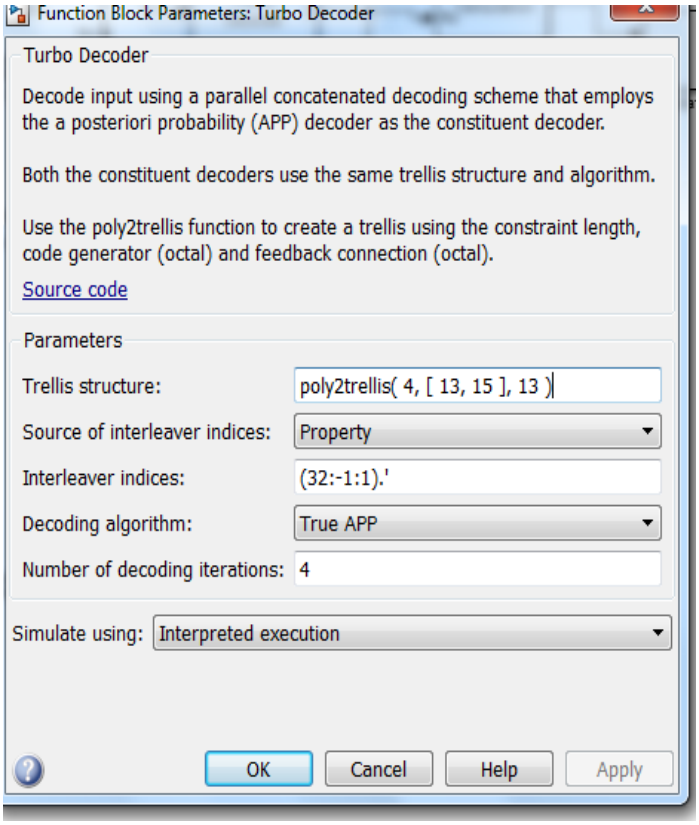


Рис. 3.112. Настройка Turbo Decoder

На следующем шаге необходимо выполнить настройку двоичного канала передачи информации (Binary Symmetric Channel) (рис. 3.113):

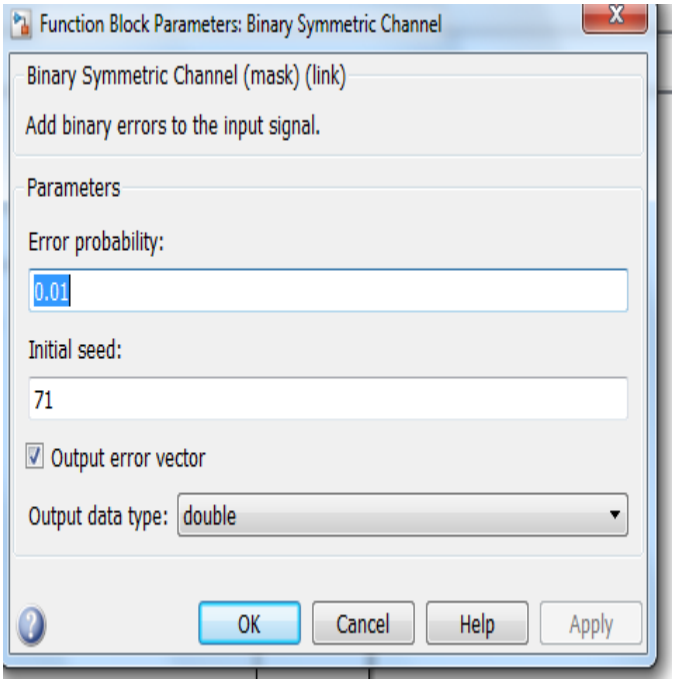


Рис. 3.113. Настройка Binary Symmetric Channel

Стоит отметить, что на данном этапе параметр «вероятность ошибки» необходимо задать малым, для того чтобы проверить работоспособность системы в целом (вероятность ошибки выступает аналогом отношения сигнал/шум в реальном аналоговом канале связи).

Далее настройка анализатора ошибок (Error Rate Calculation) (рис.3.6).

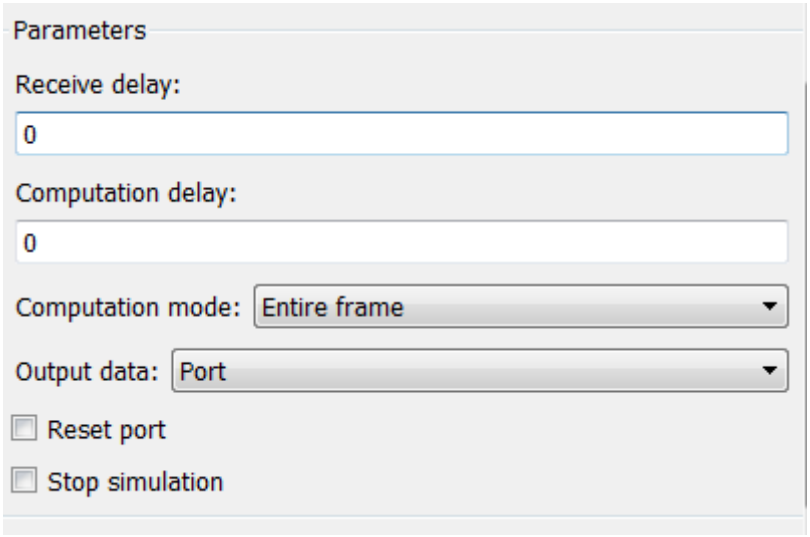


Рис. 3.114. Настройка Error Rate Calculation

Заключительный шаг – запуск работы всей системы. После запуска спроектированной системы будут получены следующие результаты:

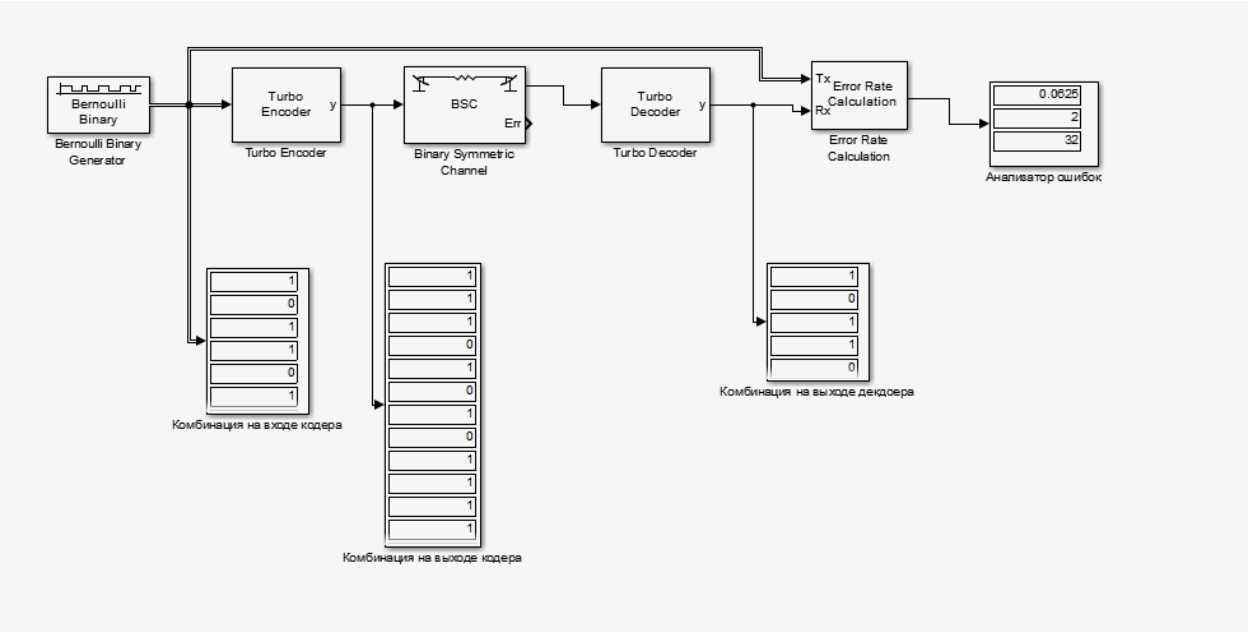


Рис. 3.115. Результаты работы системы связи на основе турбо-кодера

На основе полученных результатов можно сделать вывод, что операции кодирования и декодирования выполнены правильно, следовательно, спроектированная система исправна и работоспособна.

Одной из основных характеристик, характеризующих помехоустойчивость кода, является зависимость количества обнаруженных ошибочных бит от вероятности ошибки. Вероятность ошибки – является аналогом шумов в реальном аналоговом канале передаче. Таким образом для исследования помехоустойчивости турбо-кода представляется целесообразным построить вышеупомянутую зависимость.

Для турбо кода, с входной последовательностью, состоящей из четырех бит и со скоростью кодирования R=1/3 зависимость количества ошибок от вероятности ошибок представлена в таблице 3.7:

Таблица 3.7. Зависимость вероятности ошибок от количества ошибочных бит для турбо кода

$P_{ош}$	0,01	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
N_{bits}	0,062	0,281	0,343	0,375	0,437	0,437	0,406	0,531	0,625	0,562	0,562

Для удобства обработки полученной информации, представляется необходимым и целесообразным представить таблицу 3.7 в виде графика:

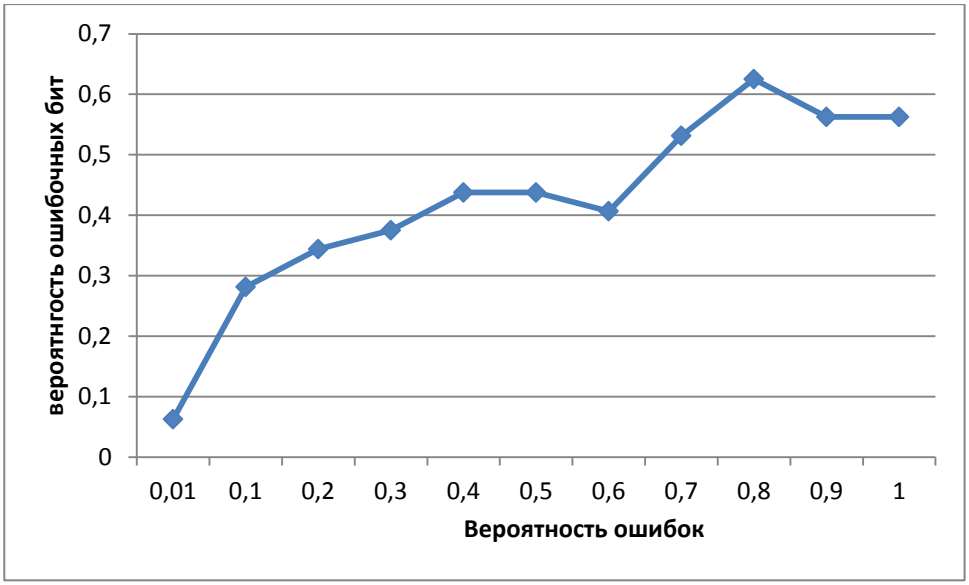


Рис. 3.116. Зависимость вероятности ошибок от количества ошибочных бит

Для турбо кода, с входной последовательностью, состоящей из 64 бит и со скоростью кодирования R=1/3 зависимость количества ошибок от вероятности ошибок представлена в таблице 3.8:

Таблица 3.8. Зависимость вероятности ошибок от количества ошибочных бит для турбо кода

$P_{ош}$	0,01	0,1	0,15	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
N_{bits}	0,250	0,375	0,390	0,468	0,5	0,5	0,5	0,51	0,54	0,57	0,56	0,54

Для удобства обработки полученной информации, представляется необходимым и целесообразным представить таблицу 3.8 в виде графика:

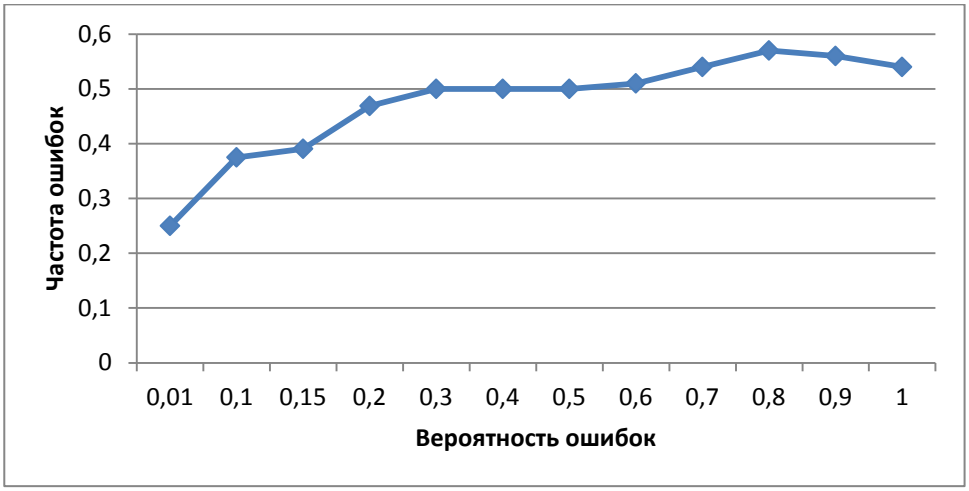


Рис. 3.117. Зависимость вероятности ошибок от количества ошибочных бит

Построим зависимости количества ошибочных битов от вероятности возникновения ошибки (рис. 3.118):

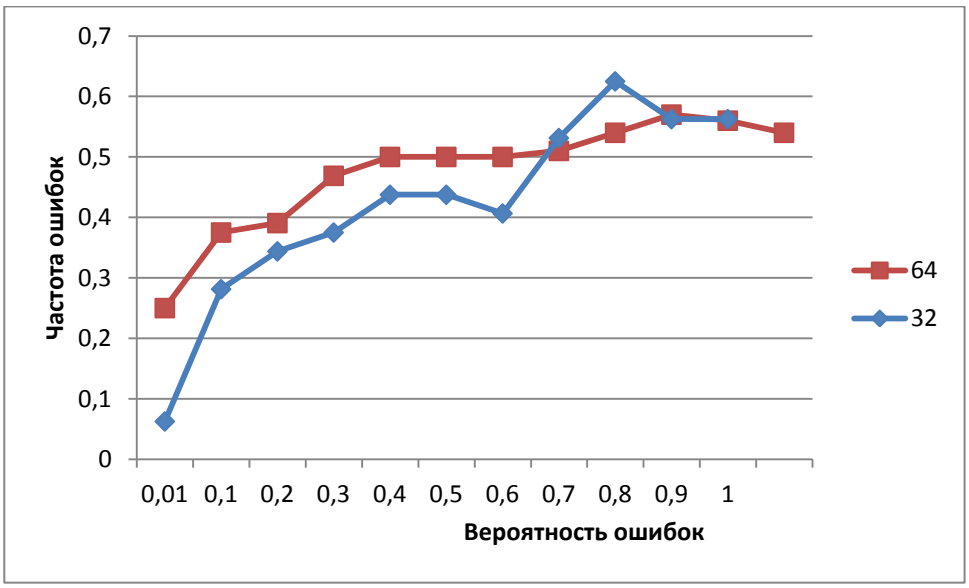


Рис. 3.118. Зависимость количества ошибочных бит

В результате выполнения данной лабораторной работы был проведен теоретический обзор современных алгоритмов использующих для передачи информации по каналам связи турбо-коды.

В ходе работы была разработана модель системы передачи информации с использованием турбо кодов. Кодер представляет собой классический турбо-кодер со скоростью кодирования R = 1/3 реализованный путем параллельного каскадирования двух сверточных кодеров. Декодер представляет собой итеративный декодер турбо-кодов. Основным его достоинством его является то, что на каждой итерации с помощью простых процедур декодирования анализируются данные, принадлежащие простым итерационным кодам.

При помощи разработанной модели было установлено, что турбо-коды с большей длиной кодовых комбинацией обладают лучшей помехоустойчивостью, что соответствует теории.

3.6. Низкоплотностные коды. Классификация LDPC-кодов. Методы построения проверочных матриц. Алгоритмы декодирования низкоплотных кодов. Оценка сложности алгоритмов декодирования на базе MATLAB и LabVIEW

Низкоплотностные коды. Основные положения [16]

В 1948 году Клод Элвуд Шеннон опубликовал свою работу по теории передачи информации. Одним из ключевых результатов работы считается теорема о передачи информации для канала с шумами, которая говорит о возможности свести вероятность ошибки передачи по каналу к минимуму при выборе достаточного большой длины ключевого слова — единицы информации передаваемой по каналу.

Рис. 3.119. Упрощенная схема передачи информации по каналу связи.

При передаче информации её поток разбивается на блоки определённой (чаще всего) длины, которые преобразуются кодером (кодируются) в блоки, называемыми ключевыми словами. Ключевые слова передаются по каналу, возможно с искажениями. На принимающей стороне декодер преобразует ключевые слова в поток информации, исправляя (по возможности) ошибки передачи.

Теорема Шеннона утверждает, что при определённых условиях вероятность ошибки декодирования (то есть невозможность декодером исправить ошибку передачи) можно уменьшить, выбрав большую длину ключевого слова. Однако, данная теорема (и работа вообще) не показывает, как можно выбрать большую длину, а точнее как эффективно организовать процесс кодирования и декодирования информации с большой длиной ключевых слов.

Если предположить, что в кодере и декодере есть некие таблицы соответствия между входным блоком информации и соответствующим кодовым словом, то такие таблицы будут занимать очень много места. Для двоичного симметричного канала без памяти (если говорить упрощённо, то на вход кодера поступает поток из нулей и единиц) количество различных блоков составляет2^n, где n — количество бит (нулей или единиц) которые будут преобразовываться в одно кодовое слово. Для 8 бит это 256 блоков информации, каждый из которых будет содержать в себе соответствующее кодовое слово. Причём кодовое слово обычно большей длины, так как содержит в себе дополнительные биты для защиты от ошибок передачи данных. Поэтому одним из способов кодирования является использование проверочной матрицы, которые позволяют за одно математическое действие (умножение строки на матрицу) выполнить декодирование кодового слова.

Аналогичным образом каждой проверочной матрице соответствует порождающая матрица, аналогичным способом одной операцией умножения строки на матрицу генерирующей кодовой слово.

Таким образом, для сравнительно коротких кодовых слов кодеры и декодеры могут просто содержать в памяти все возможные варианты, или даже реализовывать их в виде полупроводниковой схемы. Для большего размера кодового слова эффективнее хранить порождающую и проверочную матрицу. Однако, при длинах блоков в несколько тысяч бит хранение матриц размером, соответственно, в мегабиты, уже становится неэффективным.

Одним из способов решения данной проблемы становится использования **кодов с малой плотностью проверок на чётность**, когда в проверяющей матрице количество единиц сравнительно мало, что позволяет эффективнее организовать процесс хранения матрицы или же напрямую реализовать процесс декодирования с помощью полупроводниковой схемы.

Коды с низкой плотностью проверок на чётность (LDPC) – это класс линейных блоковых кодов, позволяющих получить превосходную эффективность с относительно малыми вычислительными затратами на их декодирование. Эти коды были предложены

Робертом Галлагером еще в 1963-ем году, однако были забыты на сорок лет в связи со сложностью реализации алгоритмов их декодирования.

Развитие цифровой техники позволило преодолеть многие проблемы и в конце 20-го столетия исследования в области LDPC – кодов получили новый импульс.

LDPC-коды становятся востребованными в системах передачи информации, требующих максимальной скорости передачи при ограниченной полосе частот. Основным конкурентом LDPC-кодов на данный момент являются турбо-коды, которые нашли свое применение в системах спутниковой связи, ряде стандартов цифрового телевидения и мобильных системах связи третьего поколения. Однако LDPC-коды по сравнению с турбо-кодами имеют ряд преимуществ.

Во-первых, LDPC-коды обгоняют турбо-коды по скорости декодирования.

Во-вторых, LDPC-коды более предпочтительны в каналах с меньшими вероятностями ошибок. С развитием методов передачи информации каналы передачи улучшаются, что дает хорошую перспективу для развития LDPC-кодов.

Применение методов итеративного декодирования к данным кодам позволяет практически вплотную приблизиться к пропускной способности канала при относительно небольшой сложности реализации. В связи с этим во многих новых стандартах передачи различного рода данных (DVB-S2, 802.11n, 802.16e) именно LDPC- коды рекомендованы для исправления ошибок. LDPC - коды представляют собой линейные блоковые коды, задаваемые с помощью проверочной матрицы H, характеризуемой относительно малым (<10) числом единиц в строках и столбцах.

В 1981-ом году Р.М. Таннером было предложено использовать двудольные неориентированные графы, впоследствии названные графами Таннера, для описания структуры итеративно декодируемых кодов. В принципе любой блоковый код размерности (M,N), где N – число битов, а M – число проверок в кодовом слове, можно представить в виде двудольного графа Таннера. Например, на рисунке 3.131 изображен такой граф для кода Хэмминга (7,4), проверочная матрица которого имеет вид:

$$H = \begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{matrix}$$

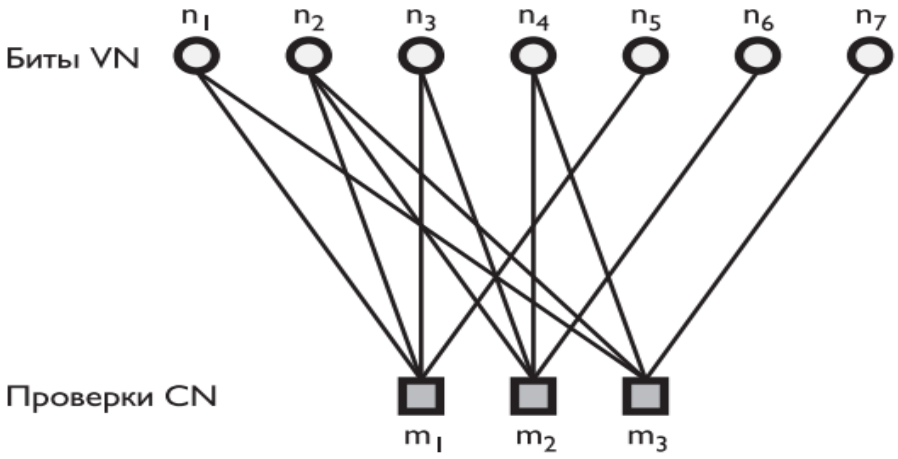


Рис. 3.120. Граф Таннера для кода Хэмминга (7,4)

Вершины графа называются проверочными (check nodes - CN) и битовыми узлами (variable nodes - VN), они обозначены на рисунке m и n соответственно.

При помощи графа Таннера большинство алгоритмов декодирования LDPC-кодов можно представить в виде процессов последовательного обмена сообщениями между соединенными ребрами вершинами. Для проверочных и битовых узлов графа вводится понятие степени – величины, показывающей число рёбер, входящих в рассматриваемый узел. Степени битовых и проверочных узлов обозначаются d_c и d_r соответственно. Если d_c и d_r фиксированы для всех узлов, такой код называют регулярным, а если хотя бы один из этих параметров изменяется от узла к узлу – нерегулярным. Для описания нерегулярных кодов вводится ряд распределения степеней, показывающий долю узлов, имеющих конкретную степень.

В 1996-ом году вышла в свет первая после Р.Галлагера работа, посвященная использованию LDPC – кодов в качестве кодов, способных вплотную приблизиться к границе Шеннона при достаточно большой длине кодового слова. Появление этой статьи породило целую волну исследований, посвященных поиску новых, более эффективных структур LDPC – кодов, а также альтернативных алгоритмов их декодирования с различными соотношениями эффективность/производительность.

В последнее время LDPC – коды получили широкое распространение благодаря превосходной эффективности. Использование LDPC – кодов предусматривает большинство современных стандартов передачи данных (например, стандарты IEEE 802.11, IEEE 802.16), стандартов цифрового вещания (например, стандарты DVB-S2, DVB-T2, DVB-C2).

Классификация LDPC - кодов.

По определению, данному Р. Галлагером, низкоплотностный код — это линейный код, проверочная матрица H которого размерности (M × N) содержит $d_c \ll M$ единиц в каждом столбце и $d_r \ll N$ единиц в каждой строке. Причем распределение единиц по столбцам и строкам в общем случае случайно.

На практике случайное распределение единиц крайне неудобно — для кодирования и декодирования приходиться хранить проверочные и генераторные матрицы, что достаточно накладно, особенно при больших длинах кодов.

Очевидным средством борьбы с этой проблемой является переход к низкоплотностным кодам, проверочная матрица которых обладает какой-то структурой. Простейший вариант структуризации проверочной матрицы — использование циклических кодов.

Формально проверочная матрица такого кода представляет собой циркулянтную матрицу размерности N × N, в которой каждая строка получается циклическим сдвигом вправо предыдущей строки. Значение влияния цикличности проверочной матрицы на сложность декодера LDPC-кода сложно переоценить, поскольку каждая из строк матрицы однозначно определяется предыдущей строкой, в связи с чем реализация декодера может быть существенно упрощена по сравнению со случайной структурой проверочной матрицы. Как известно, кодер циклического кода достаточно просто реализовать с использованием сдвигового регистра и набора сумматоров.

К недостаткам циклических кодов можно отнести фиксированный, для всех скоростей кодирования, размер проверочной матрицы N × N, что подразумевает более сложный декодер, а также высокий Хэммингов вес строк, что усложняет структуру декодера. Дополнительно стоит заметить, что циклический код — всегда регулярный.

К достоинствам помимо упрощения кодирования/декодирования следует отнести большое минимальное расстояние и очень низкий порог при итеративном декодировании.

Желание преодолеть недостатки циклических LDPC-кодов привело к появлению квазициклических LDPC-кодов. Квазициклические коды также имеют хорошую структуру, позволяющую упростить кодер и декодер. В дополнение к этому они позволяют более гибко подойти к разработке кода, в частности позволяют синтезировать нерегулярные коды. Проверочная матрица такого кода представляет собой не что иное, как набор циркулянтных подматриц:

$$H = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & & \vdots \\ A_{M1} & \cdots & A|_{MN} \end{bmatrix}$$

Очевидно, что для получения низкоплотностного кода, циркулянтные матрицы должны быть разреженными, что на практике означает использование в качестве циркулянтов единичных матриц. Для того чтобы получить нерегулярный код, какие-то подматрицы просто объявляются нулевыми.

Методы построения проверочных матриц

Методы построения LDPC-кодов также можно разбить на классы. К первому классу относятся все алгоритмические способы и способы, использующие вычислительную технику. А ко второму — способы, основанные на теории графов, математике конечных полей, алгебре и комбинаторике.

При этом стоит заметить, что первый класс методов позволяет получать как случайные, так и структурированные LDPC-коды, в то время как второй нацелен на получение только структурированных LDPC-кодов, хотя бывают и исключения.

В отличие от других линейных блоковых кодов, таких как БЧХ или кодов Рида–Соломона, имеющих строгий алгоритм синтеза кодов с заданными параметрами, для LDPC-кодов существует множество способов построения кодов.

Существуют способы построения LDPC-кодов, предложенные Галлагером и МакКеем, о которых ниже и пойдет речь.

Для начала будет рассмотрен метод, предложенный Р. Галлагером.

Р. Галлагер предложил следующий алгоритм построения низкоплотностных кодов. Пусть проверочная матрица кода имеет вид:

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{d_c} \end{bmatrix}$$

В ней подматрицы H_a , a = 1, 2… d_c имеют структуру, которая может быть описана следующим образом.

Для любых двух целых μ и d_r, больших 1, каждая подматрица H_a имеет размерность $(\mu \times \mu \cdot d_r)$, при этом веса строк этой подматрицы - d_r, а столбцов – 1. Подматрица H_1 имеет специфическую форму: для i=0,1,…, μ-1 i-ая строка содержит все d_r единиц на позициях с i от d_r до $(i + r) \cdot r - 1$. Очевидно, что результирующая матрица H – регулярная матрица размерности $\mu \cdot d_c \times \mu \cdot d_r$ с весами строк и столбцов d_r и d_c соответственно.

Важной характеристикой матрицы LDPC-кода является отсутствие циклов определенного размера (кратности). Под циклом кратности 4 понимается наличие в двух разных столбцах проверочной матрицы ненулевых элементов на совпадающих позициях. Отсутствие цикла кратности 4 определяется вычислением скалярного произведения столбцов матрицы: если всевозможные скалярные произведения всех столбцов матрицы не превосходят 1, то это означает отсутствие в матрице циклов кратности 4. Цикл кратности 4 является минимально возможным и встречается существенно чаще циклов большей длины (6, 8, 10 и т. д.). Присутствие в матрице LDPC-кода циклов любой кратности свидетельствует о заложенной в структуру матрицы избыточности, не приводящей к улучшению помехоустойчивых свойств кода. Пример циклов кратности 4 приведен на рисунке 3.121

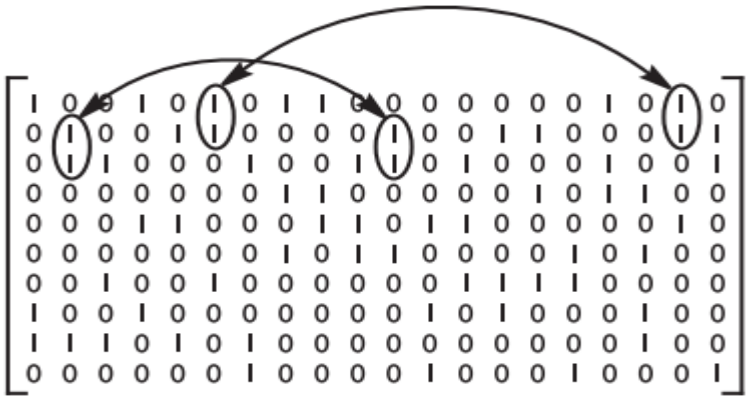


Рис. 3.121. Циклы кратности 4 в матрице LDPC - кода

Рассмотренный выше алгоритм не гарантирует отсутствие циклов кратности 4, однако они могут быть удалены впоследствии. Галлагер показал, что ансамбль таких кодов обладает прекрасными свойствами. Также была показана возможность реализации достаточно простых кодеров, поскольку проверочные биты такого кода могут быть найдены по проверочной матрице кода как функция информационных узлов.

По прошествии тридцати пяти лет МакКей, будучи незнакомым с работой Галлагера, повторно открыл преимущества кодов с разреженными матрицами, и был первым, кто при помощи компьютерного моделирования показал возможность этих кодов вплотную приблизиться к границе Шеннона как для двоичного симметричного канала, так и для канала с аддитивным белым гауссовским шумом. МакКей предложил несколько компьютерных алгоритмов построения проверочных матриц низкоплотностных кодов. Приведем некоторые из них в порядке увеличения сложности реализации.

Проверочная матрица H синтезируется путём случайного генерирования столбцов веса d_c и настолько это возможно, равномерным распределением весов строк d_r.

Проверочная матрица H синтезируется путём случайного генерирования столбцов веса d_c и строк веса d_r, с дополнительной проверкой на отсутствие циклов кратности 4.

Проверочная матрица H синтезируется по алгоритму 2, с дополнительным удалением циклов кратности 4.

Проверочная матрица H синтезируется по алгоритму 3, с дополнительным условием, что проверочная матрица имеет вид H = $[H_1 H_2]$, где H_2 – обратимая матрица.

Недостатком алгоритмов МакКея является отсутствие какой-либо структуры в проверочных матрицах, что усложняет процесс кодирования.

Кодирование осуществляется приведением матрицы H к виду H = $[P^T I]$, из которого можно получить генераторную матрицу в систематической форме G = [PI]. Проблема при кодировании по матрице G заключается в том, что подматрица P, в общем случае, не является разреженной. То есть для кодов, представляющих интерес, сложность кодирования оказывается достаточно высокой.

Декодирование LDPC – кодов.

Наибольший интерес для исследователей представляет процедура декодирования, ввиду того что она является более время затратной и ресурсоемкой.

Декодирование – это процедура поиска и исправления ошибки, наложенной каналом на кодовое слово, по принятому из канала вектору или собственно поиск кодового слова по вектору, принятому из канала.

Декодирование по максимуму правдоподобия кода $\boldsymbol{C}$ обозначает нахождение по заданному принятому вектору $\boldsymbol{y}$ такого кодового слова $\boldsymbol{c}$ из $\boldsymbol{C}$ (множества всех кодовых слов), которое максимизирует вероятность того, что передавалось слово $\boldsymbol{c}$ при условии принятия вектора $\boldsymbol{y}$. Задача декодирования по максимуму правдоподобия является NP-полной.

Для оценки качества работы различных декодеров используется оценка вероятности ошибки декодирования (BER) на информационный бит, вычисляемая как отношение количества ошибочных информационных бит после декодирования к общему количеству переданных информационных бит. итеративные схемы декодирования кодов с низкой плотностью проверок на четность не являются декодерами по максимуму правдоподобия, но

позволяют получить разумный баланс по сложности и вероятности ошибки декодирования по сравнению с декодированием по максимуму правдоподобия. итеративное декодирование подразумевает, что нахождение кодового слова будет производиться не за один проход, а за несколько, с последовательным уточнением результата на каждом шаге. Применяются следующие основные схемы декодирования: «жёсткое» декодирование, быстрое декодирование, многопороговое декодирование.

«Жесткое» декодирование – это схема декодирования для двоичного симметричного канала при небольшом количестве ошибок в канале. «Жесткое» декодирование инвертированием битов – самая простая схема декодирования кодов с низкой плотностью проверок на четность.

Под проверкой понимается любая строка h = $\{h_0, h_1, \dots, h_{N-1}\}$ из проверочной матрицы кода с низкой плотностью проверок на чётность. Будем говорить, что проверка для некоторого вектора y = $\{y_0, y_1, \dots, y_{N-1}\}$ выполняется тогда, когда скалярное произведение вектора $\boldsymbol{y}$ на проверку даёт нуль. Будем говорить, что элемент y_i принятого вектора $\boldsymbol{y}$ участвует в проверке h = $\{h_0, h_1, \dots, h_{N-1}\}$ тогда, когда соответствующий элемент проверки h_i не равен нулю.

Одна итерация «жёсткого» декодирования инвертированием битов производится следующим образом:

1. Для принятого вектора вычисляются все проверки.

2. Если некоторый бит принятого вектора участвовал более чем в половине не выполнившихся проверок, бит инвертируется.

3. После такого анализа всех символов принятого вектора вектор проверяется на принадлежность коду. Если вектор является кодовым словом, декодирование заканчивается, в противном случае выполняется следующая итерация алгоритма.

Такая процедура декодирования применима для кодов с низкой плотностью проверок на четность потому, что большинство проверок в таком случае будут содержать одну ошибку или не будут содержать ошибок вообще и тогда невыполнение большого количества проверок для символа принятого слова будет обозначать наличие в нем ошибки.

Сложность одной итерации «жесткого» декодирования инвертированием бит является линейной, количество итераций декодирования обычно выбирается около $\log_2(N)$, гдеN – длина кодового слова.

Декодирование по вероятностям является «мягким» декодированием, т.е. декодированием на основе вектора, состоящего не из дискретных значений (0 и 1), а из вещественных величин, полученных на выходе канала путем пересчета вероятностей (англ. belief propagation decoding).

На основе принятого из канала вектора формируются два (для двоичного случая) вектора вероятностей того, что в принятом векторе на данной позиции находился заданный символ.

Каждому ненулевому элементу проверочной матрицы кода с низкой плотностью проверок на чётность приписываются две величины: $q_{i,j}^{x}$ и $r_{i,j}^{x}$. Величина $q_{i,j}^{x}$ является вероятностью того, что j-ый символ принятого вектора имеет значение x по информации, полученной из всех проверок, кроме i-й. Величина $r_{i,j}^{x}$ является вероятностью того, что проверка i выполняется, если j-ый символ принятого вектора равен x, а все остальные символы проверок имеют распределение вероятностей, заданное величинами $\{q_{i,j}^{x}: j$ из N(i)/j$\}$, где N(i) – множество символов, входящих в i-ую проверку.

Перед началом работы алгоритму требуется инициализация, далее алгоритм работает по принципу пересчета вероятностей символов принятого вектора (belief propagation), используя для пересчета вероятностей правило Байеса для апостериорной вероятности события. Одна итерация алгоритма представляет собой следующую последовательность действий:

1. Для всех проверок вычисляются величины $\Delta r_{i,j}^{x}$ и пересчитываются вероятности $r_{i,j}^{x}$ для x = $\{0,1\}$.
2. Для всех символов принятого вектора пересчитываются вероятности $q_{i,j}^{x}$.
3. Формируются векторы псевдоапостериорной вероятности q_j^0 и q_j^1.
4. Формируется вектор решения c' по следующему правилу: c'_j=1, если $q_j^1 > S$, иначе 0.

Если вектор c' является кодовым словом, декодирование заканчивается, в противном случае выполняется следующая итерация алгоритма.

Сложность данного алгоритма выше, чем сложность «жесткого» декодирования инвертированием битов, но качество декодирования повышается за счет использования дополнительной информации на выходе канала. Однако точность работы такого алгоритма зависит от инициализации: чем точнее она произведена, тем точнее будет конечный результат. Для канала с гауссовским шумом инициализация может быть произведена при

помощи информации о дисперсии шума в канале. Для других распределений шума в канале или при неизвестных характеристиках шума точная инициализация алгоритма может оказаться сложной задачей.

Несмотря на то что декодирование пересчетом вероятностей является эффективным методом для каналов с непрерывным выходом, тот факт, что сложность его значительно выше, чем сложность «жесткого» декодирования, создает предпосылки для поиска более быстрых алгоритмов декодирования, обладающих приемлемым качеством.

Среди известных алгоритмов быстрого декодирования кодов с низкой плотностью проверок на четность для каналов с непрерывным выходом наиболее известен алгоритм «min-sum», являющийся упрощением декодера «belief propagation», а также алгоритм UMP (Uniformly Most Powerful).

Сложность декодера UMP (быстрого декодирования по надежностям) значительно ниже, чем сложность декодера, пересчитывающего вероятности, за счет того, что пересчет надежностей выполняется по упрощенной схеме (схеме «взвешенного» мажоритарного голосования, в качестве «весов» используется надежность проверок), а также за счет возможности использования исключительно целочисленных операций сложения и сложения по модулю два. также к достоинствам быстрого декодера по надежностям можно отнести то, что декодеру не требуется знать характеристики шума в канале (дисперсию и т. д.), следовательно, такой декодер может работать в любом симметричном канале с двоичным входом.

Недостатком быстрого декодера по надежностям является оценка вероятности ошибки декодирования, которая для канала с аддитивным гауссовским шумом оказывается на 0,5 дБ хуже, чем вероятность ошибки декодирования вероятностного декодера.

Далее будет рассмотрено многопороговое декодирование.

Основная идея многопорогового декодирования по надежностям состоит в том, чтобы изменять значения порогов инвертирования символов от одной итерации к другой следующим образом: на первых итерациях порог инвертирования символов выбирается так, чтобы количество инвертированных символов было минимальным (вплоть до инвертирования только одного символа на первой итерации); на последующих итерациях пороги инвертирования постепенно повышаются.

При многопороговом декодировании, если на первой итерации была исправлена хотя бы одна ошибка, декодирование на последующих итерациях становится значительно проще и общее качество декодирования улучшается. По-прежнему для работы декодеру не требуется информация о шуме в канале, достаточно лишь задать надежности.

Декодер, работающий по многопороговой схеме, позволяет получить вероятность ошибки декодирования на 0,1–0,4 дБ лучше, чем обеспечивает быстрый декодер по надежностям UMP, практически приближаясь к вероятности ошибки, получаемой при вероятностном декодировании кодов с низкой плотностью проверок на четность. Помимо независимости от характеристик канала многопороговый декодер обладает свойством декодеров кодов с низкой плотностью проверок на четность, а именно универсальностью и применимостью для любой конструкции таких кодов.

Следует отметить, что эффективность нерегулярных LDPC-кодов оказывается выше эффективности регулярных кодов. это объясняется тем, что в нерегулярных кодах из-за различного числа единиц в строках и столбцах информационные символы защищены по разному. В результате при декодировании проявляется так называемый эффект волны, когда более защищенные биты декодируются быстрее и затем как бы помогают при декодировании менее защищенных бит.

Экспериментальная часть

Задание на лабораторную работу:

1. Изучить работу приложения «LDPC Кодер-Декодер»;
2. Установить параметры матрицы, выставив значения n=100, m=50, j=3, Message Bits = 1000;
3. Меняя значение параметра Noise выписать показания контроллеров BER с использованием кодирования и без него
4. По полученным данным построить соответствующие графики.
5. Привести рисунки, демонстрирующие исходное сообщение, закодированное сообщение, декодированное сообщение и разницу между исходным и декодированным сообщением.

В таблице 3.7 приведены данные, которые были получены в результате эксперимента.

Таблица 3.9. Данные эксперимента

Noise		0	0,0	0,0	0,0	0,1	0,1	0,1	0	0

		,02	4	6	8		2	4	,16	,18
BER(coding)		0,001	0,007064	0,018164	0,030273	0,076832	0,087948	0,125126	1	1
BER(NO_coding)		0,0242	0,0565	0,0777	0,0843	0,1129	1	1	1	1

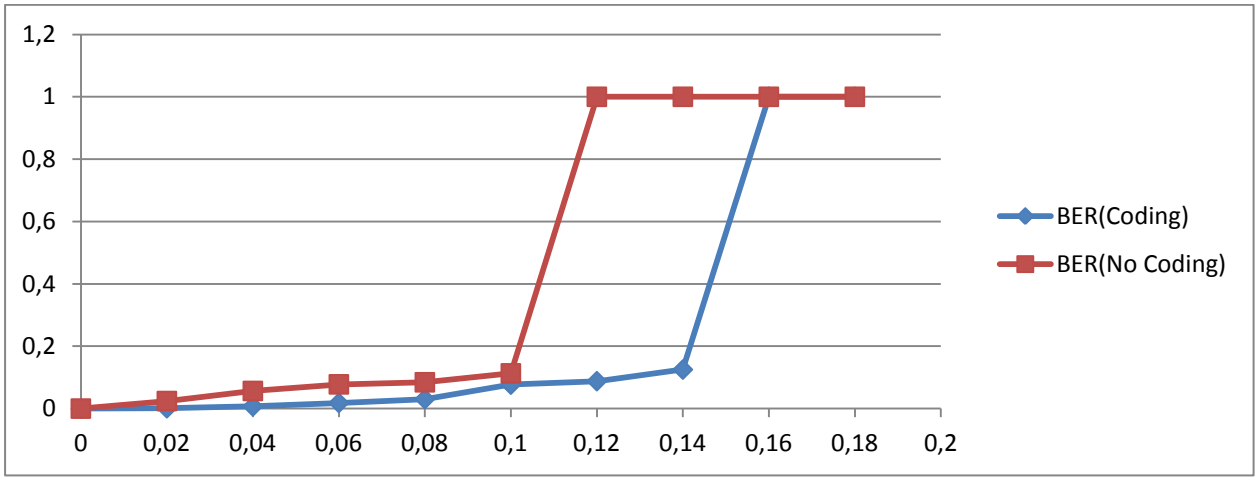


Рис. 3.122. График зависимости BER от Noise

На данном графике видно, что быстрее единицы достиг счётчик, который считал BER в канале без кодирования LDPC.

На рисунках 3.123, 3.124, 3.125 и 3.126 представлены примеры работы LDPC кодера-декодера [17].

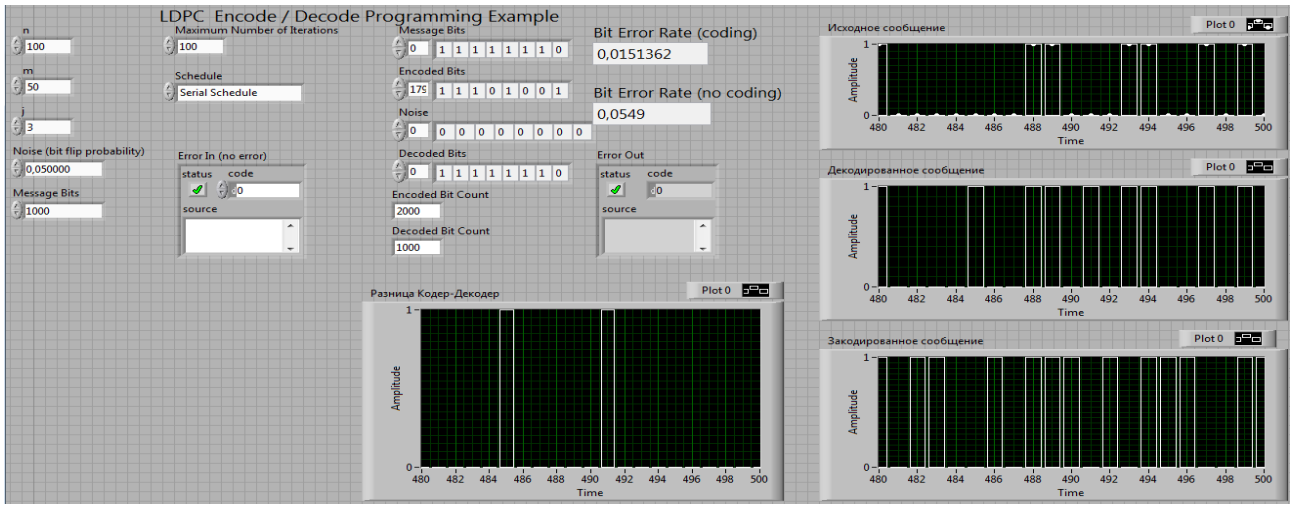


Рис.3.123. Панель управления аппаратно-программного комплекса LDPC кодера-декодера

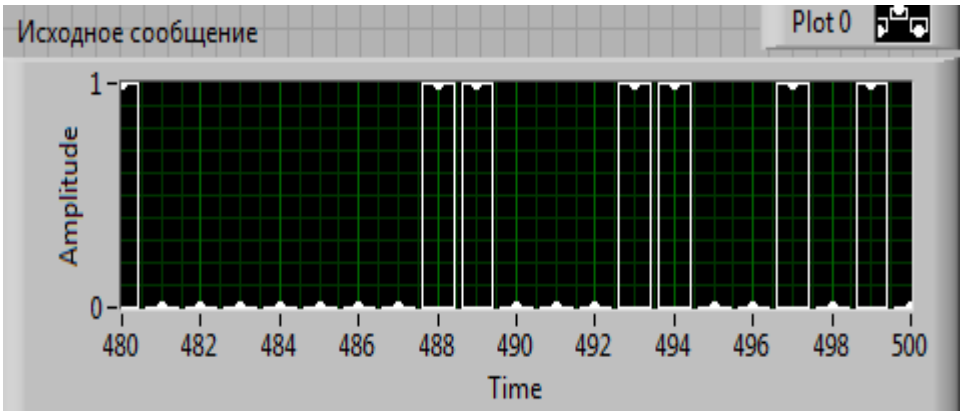


Рис. 3.124. Исходное сообщение

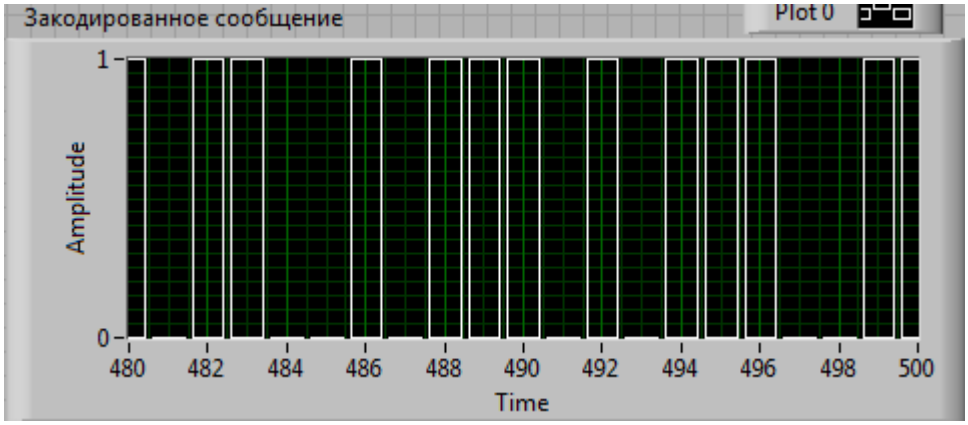


Рис. 3.125. Закодированное сообщение

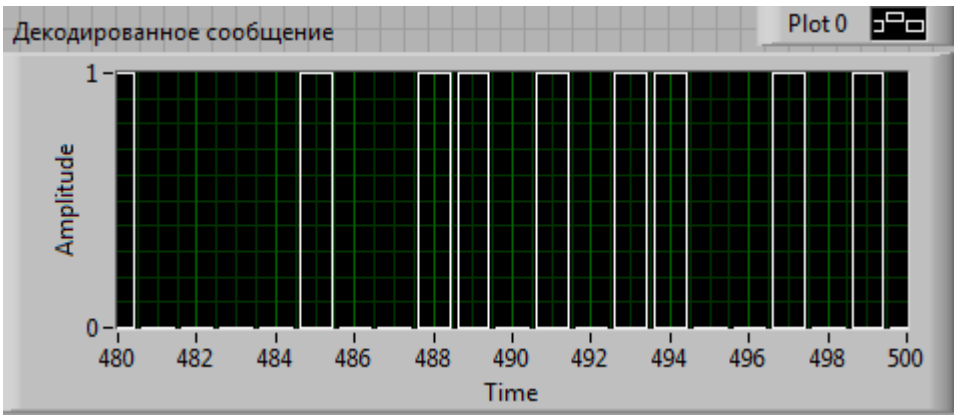


Рис. 3.126. Декодированное сообщение

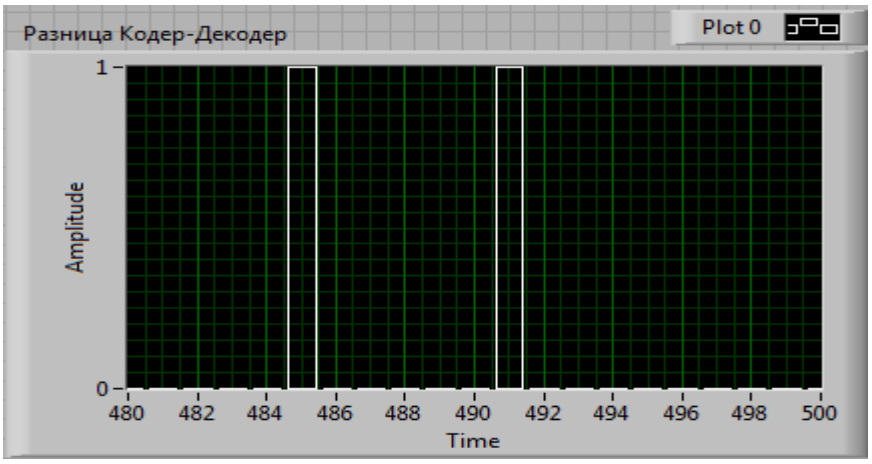


Рис. 3.127. Разница между исходным и декодированным сообщением

В ходе данной лабораторной работы были закреплены навыки работы с ПО LabVIEW, изучен учебный аппаратно-программный комплекс для визуализации и исследования методов канального кодирования/декодирования в беспроводных системах цифрового вещания и связи. Данный комплекс позволяет проводить исследование кодов LDPC. Были сделаны выводы о том, что в канале с кодированием LDPC стопроцентная ошибка возникнет лишь при значении шума, равном 0,16. Тогда как в канале без кодирования стопроцентная ошибка достигается уже при 0,1.

Моделирование LDPC кодов в MATLAB Simulink

В рабочем поле необходимо собрать схему для работы кода LDPC. Схема представлена на рисунке 3.128 [21]

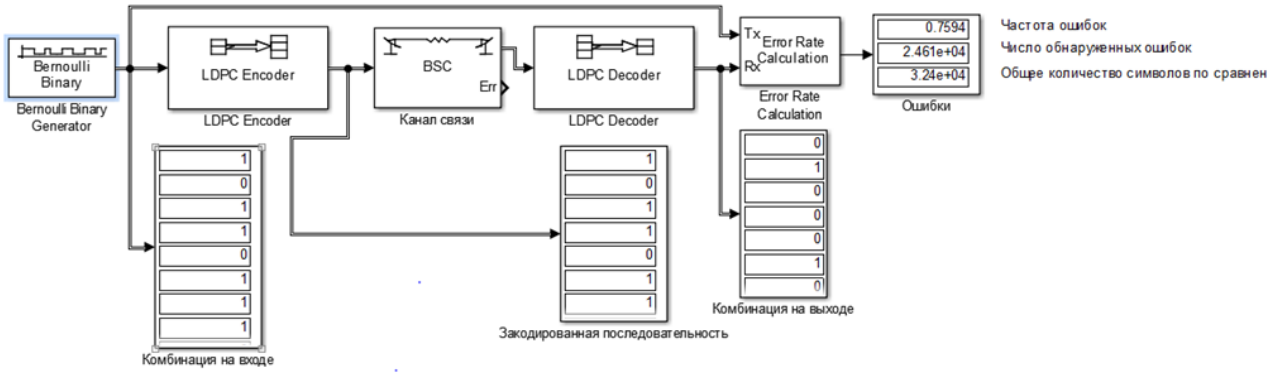


Рис.3.128 Линия передачи с применением код LDPC

В состав линии с кодированием входят:

7. Bernoulli Binary Generator
8. LDPC Encoder
9. Binary Symmetric Channel (канал передачи)

10.LDPC Decoder

11.Error Rate Calculation (анализатор ошибок)

12.Display

Данные блоки необходимо найти в окне «Simulink Library», представленной на рисунке 11. Для упрощения поиска, можно воспользоваться окошком поиска элементов.

Характеристики блоков выставить следующие:

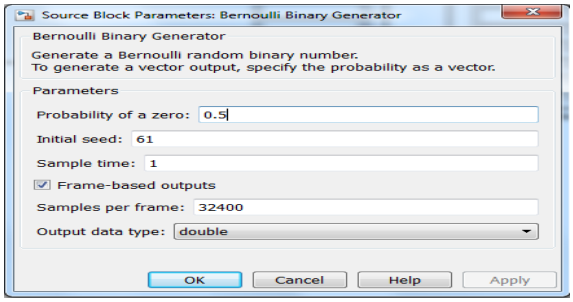


Рис.3.129 Bernoulli Binary Generator

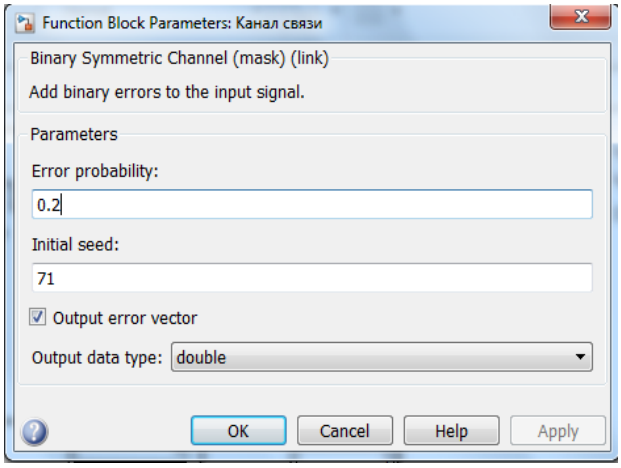


Рис.3.130. LDPC Encoder

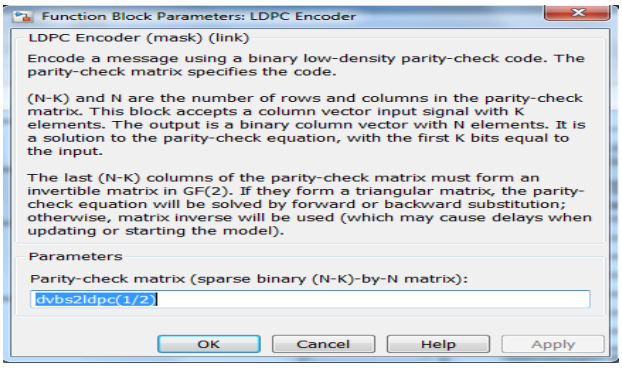


Рис.3.131. Binary Symmetric Channel

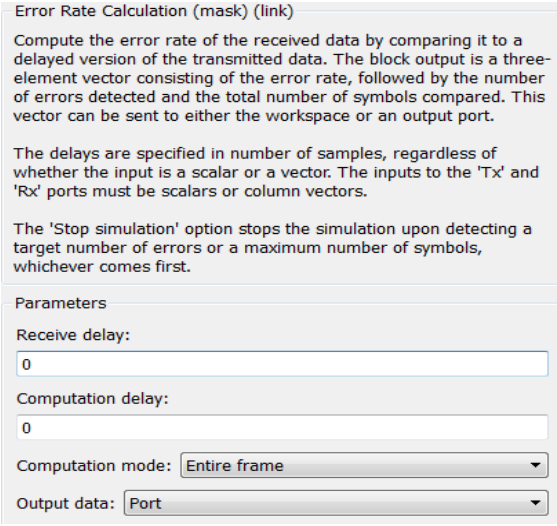


Рис.3.132. BCH Decoder

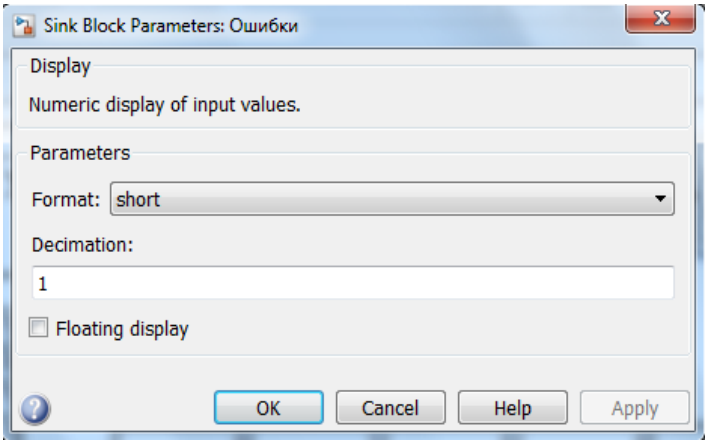


Рис.3.133. Error Rate Calculation (анализатор ошибок)

Представим полученные результаты:

1	0	1	1	0	1	1	1
0	0	0	1	1	0	1	1
0	0	0	0	1	1	0	1
0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	0	0	1	1	0	1
1	1	0	1	1	0	1	1
0	1	1	0	0	0	0	1
1	1	1	0	0	0	0	0
0	1	0	1	1	1	1	0
1	0	1	1	1	0	0	1
0	1	0	1	0	0	1	1
1	1	0	0	1	0	0	0
0	1	0	1	0	1	0	1
1	1	1	1	1	0	0	1
1	0	1	1	1	1	0	0
0	0	0	0	0	0	1	1
0	0	1	1	1	0	0	1
1	0	1	1	1	1	1	1
0	0	0	0	1	0	0	1
0	1	0	1	1	0	1	1
0	1	1	1	0	0	0	1
0	1	1	1	0	1	0	1

Рис. 3.134. Комбинация на входе

1	0	1	1	0	1	1	1
0	0	0	1	1	0	1	1
1	0	1	1	0	1	1	1
0	0	0	1	1	0	1	1
0	0	0	0	1	1	0	1
0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	0	0	1	1	0	1
1	1	0	1	1	0	1	1
0	1	1	0	0	0	0	1
1	1	1	0	0	0	0	0
0	1	0	1	1	1	1	0
1	0	1	1	1	0	0	1
0	1	0	1	0	0	1	1
1	1	0	0	1	0	0	0
0	1	0	1	0	1	0	1
1	1	1	1	1	0	0	1
1	0	1	1	1	1	0	0
0	0	0	0	0	0	1	1
0	0	1	1	1	0	0	1
1	0	1	1	1	1	1	1
0	0	0	0	1	0	0	1
0	1	0	1	1	0	1	1
0	1	1	1	0	0	0	1
0	1	1	1	0	1	0	1

Рис. 3.135. Закодированная последовательность

0	1	0	0	0	1	0	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	1	0
1	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	0	1	1	0	0	0
0	0	0	0	0	1	0	1
0	0	1	0	1	0	0	0
0	0	1	1	0	1	0	0
1	0	1	1	0	0	1	0
0	0	0	1	1	1	1	0
0	1	1	0	0	0	1	1
0	1	0	1	0	0	0	0
1	0	1	0	0	0	0	0
0	0	1	1	1	1	1	0
0	1	1	0	1	0	1	1
1	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0
1	1	1	1	0	0	1	0
1	0	0	0	1	1	1	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	0
1	0	0	0	1	1	0	0
0	0	0	0	1	0	0	0

Рис. 3.136. Последовательность на выходе

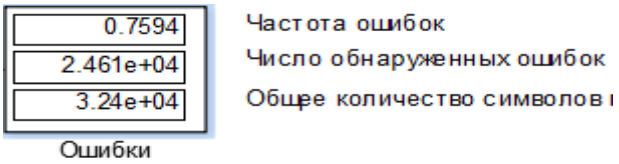


Рис. 3.137. Работа блока оценки ошибки в канале

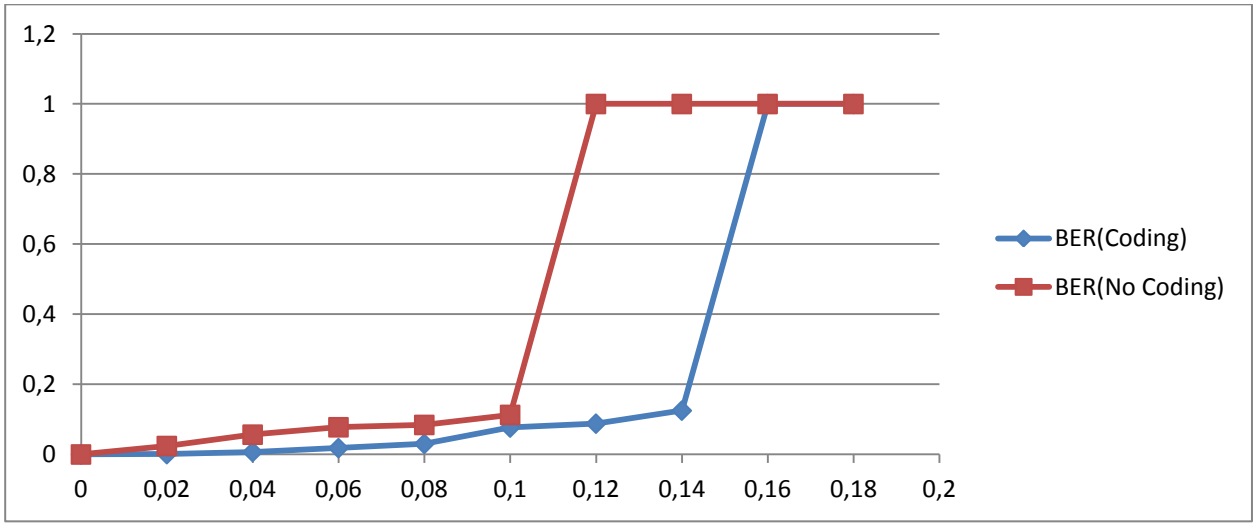


Рис. 3.138 График зависимости BER от Noise

3.7. Исследование каскадных кодов

Каскадными называют коды, в которых кодирование осуществляется в два уровня; имеется внутренний и внешний коды, с помощью которых и достигается желаемая надёжность передачи сообщений. Внутренний код связан с модулятором. Демодулятор, как правило, настраивается для исправления большинства канальных ошибок. Внешний код, чаще всего высокоскоростной (с низкой избыточностью), снижает вероятность появления ошибок до заданного значения. Основной причиной использования каскадного кода является низкая степень кодирования и общая сложность реализации, меньшая той, которая потребовалась бы для осуществления отдельной процедуры кодирования.

В одной из наиболее популярных систем каскадного кодирования для внутреннего кода применяется сверточное кодирование по алгоритму Витерби, а для внешнего — код Рида-Соломона с чередованием между двумя этапами кодирования [1]. Функционирование таких систем при E_b / N_0, находящемся в пределах от 0,2 до 2,5 дБ, для достижения $P_B = 10^{-5}$ реально достижимо в прикладных задачах. В этой системе демодулятор выдает мягко квантованные кодовые символы на внутренний свёрточный декодер, который, в свою

очередь, выдает жестко квантованные кодовые символы с пакетными ошибками на декодер Рида-Соломина.

Внешний код Рида-Соломона образуется из *m*-битовых сегментов двоичного потока данных. Производительность такого (недвоичного) кода Рида-Соломона зависит только от числа *символьных ошибок* в блоке. Код не искажается пакетами ошибок внутри *m*-битового символа. Иными словами, для данной символьной ошибки производительность кода Рида-Соломона такова, как если бы символьная ошибка была вызвана одним битом или *m* бит. Тем не менее производительность каскадных систем несколько ухудшается за счет коррелирующих ошибок в последовательных символах. Поэтому чередование между кодированиями нужно выполнять на уровне символов (а не битов). Работа [22] представляет собой обзор каскадных кодов, которые были разработаны для дальней космической связи.

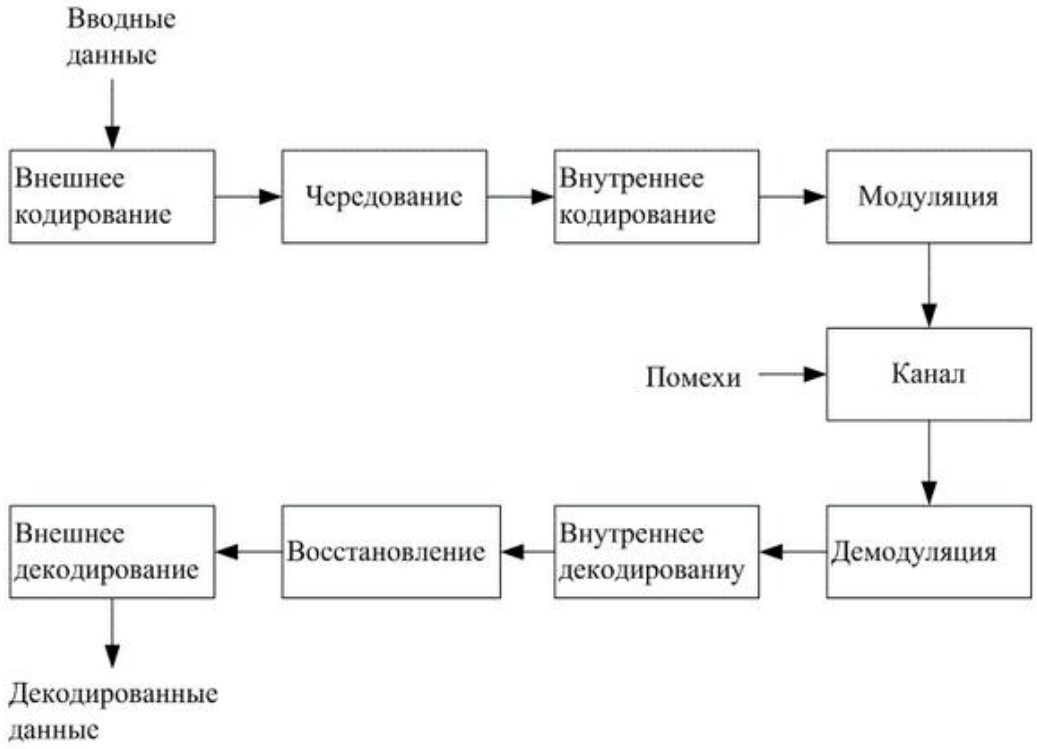


Рис. 3.139. Обобщенная структурная схема исследования каскадных кодов

Однако в данной работе исследуется случай, когда в качестве внутреннего кодирования используются турбо-коды.

Турбо-код – параллельный каскадный блоковый систематический код, способный исправлять ошибки, возникающие при передаче информации.

Схема каскадного кодирования впервые была предложена Форни [23] как метод получения высокоэффективного кода посредством комбинаций двух или более компонуемых кодов (иногда называемых составными). В результате, такие коды могут корректировать

ошибки в значительно более длинных кодах и имеют структуру, которая позволяет относительно легко осуществить декодирование средней сложности. Последовательные каскадные коды часто используются в системах с ограничением мощности, таких как космические зонды. Самая распространенная из этих схем содержит внешний код Рида-Соломона (выполняется первым, убирается последним), который следует за сверточным внутренним кодом (выполняется последним, убирается первым). Турбо-код можно считать обновлением структуры каскадного кодирования с итеративным алгоритмом декодирования связанной кодовой последовательности.

Турбо-коды впервые были введены в 1993 году Берру, Главье и Цитимаджимой. В описываемой схеме достигалась вероятность появления ошибок 10^{-5} при степени кодирования 1/2 и модуляции BPSK в канале с белым аддитивным гауссовым шумом с E_b / N_0, равным 0,7 дБ. Коды образуются посредством компоновки двух или более составных кодов, являющихся разными вариантами чередования одной и той же информационной последовательности. Тогда как для сверточных кодов на финальном этапе декодер выдает жестко декодированные биты (или в более общем случае — декодированные символы), в каскадной схеме, такой как турбо-код, для хорошей работы алгоритм декодирования не должен ограничивать себя, подавая на декодеры жесткую схему решений. Для лучшего использования информации, получаемой с каждого декодера, алгоритм декодирования должен применять, в первую очередь, мягкую схему декодирования, вместо жесткой. Для систем с двумя составными кодами концепция, лежащая в основе турбо-декодирования, заключается в том, чтобы передать мягкую схему принятия решений с выхода одного декодера на вход другого и повторять эту процедуру до тех пор, пока не будут получены надежные решения.

Турбокоды представляют собой сравнительно новый тип кодов для исправления ошибок, возникающих при передаче цифровой информации по каналам связи с шумами. Впервые они были введены в рассмотрение в 1993 году и сразу же привлекли к себе пристальное внимание специалистов в области помехоустойчивого кодирования. Причина этому – уникальная способность турбокодов обеспечивать характеристики помехоустойчивости передачи информации, близкие к теоретически достижимым значениям при умеренной сложности реализации кодеков. Разработка турбокодов развивается по двум направлениям: свёрточные турбокоды, образованные путём параллельного соединения двух или более свёрточных кодеров, и блочные турбокоды, образованные путём последовательного соединения двух или более блочных кодеров. Как показали исследования, блочные турбокоды являются более эффективными при относительно высоких кодовых скоростях.

Кодирование

На рисунке 3.140 представлена структурная схема турбо-кодера:

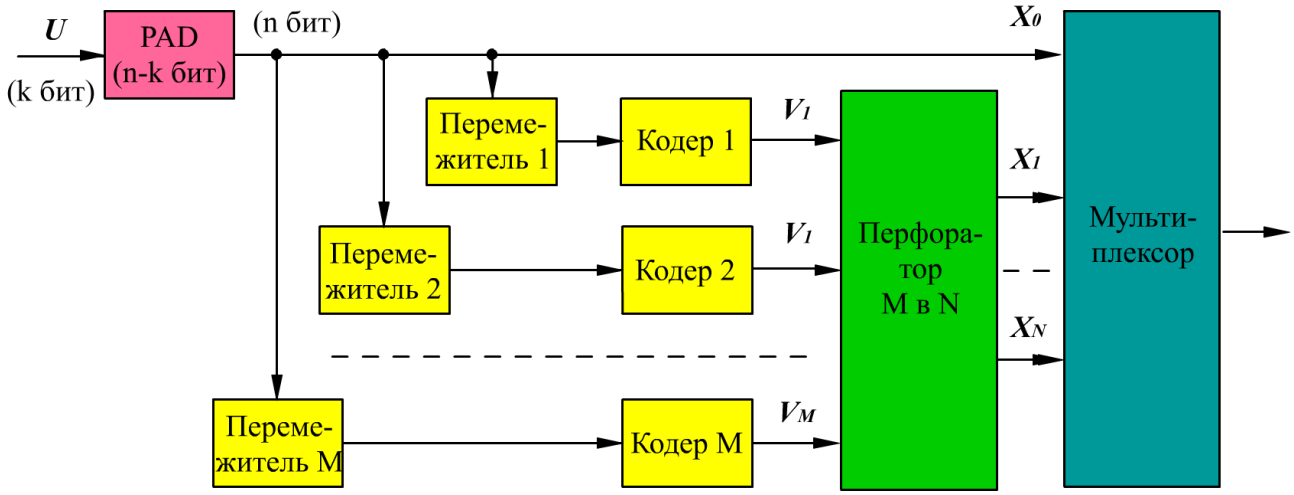


Рис. 3.140. Структурная схема турбо-кодера

Сначала на вход формирователя пакетов (PAD) поступает блок данных U длиной k бит. В формирователе пакетов к данным прибавляется ещё (n-k) дополнительных бит служебной информации, соответствующих используемому стандарту формирования пакета и включающих в себя символы его начала и окончания. То есть получается пакет X_0, состоящий из n бит.

Далее последовательность бит X_0 поступает параллельно на M ветвей, содержащих последовательно соединённые перемежитель и компонентный кодер. Таким образом X_0 используется в качестве входных данных сразу всеми компонентными кодерами.

В перемежителях по псевдослучайному закону происходит перемешивание поступающих бит. В отличие от посимвольного прямоугольного перемежителя, используемого в кодах Рида-Соломона, в турбо-кодах используется перемежение отдельных бит, которое подобно случайным перестановкам. Причём впоследствии, при операциях декодирования этот закон перемежения будет считаться известным. Полученные последовательности поступают на входы кодеров.

Задача перемежителя — преобразовать входную последовательность так, чтобы комбинации бит X_0, соответствующие кодовым словам с низким весом (весом называется число ненулевых бит кодового слова) на выходе первого кодера, были преобразованы в комбинации, дающие кодовые слова с высоким весом на выходах остальных кодеров. Таким образом кодеры получают на выходе кодовые слова с различными весами. При кодировании

формируются кодовые слова так, чтобы получалось максимально возможное среднее расстояние между ними (расстоянием между двумя кодовыми словами называется число бит, в которых они различаются). Из-за того, что кодовые блоки формируются из почти независимых частей, на выходе турбо-кодера среднее расстояние между кодовыми словами больше, чем минимальное расстояние для каждого компонентного кодера, а, следовательно, растёт эффективность кодирования.

Кодовая скорость — отношение длины кодового блока на входе к длине преобразованного кодового блока на выходе кодера.

В отсутствие перфоратора исходная последовательность X_0 мультиплексируется с последовательностями проверочных бит, образуя кодовое слово, подлежащее передаче по каналу.

Для увеличения кодовой скорости применяется выкалывание (перфорация) определённых проверочных битов выходной последовательности. Таким образом кодовая скорость возрастает.

Если учесть, что турбо-коды оперируют с блоками большой длины с $k > 10000$, то $k \approx n$.

С помощью перфоратора, выкалывая разное число проверочных бит, возможно регулирование кодовой скорости. То есть можно построить кодер, адаптирующийся к каналу связи. При сильном зашумлении канала перфоратор выкалывает меньше бит, чем вызывает уменьшение кодовой скорости и рост помехоустойчивости кодера. Если же канал связи хорошего качества, то выкалывать можно большое число бит, вызывая рост скорости передачи информации.

Алгоритм декодирования по максимуму апостериорной вероятности

Алгоритм Бала дает «мягкую» оценку достоверности декодированного бита. То есть предъявляет на выходе степень доверия результату декодирования. В противоположность «жёсткой» структуре, при которой на выходе декодера формируется лишь наиболее вероятное значение декодированного бита («0» или «1»), при вынесении «мягкого» решения используется более подробная дискретизация выходного сигнала, характеризующая вероятность корректного приема бита. Благодаря использованию «мягких» решений в турбо-декодерах оказывается эффективным использование нескольких итераций декодирования. Апостериорная информация, полученная о кодовом слове на выходе первой итерации декодирования, поступает на вход блока следующей итерации и является для него уже априорной вероятностью. Такой подход позволяет улучшать качество декодирования от итерации к итерации. Таким образом, изменяя число итераций декодирования, можно

адаптировать декодер к текущему состоянию канала передачи и достичь требуемой вероятности ошибки на бит.

На рисунке 3.141 представлена структурная схема одной итерации турбо-декодера при двухкаскадном кодировании:

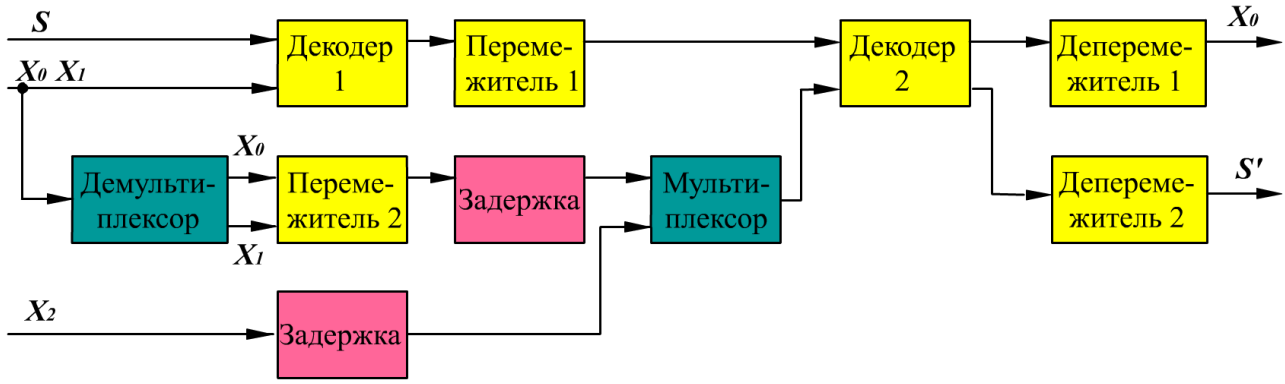


Рисунок 3.141. Структурная схема одной итерации турбо-декодера при двухкаскадном кодировании

На рис. 3.141 для простоты понимания представлен вариант схемы одной итерации турбо-декодирования при двухкаскадном кодировании. Эта схема несложно обобщается на случай произвольного количества каскадов кодирования.

Декодер для одной итерации содержит каскадное соединение двух элементарных декодеров, каждый из которых, основываясь на критерии максимума апостериорной вероятности, выносит «мягкое» решение о переданном символе. На первый декодер первой итерации с выхода демодулятора поступают «мягкие» решения символов последовательностей X_0 и X_1. Таким образом, на выходе первого декодера появляется оценка информационного символа, которая после последующего перемежения попадает на вход второго декодера и является для него априорной информацией. Используя «мягкое» решение о последовательности X_2, второй декодер формирует свою оценку.

На рисунке 3.142 представлена структурная схема трёх итераций турбо-декодера при двухкаскадном кодировании:

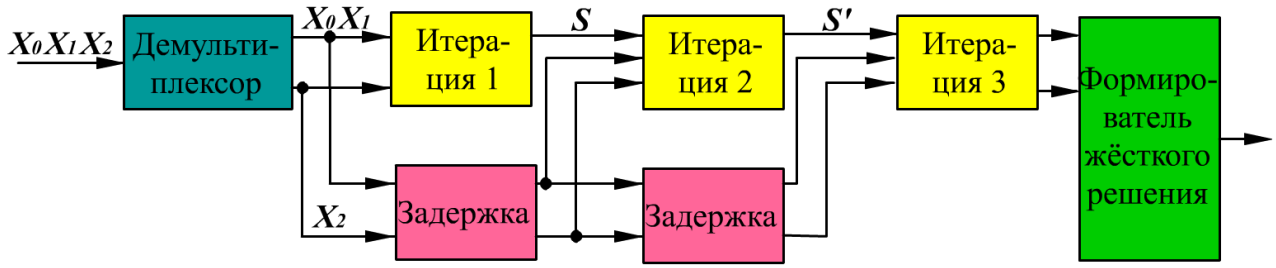


Рис. 3.142. Структурная схема трёх итерации турбо-декодера при двухкаскадном кодировании

С выхода каждой итерации решение переходит на вход следующей. От итерации к итерации происходит уточнение решения. При этом каждая итерация работает с «мягкими» оценками и на выход отдает также «мягкие». Поэтому такие схемы получили название декодеров с мягким входом и мягким выходом. Процесс декодирования прекращается либо после выполнения всех итераций, либо когда вероятность ошибки на бит достигнет требуемого значения. После декодирования из полученного «мягкого» решения производится окончательное «жёсткое».

Преимущества и недостатки турбо-кодов

Преимущества

Среди всех практически используемых современных методов коррекции ошибок турбо-коды и коды с низкой плотностью проверок на чётность наиболее близко подходят к границе Шеннона, теоретическому пределу максимальной пропускной способности зашумленного канала. Турбо-коды позволяют увеличить скорость передачи информации, не требуя увеличения мощности передатчика, или они могут быть использованы для уменьшения требуемой мощности при передаче с заданной скоростью. Важным преимуществом турбо-кодов является независимость сложности декодирования от длины информационного блока, что позволяет снизить вероятность ошибки декодирования путём увеличения его длины.

Недостатки

Основной недостаток турбо-кодов — это относительно высокая сложность декодирования и большая задержка, которые делают их неудобными для некоторых применений. Но, например, для использования в спутниковых каналах этот недостаток не является определяющим, так как длина канала связи сама по себе вносит задержку, вызванную конечностью скорости света.

Ещё один важный недостаток турбо-кодов — сравнительно небольшое кодовое расстояние (то есть минимальное расстояние между двумя кодовыми словами в смысле выбранной метрики). Это приводит к тому, что, хотя при большой входной вероятности ошибки (то есть в плохом канале) эффективность турбо-кода высока, при малой входной вероятности ошибки эффективность турбо-кода крайне ограничена. Поэтому в хороших каналах для дальнейшего уменьшения вероятности ошибки применяют не турбо-коды, а LDPC-коды. Хотя сложность используемых алгоритмов турбо-кодирования и недостаток открытого программного обеспечения препятствуют внедрению турбо-кодов, в настоящее время многие современные системы используют турбо-коды.

Применение турбо-кодов

Компании France Telecom и Telediffusion de France запатентовали широкий класс турбо-кодов, что ограничивает возможность их свободного применения и, в то же время, стимулирует развитие новых методов кодирования таких, как, например, LDPC.

Турбо-коды активно применяются в системах спутниковой и мобильной связи, беспроводного широкополосного доступа и цифрового телевидения. Турбо-коды утверждены в стандарте спутниковой связи DVB-RCS. Турбо-коды также нашли широкое применение в мобильных системах связи третьего поколения (стандарты CDMA2000 и UMTS).

Моделирование каскадных кодов в MATLAB Simulink [21]

Виртуальная модель передачи данных с исправлением ошибок при помощи каскадного кода была реализована в среде Simulink Matlab. Модель демонстрирует работу кодера Рида-Соломона (внешний код) и кодера Турбо-кодов (внутренний код), позволяет исследовать исправляющую способность кодов для разных видов модуляции и сравнить её характеристики с работой указанных выше кодеров в отдельности.

На рисунке 3.143 приведена разработанная модель:

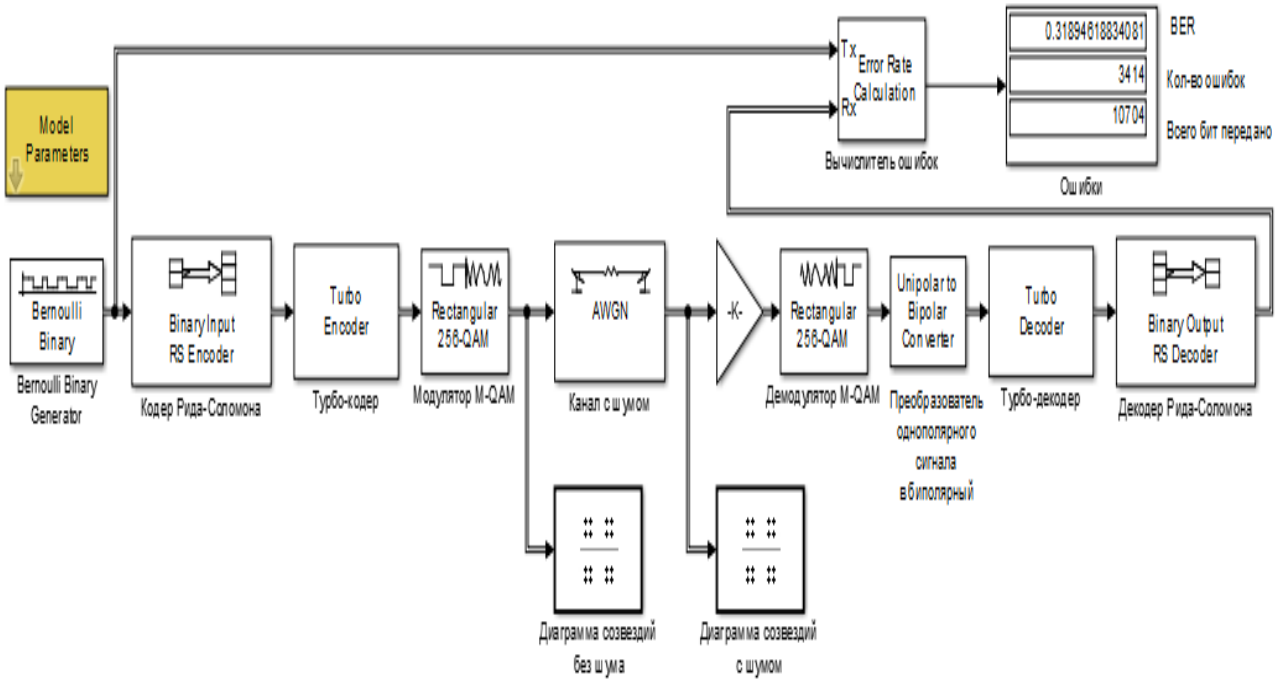


Рис. 3.143. Разработанная модель исследования каскадных кодов

В её основу положены следующие элементы, встроенные в библиотеку Simulink:

- Bernoulli Binary Generator
- Binary Input RS Encoder
- Turbo Encoder
- Rectangular QAM Modulator Baseband
- AWGN Channel
- Rectangular QAM Demodulator Baseband
- Unipolar to Bipolar Converter
- Turbo Decoder
- Binary Output RS Decoder
- Error Rate Calculation
- Discrete Time Scatter Plot Scope
- Gain
- Display (Дисплей, отражающий ошибки)

Далее представлено описание основных блоков:

Bernoulli Binary Generator (генератор псевдослучайной последовательности) – генерирует случайную бинарную последовательность (рисунок 3.144).

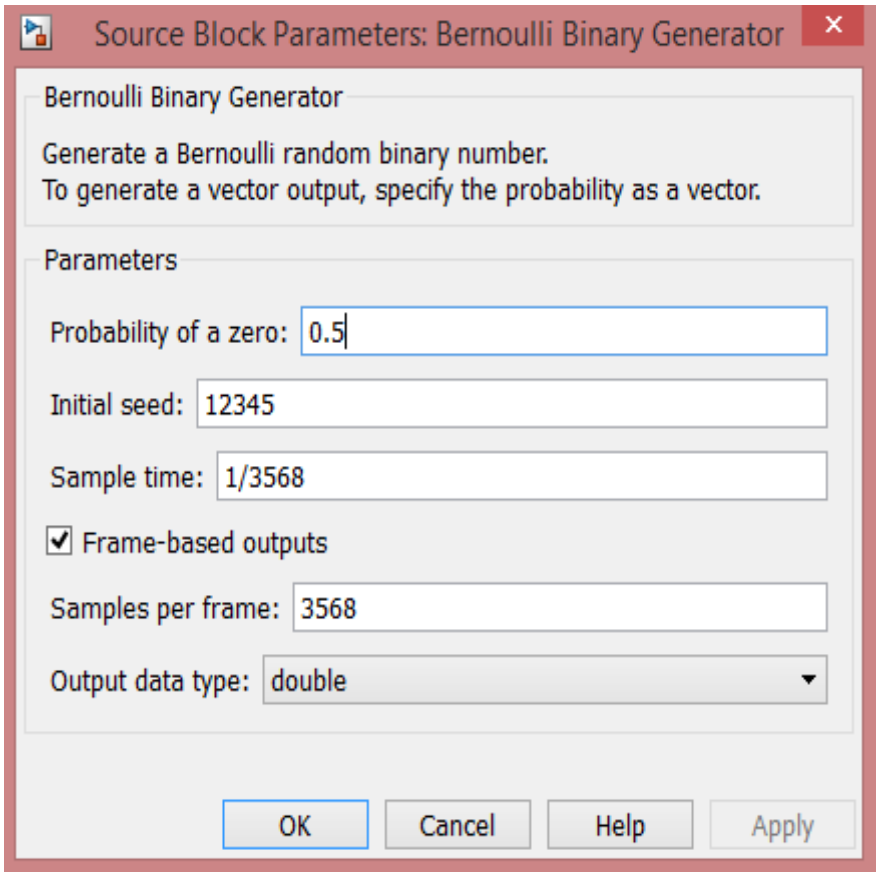


Рис. 3.144. Параметры блока «Bernoulli Binary Generator»

«Probability of a zero» - вероятность появления нуля;

«Initial seed» - начальное значение для генерации;

«Sample time» - длительность сэмпла;

«Samples per frame» - размер фрейма.

Binary Input RS Encoder – кодер Рида-Соломона (рисунок 3.7).

«Codeword length N» - общее количество бит;

«Message length K» - количество информационных бит.

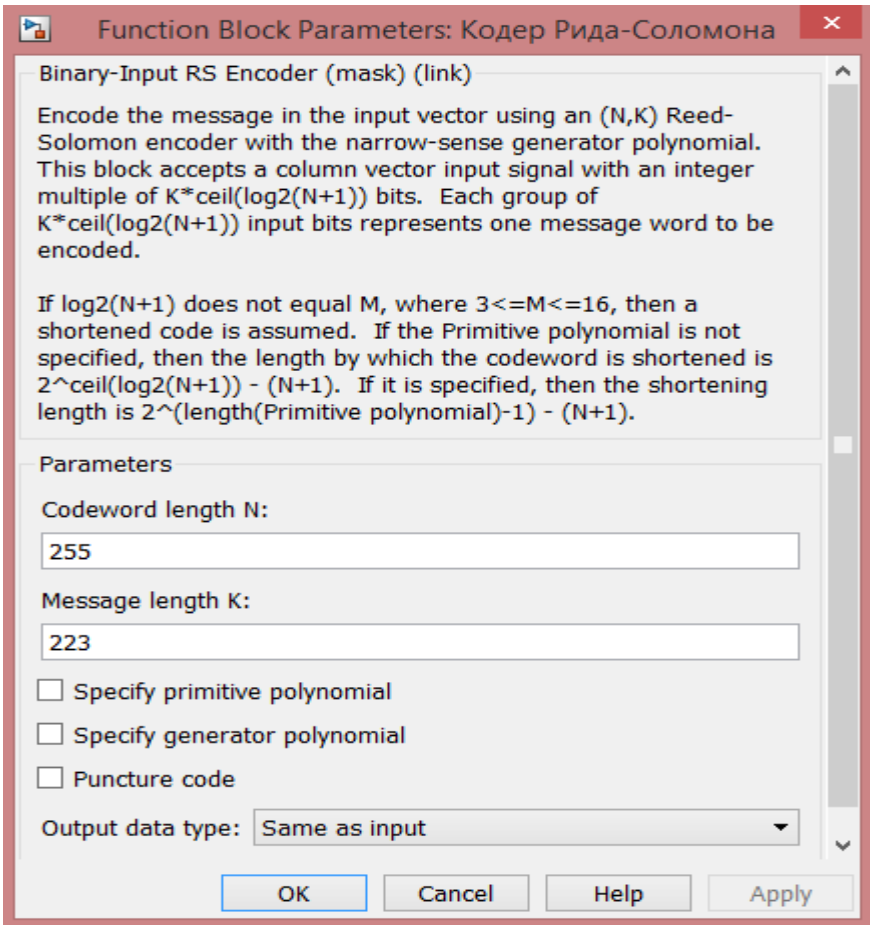


Рис. 3.145. Параметры блока «Binary Input RS Encoder»

Данный блок имеет следующую структуру внутри (рисунок 3.8):

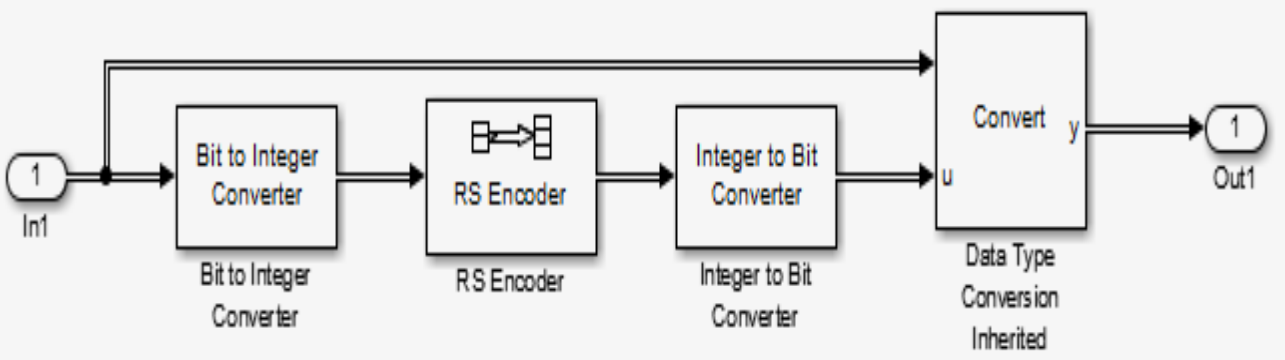


Рис. 3.146. Состав блока «Binary Input RS Encoder»

Таким образом, информационные биты, поступающие со входа генератора преобразуются в тип «Integer», кодируются и преобразуются обратно, затем полученные биты конвертируются в тот тип данных, который изначально был на входе кодера.

Turbo Encoder – Турбо-кодер (рисунок 3.147).

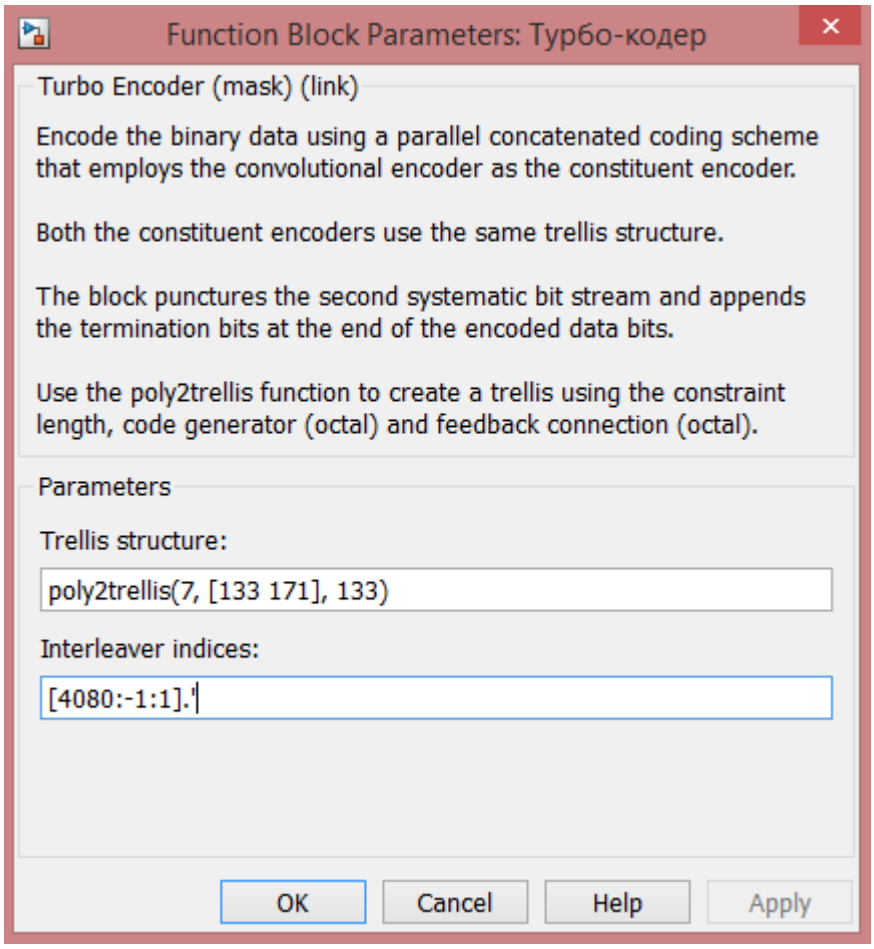


Рис. 3.147. Параметры блока «Turbo Encoder»

«Trellis structure» - структура треллис-модуляции.

Треллис-модуляция (TCM – Trellis Coded Modulation) представляет собой способ, который позволяет обеспечить повысить скорость передачи сообщения с сохранением уровня помехоустойчивости. Этот способ отличается тем, что помехоустойчивое кодирование и тип модуляции используются совместно. Выбранная соответствующим образом пара помехоустойчивый код – способ модуляции часто также носит название сигнально-кодовая конструкция (СКК).

Для кодирования использован один из наиболее часто употребляемых свёрточных кодов – код (171,133,7), который кодирует последовательность со скоростью 1/2.

«Interleaver indices» - входные параметры перемежителя.

Схема турбокодера имеет следующую структуру (рисунок 3.148):

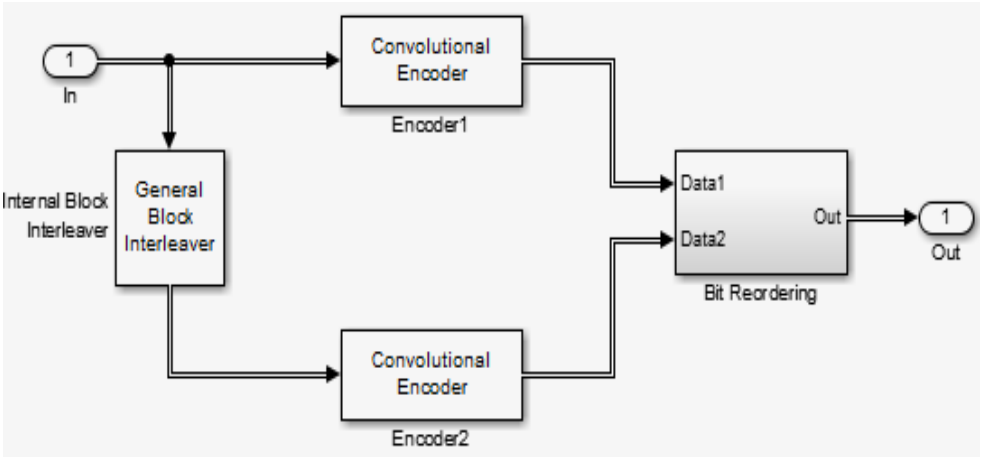


Рис. 3.148. Структура блока «Turbo Encoder»

Поток бит распараллеливается на два. В первом случае биты поступают на свёрточный кодер (133, 171), а во втором потоке биты сначала проходят перемежитель, затем поступают на аналогичный свёрточный кодер. Блок «Bit Reordering» выстраивает биты в последовательный поток.

Rectangular QAM Modulator Baseband – модулятор QAM-M (рисунок 3.149).

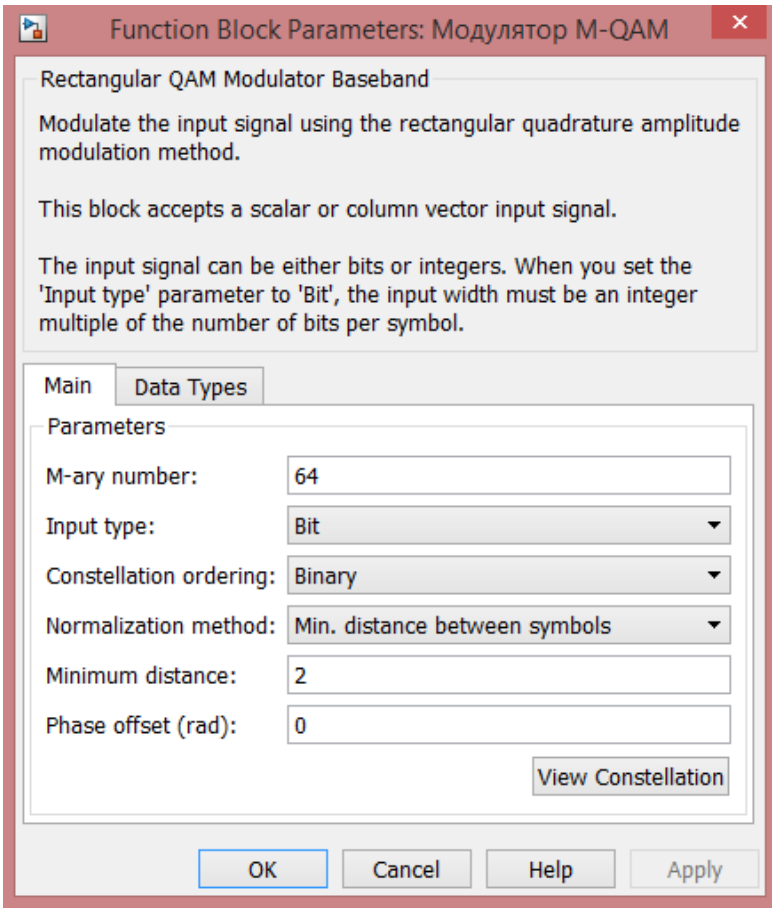


Рис. 3.149. Параметры блока «Turbo Encoder»

«M-ary number» - количество позиций в QAM-M;

«Input type» - тип входных данных;

«Constellation ordering» - порядок построения созвездия (рисунок 3.150).

Остальные параметры по умолчанию.

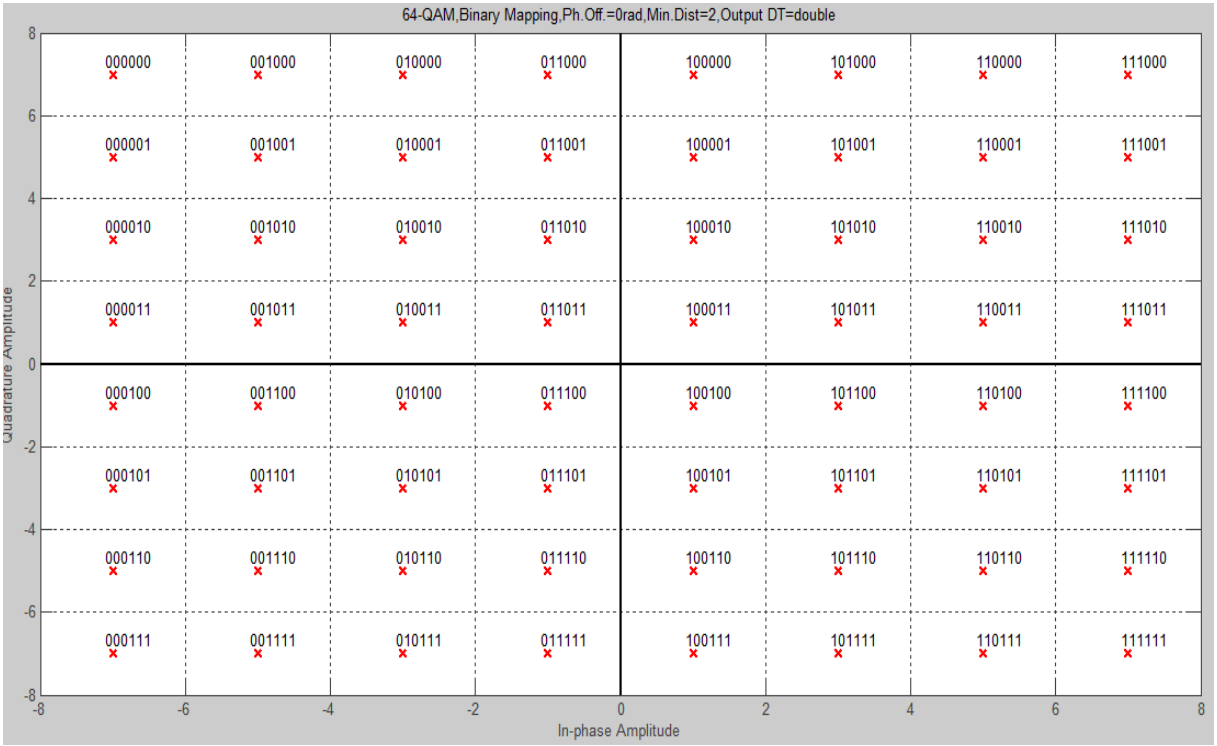


Рис. 3.150. Диаграмма построения созвездий

AWGN Channel (Канал связи) – добавляет «белый» гауссовский шум в канале (рисунок 3.13).

«Variance» - считывает параметр Eb/N0 из блока Model Parameters.

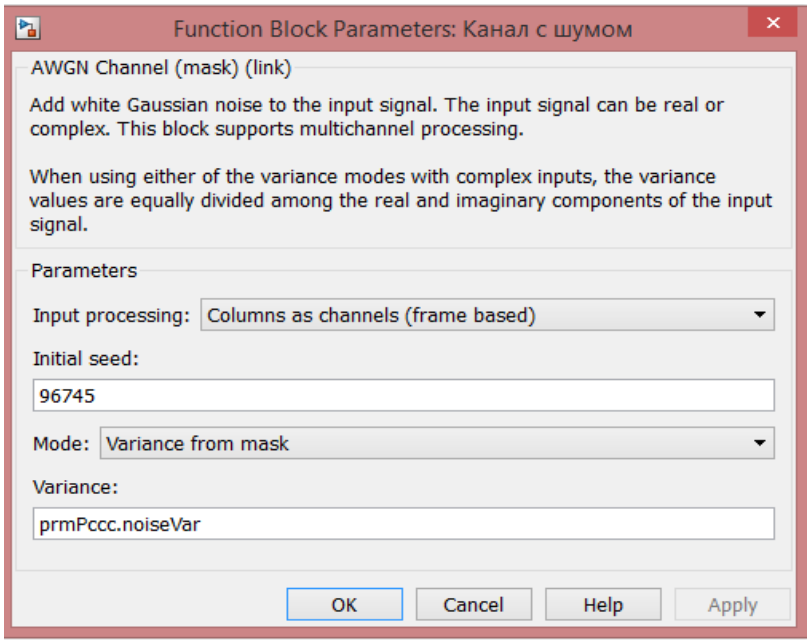


Рис. 3.151. Параметры блока «AWGN»

Discrete Time Scatter Plot Scope – Блок для отражения диаграммы созвездий.

Rectangular QAM Demodulator Baseband – демодулятор QAM-M, обладает теми же параметры, что и модулятор.

Unipolar to Bipolar Converter – Преобразователь сигнала из однополярного в биполярный. На вход турбо-декодера необходимо подавать биполярный сигнал.

Turbo Decoder – Турбо-декодер (рисунок 3.152).

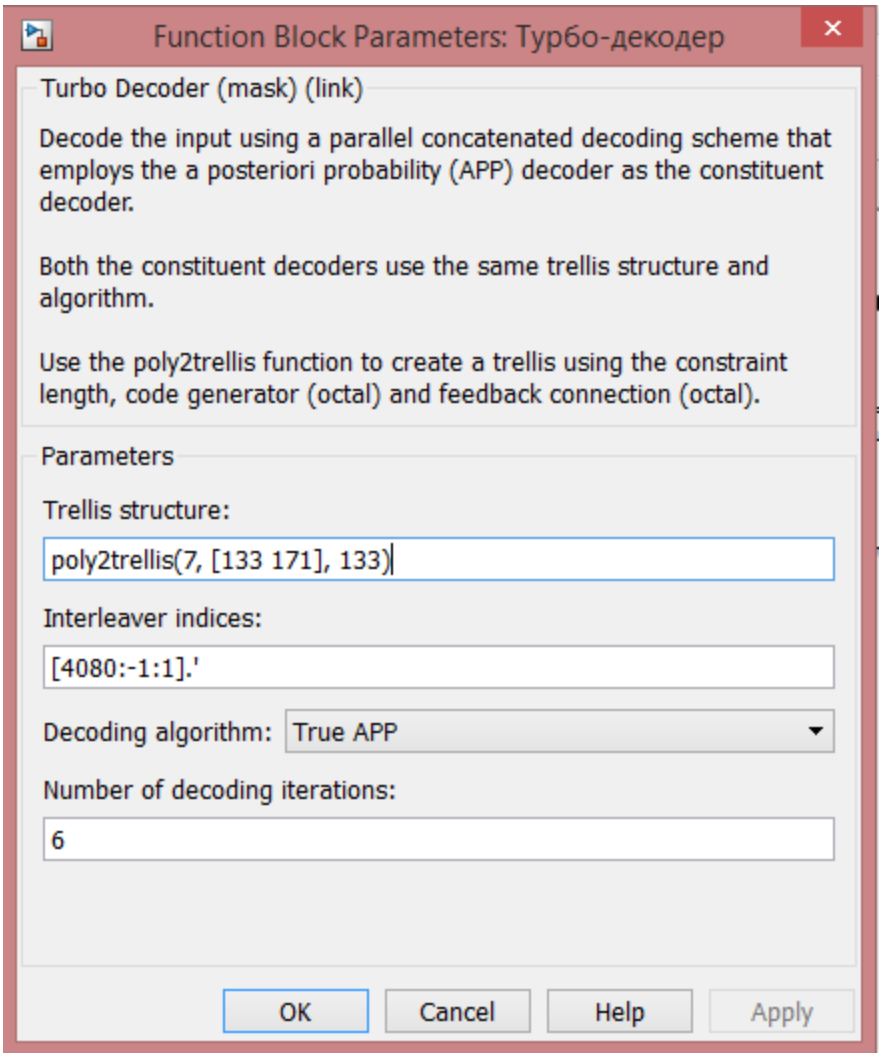


Рис. 3.152. Параметры блока «Turbo decoder»

Первые два параметра задаются аналогично параметрам кодера.

«Number of decoding iterations» – количество итераций декодирования. Декодирование в турбо-декодере происходит в несколько итераций. Чем больше итераций, тем точнее декодирование. Однако, большое количество не даёт результата, а лишь увеличивает длительность вычислений и может даже ухудшить помехоустойчивость.

Схема блока имеет следующую структуру (рисунок 3.153):

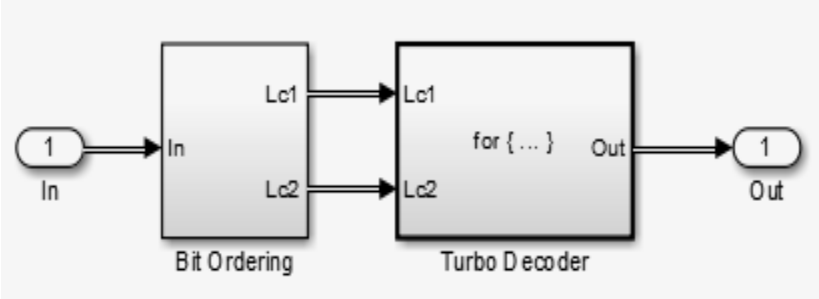


Рисунок 3.153. Структура блока «Turbo-decoder»

«Bit Ordering» - Выстраивание потока бит в параллельный поток.

Схема самого турбо-декодера представлена на рисунке 3.154:

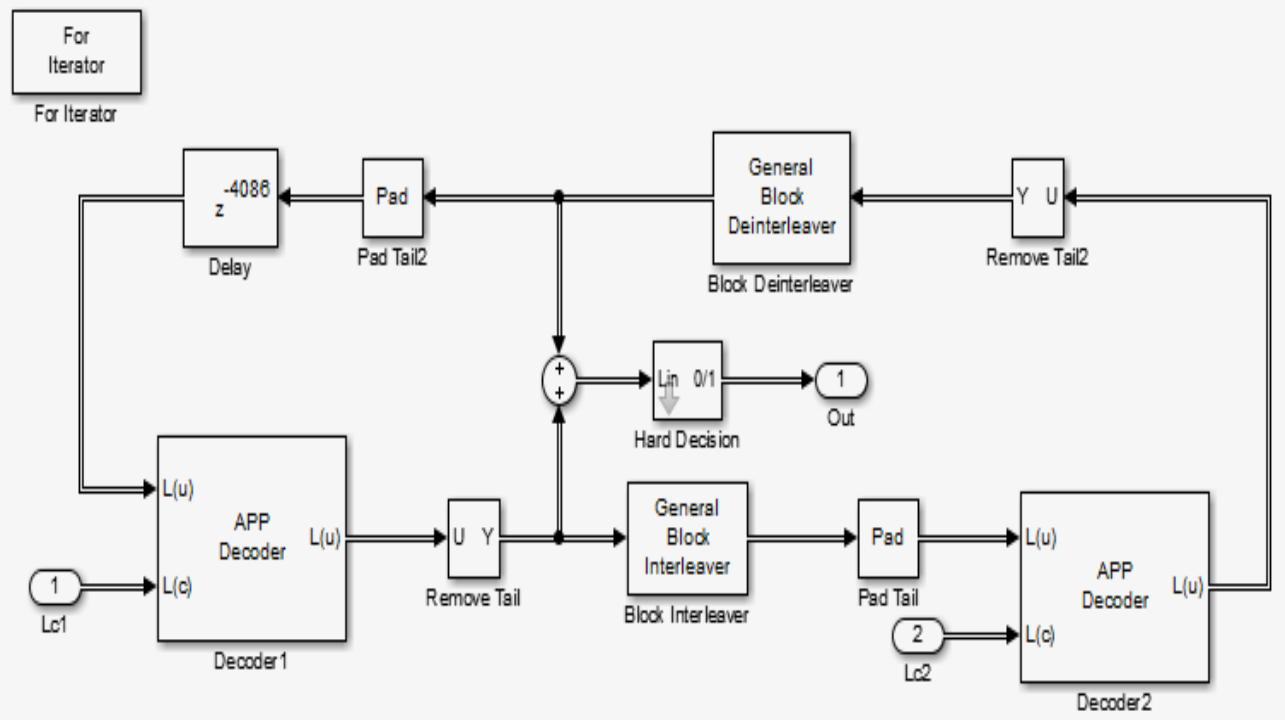


Рисунок 3.154. Схема турбо-декодера

Декодер имеет сложную структуру. Параллельный поток приходит на входы (1) и (2). Данные с входа (2) декодируются и поступают на вход блока на деперемежитель. Затем данные поступают на сумматор и через задержку на декод «Decoder1». Выход этого декодера поступает на сумматор и на перемежитель, данные с которого поступают на второй декодер. Таким образом, декодеры влияют друг на друга и помехоустойчивость и сумма их выходных значений поступает на блок принятия жёстких решений «Hard Decision». Операция декодирования повторяется столько раз, сколько указано в блоке турбо-декодера в параметре «количество итераций».

Binary Input RS Decoder – декодер Рида-Соломона. Параметры и структура аналогична блоку кодера.

Error Rate Calculation – вычислитель ошибок между переданной и принятой последовательностью.

Display - дисплей, отражающий ошибки.

Исследование каскадных кодов

1. Спроектированная модель передачи данных демонстрирует работу каскадного кодирования (рисунок 3.143). Данная модель позволяет исследовать применение последовательно-параллельного кодирования на примере использования кодера Рида-Соломона (внешний код) и Турбо-кодера (внутренний код), а также позволяет исследовать исправляющую способность кодов для разных видов модуляции и сравнить её характеристики с работой указанных выше кодеров в отдельности.

2. В качестве турбо-кода используются два параллельных свёрточных кодера с треллис-модуляцией (для ускорения передачи данных).

На рисунке 3.155 представлен график зависимости битовой вероятности ошибки (BER) от отношения сигнал/шум в канале (SNR) для разных видов модуляции:

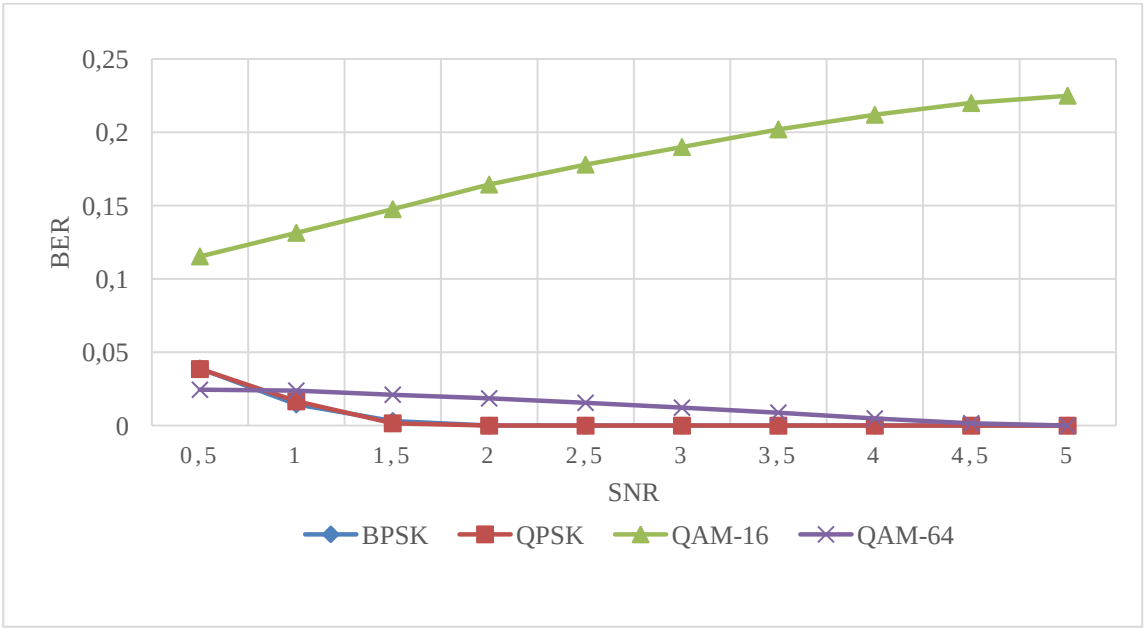


Рис. 3.155. График зависимости BER от SNR для разных видов модуляции

Из рисунка 3.155 можно заметить, что зависимость BER от SNR является неправильной в явном виде. Предположительно, данное явление связано с ошибочным программным кодом самого блока QAM-Modulator, поэтому данную зависимость рассматривать не будем. На

рисунке 3.156 представлен график зависимости BER от SNR в каскадных кодах для видов модуляции BPSK, QPSK и QAM-64:

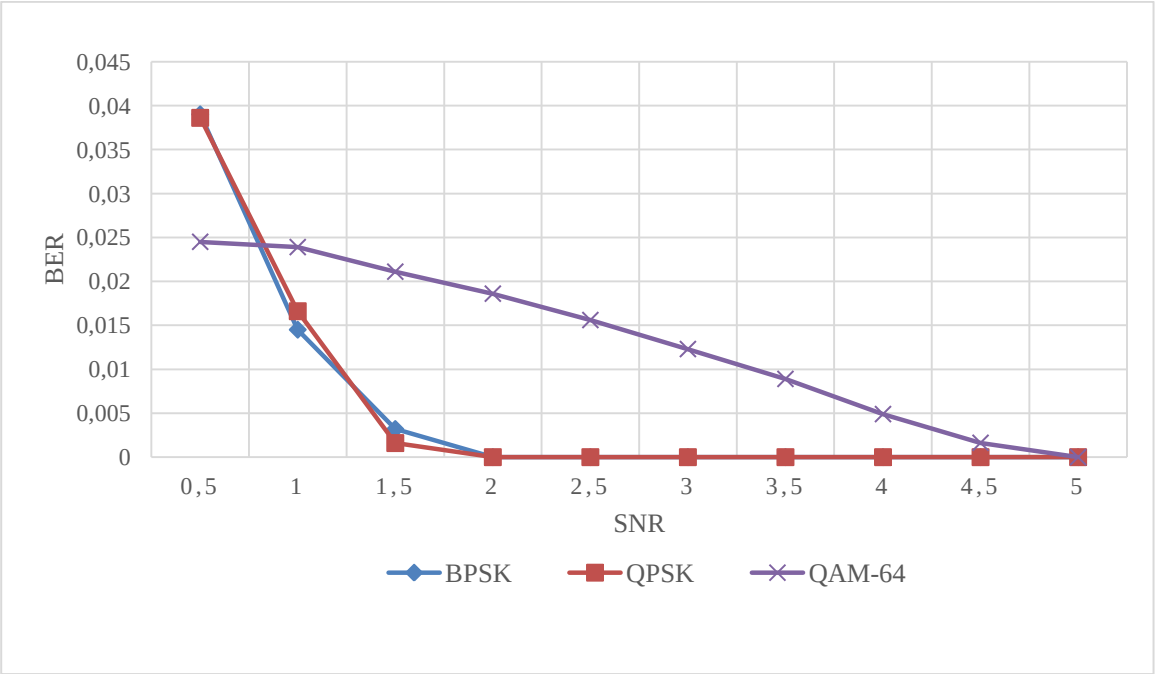


Рис. 3.156 – График зависимости BER от SNR для разных видов модуляции

На рисунке 3.157 представлена диаграмма созвездий QAM-64 сигнала на выходе передатчика:

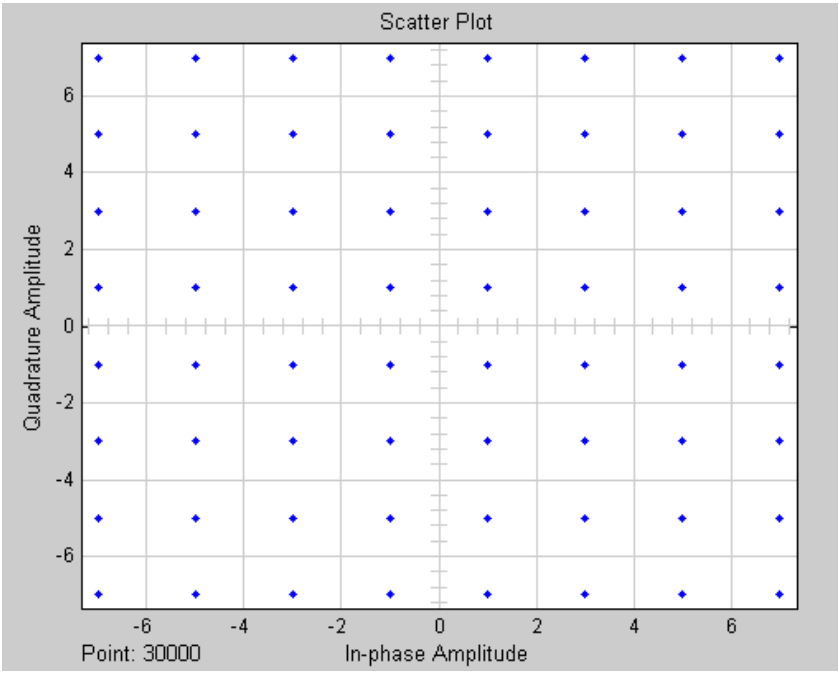


Рис. 3.158. Диаграмма созвездий QAM-64 сигнала на выходе передатчика

На рисунке 3.159 представлена диаграмма созвездий QAM-64 сигнала на приёмном конце после канала с шумом (SNR = 3 дБ) :

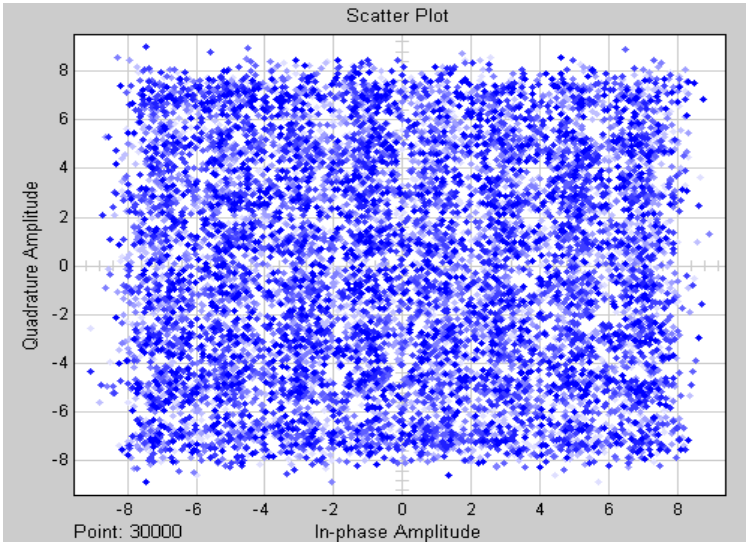


Рис. 3.160. Диаграмма созвездий QAM-64 сигнала на входе приёмника

На рисунке 4.7 представлена диаграмма созвездий QAM-64 сигнала после исправления ошибок каскадным декодером (SNR = 3 дБ) :

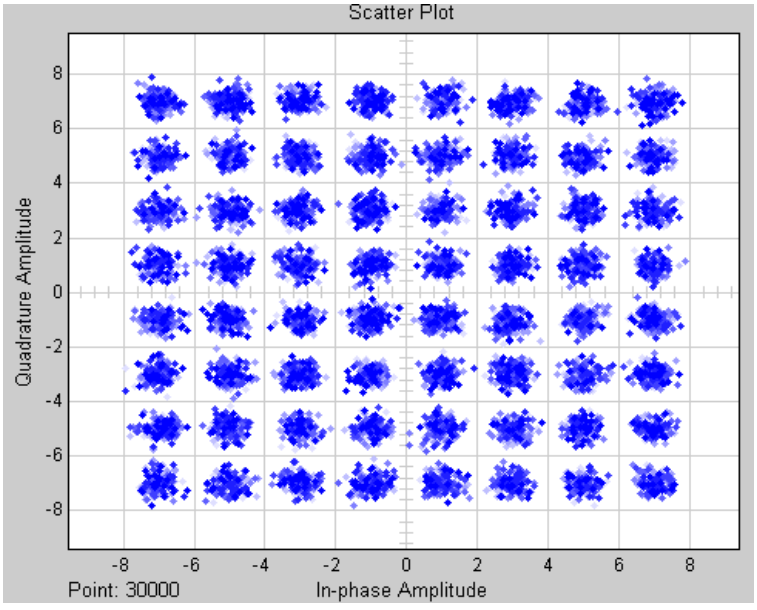


Рис. 3.161. Диаграмма созвездий QAM-64 сигнала после декодирования

На рисунке 3.162 представлены временные формы сигнала QAM-64 с каскадным кодированием (SNR = 3 дБ) :

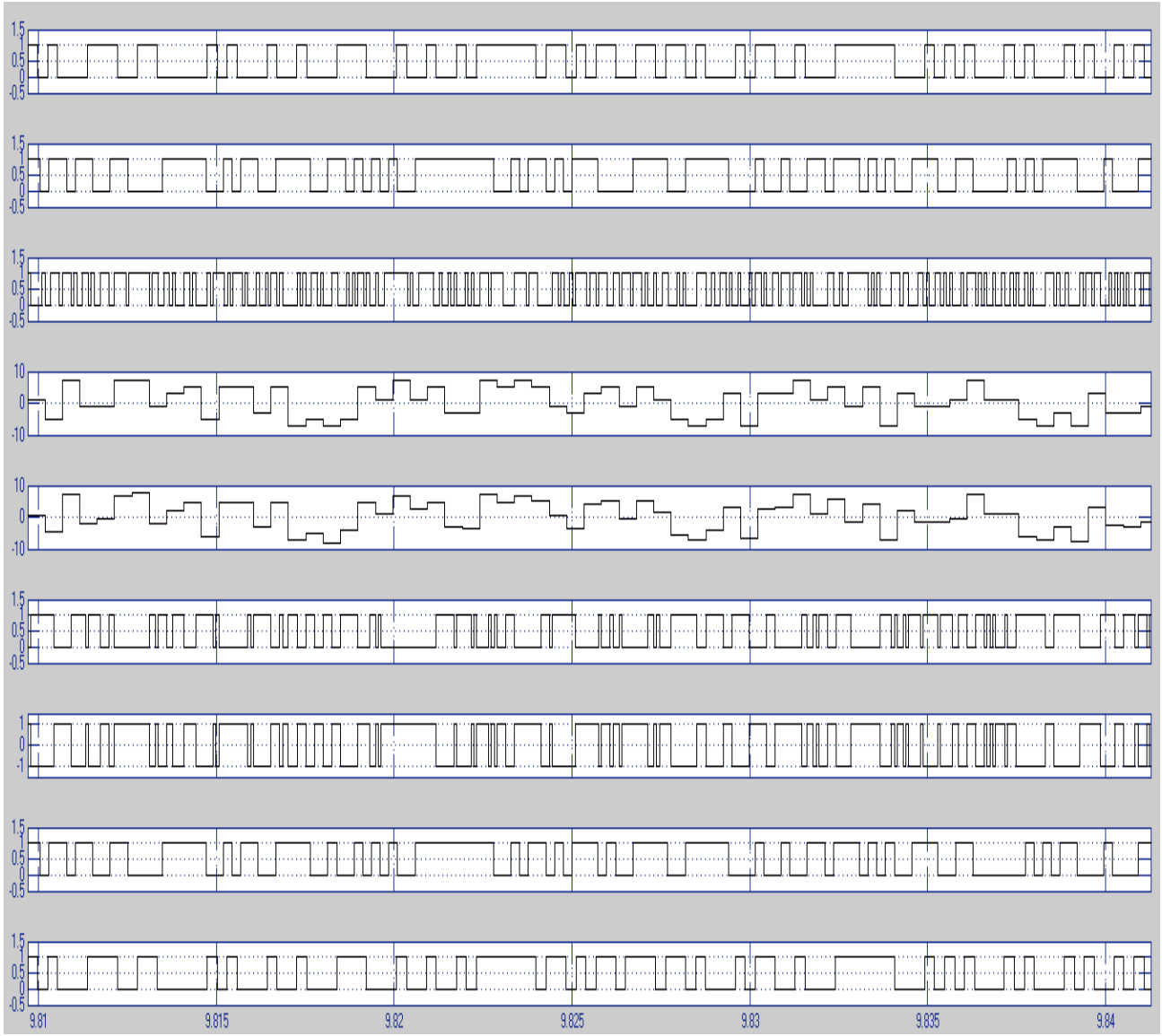


Рис. 3.162. Временные формы сигналов (сверху вниз):

1) на выходе генератора псевдослучайной последовательности;

2) на выходе внешнего кодера (РС);

3) на выходе внутреннего кодера (турбо-кодера);

4) на выходе QAM-модулятора;

5) на входе QAM-демодулятора;

6) на выходе QAM-демодулятора;

7) на выходе преобразователя сигнала из однополярного в биполярный;

8) на выходе внутреннего декодера (турбо-декодера);

9) на выходе внешнего декодера (РС).

На рисунке 3.163 представлен график зависимости BER от SNR при модуляции QAM-64 для каскадного кода, внешнего кода и внутреннего кода:

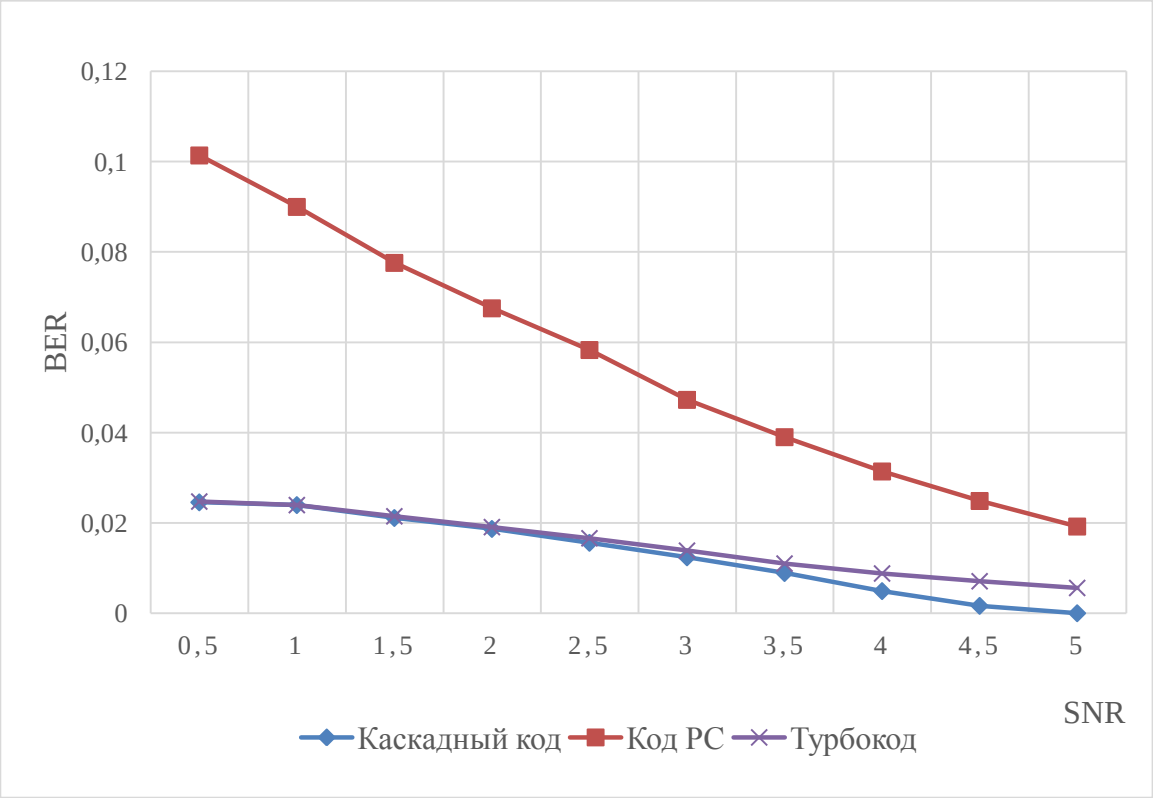


Рис. 3.163. График зависимости BER от SNR для каскадного кода, кода Рида-Соломона и турбокода

Как видно из рисунка 4.9, применение каскадного кодирования неоправданно по сравнению с применением простого турбокодирования, однако он имеет гораздо лучшую характеристику, чем применение простого помехоустойчивого кодирования (РС).

Также данная модель позволяет исследовать исправляющую способность каскадного кода в зависимости от количества итераций декодирования (рисунок 4.10).

Как видно из графика, повышение количества итераций декодирования не улучшает помехоустойчивость, а даже делает её чуть хуже, и, к тому же, приводит к повышению времени декодирования каждой посылки, кратное количеству этих итераций.

Можно сделать предположение, что повышение количества итераций необходимо при увеличение размера фрейма.

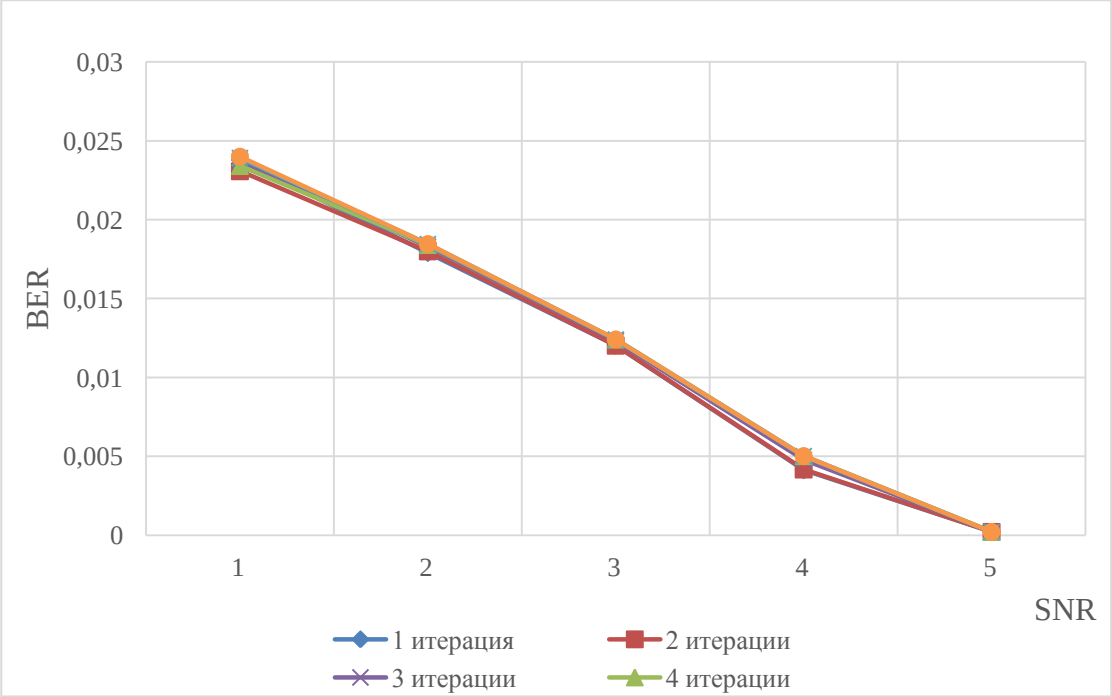


Рис. 3.164. График зависимости BER от SNR для разных значений количества итераций декодирования при количестве переданных символов ~ 75 000.

В результате работы спроектированы модель исследования каскадных кодов.

Модель позволяет исследовать работу каскадных кодов. В качестве внешнего кода используется код Рида-Соломона, в качестве внутреннего – Турбо-код на базе свёрточного кодирования и треллис-модуляции. Данная модель позволяет исследовать зависимость битовой вероятности ошибки (BER) системы от отношения сигнал/шум (SNR) в канале.

Получены следующие результаты и выводы:

Как видно из рисунка 4.9, применение каскадного кодирования неоправданно по сравнению с применением простого турбо-кодирования.

Повышение количества итераций декодирования не улучшает помехоустойчивость при одинаковой характеристики канала, и, к тому же, приводит к повышению времени декодирования каждой посылки, кратное количеству этих итераций.

Повышение количества итераций необходимо при изменении параметров канала.

Эффективность от применения каскадного кодирования заметна лишь при значительном размере фрейма (k>10000).

Методические указания позволяют использовать данные модели на лабораторных работах студентами для исследования помехоустойчивых кодов.

4. СИГНАЛЬНО-КОДОВЫЕ КОНСТРУКЦИИ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

4.1. Сигнально-кодовые конструкции на основе Треллис кодовой модуляции (ТСМ) и их анализ с использованием MATLAB [16]

В цифровых системах связи обычно повышают скорость передачи данных путем уменьшения энергетической емкости бита, т.е. количества энергии сигнала, приходящейся на один бит информации. Но чем меньше энергия, тем больше вероятность того, что бит будет искажен в канале при передаче. Поэтому при повышении скорости передачи разработчики всегда сталкиваются со снижением уровня помехоустойчивости.

Для повышения помехоустойчивости канала передачи данных в цифровых системах применяются коды, исправляющие ошибки. Однако действие таких кодов не всегда эффективно, так как снижается скорость передачи данных.

Треллис-модуляция (ТСМ – Trellis Coded Modulation) представляет собой способ, который позволяет обеспечить повысить скорость передачи сообщения с сохранением уровня помехоустойчивости. Этот способ отличается тем, что помехоустойчивое кодирование и тип модуляции используются совместно. Выбранная соответствующим образом пара помехоустойчивый код – способ модуляции часто также носит название сигнально-кодовая конструкция (СКК).

В данной работе описан способ включения сверточного кодека, используемого для передачи данных с помощью радиорелейных систем связи, в режим треллис-модуляции. Такой способ повышает скорость передачи в два раза при сохранении уровня помехоустойчивости.

В цифровых системах связи обычно повышают скорость передачи данных путем уменьшения энергетической емкости бита, т.е. количества энергии сигнала, приходящейся на один бит информации. Но чем меньше энергия, тем больше вероятность того, что бит будет искажен в канале при передаче. Поэтому при повышении скорости передачи разработчики всегда сталкиваются со снижением уровня помехоустойчивости.

Для повышения помехоустойчивости канала передачи данных в цифровых системах применяются коды, исправляющие ошибки. Однако действие таких кодов не всегда эффективно, так как снижается скорость передачи данных.

Треллис-модуляция (ТСМ – Trellis Coded Modulation) представляет собой способ, который позволяет обеспечить повысить скорость передачи сообщения с сохранением уровня помехоустойчивости. Этот способ отличается тем, что помехоустойчивое кодирование и тип модуляции используются совместно. Выбранная соответствующим образом пара помехоустойчивый код – способ модуляции часто также носит название

сигнально-кодовая конструкция (СКК).

В данной работе описан способ включения сверточного кодека, используемого для передачи данных с помощью радиорелейных систем связи, в режим треллис-модуляции. Такой способ повышает скорость передачи в два раза при сохранении уровня помехоустойчивости.

Корректирующие коды

Наряду с многопозиционными сигналами для повышения эффективности системы электрической связи (СЭС) широко используются помехоустойчивые коды. Применение корректирующих кодов позволяет повысить верность передачи сообщений или при заданной верности повысить энергетическую эффективность системы. Это особенно важно для систем с малой энергетикой, например, систем спутниковой связи.

На практике используются как блочные, так и непрерывные коды. На рис. 4.1 приведены кривые эффективности для циклического кода Боуза-Чоудхури-Хоквингема (БЧХ) и для сверточного кода (СК) с декодированием по по алгоритму Витерби.

Применение циклического кода позволяет получить энергетический выигрыш $\Delta\beta = 2...4$ дБ, а сверточного кода $\Delta\beta = 4...6$ дБ в обмен на снижение частотной эффективности примерно в 2 раза (3 дБ).

Энергетический выигрыш $\Delta\beta$ от применения помехоустойчивого кодирования тем больше, чем выше требуемая верность передачи. Для непрерывного канала с белым гауссовским шумом при требуемой вероятности ошибки 10^{-5} предельный энергетический выигрыш кодирования по сравнению с ФМн-2 без кодирования при оптимальном когерентном приеме составляет примерно 10 дБ. Расчетные кривые на рис. 9.2 показывают, что применение циклического кода в канале с фазовой манипуляцией (ФМн) или сверточного кода в канале с АФМ позволяет повысить одновременно энергетическую, так и частотную эффективности. Построение таких высокоэффективных систем на основе сигнально-кодовых конструкций ведет к неизбежному увеличению сложности системы. Не пропускная способность, а сложность является ограничивающим фактором при построении высокоэффективных систем. Задача состоит в том, чтобы построить систему, удовлетворяющую высоким показателям эффективности, при допустимой сложности.

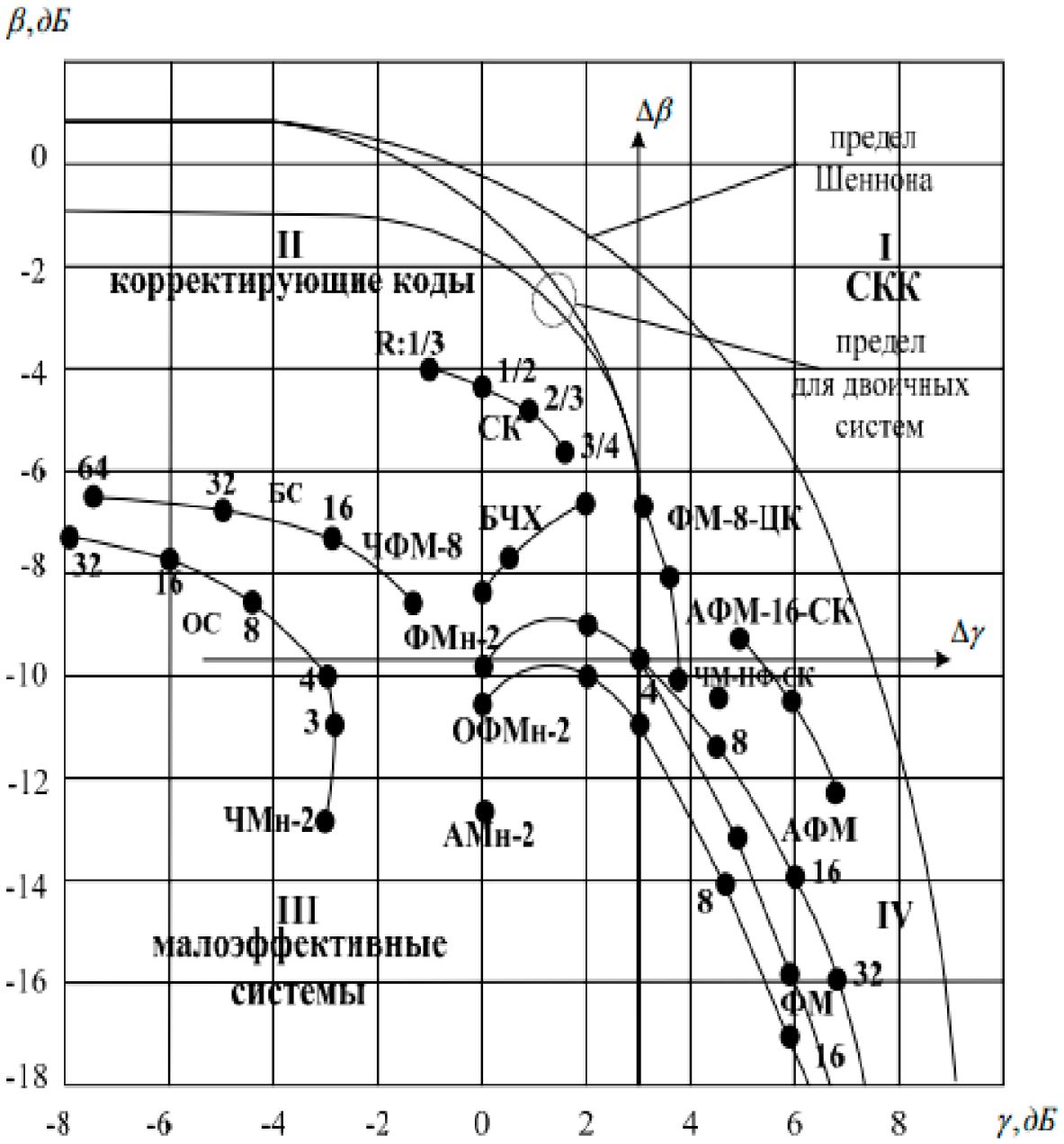


Рис.4.1. Графики зависимости энергетической и частотной эффективности систем связи

При современной элементной базе затраты на реализацию кодирующих и декодирующих устройств значительно сократились. В то же время стоимость энергетики канала практически не изменилась. Таким образом, «цена» выигрыша Δβ за счет кодирования может быть существенно меньше цены того же выигрыша, полученного за счет увеличения энергетики канала (мощности сигнала или размеров антенн).

Отметим, что выбор способов кодирования и модуляции зависит от характеристик

канала. Улучшение этих характеристик, например, путем адаптации к помехам и оценивания искажений сигнала и их последующей компенсации, снижает потери в канале и создает лучшие условия для применения корректирующих кодов.

Оптимизация систем связи

Повышение таких важнейших показателей систем электрической связи, как скорость и верность передачи, связано со значительными частотными и энергетическими затратами. Сравнение между собой различных СЭС осуществляется по степени использования ими основных ресурсов канала связи (пропускной способности, мощности, занимаемой полосы частот), выражаемой через показатели информационной, энергетической и частотной эффективности. Создание СЭС, в которых достигаются близкие к предельным показатели эффективности, требует совместного согласования кодека и модема с учетом статистических свойств непрерывного канала.

Согласование методов модуляции и кодирования

Эффективный путь повышения удельной скорости передачи информации заключается в увеличении числа используемых сигналов m на интервале T . Однако увеличение m приводит к уменьшению расстояния между ближайшими сигналами ансамбля и снижению энергетической эффективности.

При высоких требованиях к верности передачи целесообразным становится применение помехоустойчивых кодов, которые позволяют повысить энергетическую эффективность за счет снижения частотной. Помехоустойчивое кодирование позволяет снизить необходимую величину мощности сигнала, поскольку расстояние между кодовыми комбинациями увеличивается. Одновременное требование большой скорости и верности передачи в условиях ограниченного частотного и энергетического ресурса может быть выполнено при использовании многопозиционных сигналов и помехоустойчивых кодов.

При многопозиционной модуляции, когда по каналам связи передается блок из n кодовых символов, важно также правильно выбрать манипуляционный код, определяющий правило сопоставления с каждым передаваемым сигналом определенного блока кодовых символов. Общий принцип заключается в том, что большему расстоянию по Хэммингу между кодовыми блоками должно соответствовать большее расстояние по Евклиду между отображающими их сигналами.

Создание СЭС, в которых достигаются близкие к предельным показатели эффективности, требует совместного согласования кодека и модема с учетом статистических свойств непрерывного канала. Это означает, что кодирование и модуляцию необходимо рассматривать как единый процесс формирования сигнала, а демодуляцию и декодирование – как процесс оптимального приема сигнально-кодового блока в целом.

Согласование модуляции и кодирования сводится к поиску такого заполнения сигнального пространства, при котором обеспечивается высокая удельная скорость (сигналы расположены достаточно плотно) и одновременно высокая помехоустойчивость (сигналы достаточно далеко друг от друга).

Комбинирование различных ансамблей *m*-ичных сигналов, помехоустойчивых и манипуляционных кодов порождает множество конструкций. Однако только согласованные варианты обеспечивают повышение частотно - энергетической эффективности СЭС.

Эти варианты называют сигнально - кодовыми конструкциями (СКК). Рассмотрим обобщенную схему передачи дискретных сообщений, приведенную на рис.4.2.

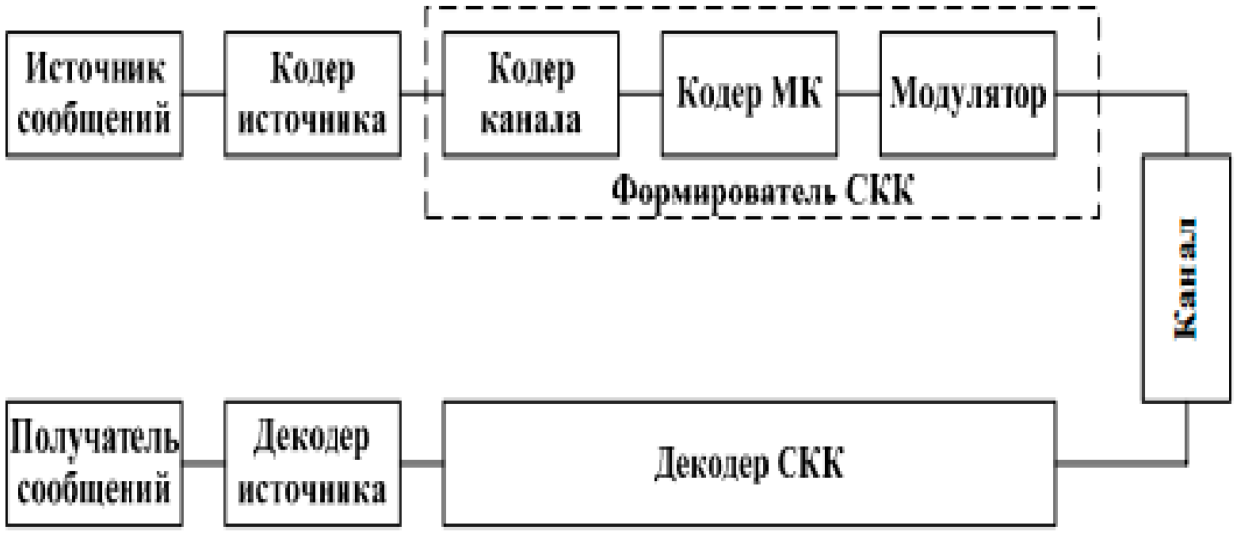


Рис. 4.2. Обобщенная схема передачи дискретных сообщений

Последовательность символов с выхода кодера канала разбивается на блоки по *n* символов. Отображение блоков в сигналы, формируемые модулятором, осуществляется по правилу манипуляционного кодирования, т.е. манипуляционный код определяет правило соответствия блоков кодовых символов *m* - ичным сигналам. Например, в случае двоичного канального кода каждому из $m = 2^n$ кодовых блоков ставится в соответствие один из 2^n сигналов.

Оптимальная процедура приема СКК заключается в обработке сигнально - кодового блока в целом. Поэтому демодулятор, декодеры канала и манипуляционный декодер рассматриваются как единое устройство – декодер СКК (рис.4.2).

Декодер СКК строится так, чтобы минимизировалась вероятность ошибки приема. Оптимальный декодер реализует принцип максимального правдоподобия. При белом

гауссовском шуме выбирается кодовое слово, находящееся на минимальном евклидовом расстоянии от принятого.

Декодирование МК можно рассматривать как последний этап обработки сигнально – кодового блока оптимальным декодером СКК. При этом декодер канала работает в метрике Евклида с сигналами, а не с их двоичными представлениями по правилу манипуляционного кода. Схема поэлементного приема, наоборот, ориентирована на применение декодера канала в метрике Хемминга, т.е. обработку двоичных величин после декодера манипуляционного кода.

Следовательно, достижение наибольшей эффективности возможно при декодировании по алгоритму максимального правдоподобия сигнально - кодового блока в целом». Необходимо отметить, что в принципе любой сигнальный ансамбль на выходе последовательно соединенных кодека и модема может быть отнесен к СКК. Введение понятия СКК отражает подход к модуляции и кодированию как процессу объединения сигналов и кодов в единую эффективную конструкцию.

Классификация сигнально – кодовых конструкций

В основе формирования СКК лежат операции отображения информационной последовательности в кодовую, путем внесения избыточности и кодовой последовательности в канальную заданием манипуляционного кода. Помехоустойчивое кодирование, повышающее энергетическую эффективность СЭС, является одной из важнейших операций формирования СКК. Получаемый при этом энергетический выигрыш от кодирования зависит от степени увеличения минимального сигнального расстояния между разрешенными кодовыми блоками. В качестве сигнального для канала АБГШ используется расстояние Евклида. Асимптотический энергетический выигрыш определяется формулой [21]:

$$ЭВК[ДБ] = 20\lg\left(\frac{d_{ef}}{d_e}\right), \qquad (4.1)$$

где, d_{ef} – минимальное евклидово расстояние между разрешенными кодовыми блоками; d_e – минимальное евклидово расстояние между различными некодированными последовательностями канальных символов одинаковой мощности с кодированными символами.

Согласно (4.1), для получения больших величин энергетического выигрыша при построении СКК необходимо подбирать коды, максимизирующие минимальное евклидово расстояние между разрешенными кодовыми комбинациями.
В основу классификации СКК можно положить отличительные особенности по типам

помехоустойчивого кода, по типам ансамблей сигналов и по способам согласования модуляции и кодирования.

По типу помехоустойчивых кодов все СКК могут быть поделены на два больших класса: СКК на основе блочных кодов и СКК на основе непрерывных кодов. Кроме того, отдельный класс составляют СКК на основе каскадных кодов, в которых применяются одновременно блочные и непрерывные коды. Каждый из классов делится на группы по конкретным видам кода.

Среди блочных наиболее употребимыми являются коды Хэмминга, Голея, БЧХ, Рида - Соломона, Рида–Маллера и др. Непрерывные коды на практике представлены сверточными кодами, которые обладают дополнительными свойствами линейности, и постоянства во времени.

При использовании сверточного кода практически удобным является случай, когда при объеме ансамбля сигналов $m = 2\ k+1$ скорость сверточного кода выбирается равной $Rкк = k/k+1$. Тогда частотная эффективность у системы с кодированием и без него одна и та же. Поскольку каждый кодовый блок длиной $(k + 1)$ переносится одним двумерным сигналом, то и СКК считается также двумерной.

Декодирование СКК ведется обычно по алгоритму Витерби, реализующему принцип максимального правдоподобия. Одно из важнейших преимуществ СК заключается в простоте применения алгоритма Витерби для мягкого решения на выходе демодулятора.

Любая СКК вне зависимости от способа согласования модуляции и кодирования представляет собой каскадный код с ансамблем сигналов на внутренней ступени и одним или несколькими помехоустойчивыми кодами на внешней. При использовании нескольких помехоустойчивых кодов говорят о построении СКК на основе обобщенного каскадного кода.

По типу ансамблей сигналов СКК делятся на конструкции с одномерными, двумерными и многомерными сигналами. Многомерные сигналы составляются из более простых (одномерных, двумерных) сигналов. При использовании в качестве составляющих двумерных сигналов число позиций M, соответствующих каждому n –мерному сигналу, определяется выражением $M = m^{n/2}$, где m – позиционность двумерного сигнала.

Каждый n –мерный сигнал в этом случае образуется последовательностью $n/2$ двумерных сигналов. Например, для получения многомерного сигнала с $n = 6$ требуется последовательность из трех двумерных сигналов, например ФМн-4.

Способы согласования модуляции и кодирования условно можно разделить на две группы: согласование кодом Грея и согласование на основе разбиения ансамбля на вложенные подансамбли.

Сигнально-кодовые конструкции, принадлежащие первой группе, представляют собой результат согласования известных двоичных помехоустойчивых кодов с многопозиционным ансамблем сигналов путем использования кода Грея в качестве манипуляционного кода. Поскольку ошибки происходят за счет переходов в области соседних сигналов, то кодовые блоки, соответствующие соседним сигналам, должны различаться наименьшим числом двоичных символов.

Вторая группа включает в себя достаточно большое число типов СКК, различающихся модификациями методов согласования. Разбиение осуществляется таким образом, что подансамбли содержат равное количество сигналов, расстояния между соседними сигналами подансамблей одинаковы, а минимальные расстояния между сигналами подансамблей увеличиваются с каждым шагом разбиения. Широкое практическое применение получило согласование путем разбиения ансамбля на вложенные подансамбли, когда внешними кодами являются сверточные коды. В основе синтеза СКК со сверточными кодами лежит поиск кодов, максимизирующих евклидово расстояние, причем обычно эти коды не являются оптимальными в метрике Хэмминга. У решетчатой диаграммы, описывающей сверточные коды в метрике Евклида, переходы между состояниями промаркированы не двоичными блоками, а сигнальными точками.

Таким образом, достижение близких к предельным показателей частотно - энергетической эффективности цифровых систем связи предполагает согласование кодека и модема с учетом статистических свойств непрерывного канала. Одно из решений подобного согласования представляют сигнально-кодовые конструкции сверточного кодирования. Мягкое декодирование по алгоритму Витерби обеспечивает энергетический выигрыш порядка 3...7 дБ без расширения занимаемом полосы частот.

Характеристики основных типов СКК. Согласование канала кодом Грея

Рассмотрим СКК, представляющие собой результат согласования известных двоичных помехоустойчивых кодов с многопозиционным ансамблем сигналов путем использования в качестве манипуляционного кода Грея (табл. 4.1).

Комбинации кода в табл. 4.1 получены по следующему правилу. Кодовая комбинация натурального кода складывается по модулю 2 с такой же комбинацией, сдвинутой на один разряд вправо, при этом младший разряд сдвинутой комбинации отбрасывается.

Таблица 4.1.

Результат согласования двоичных помехоустойчивых кодов с кодом Грея

Десятичное число	Натуральный двоичный код	Код Грея
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Поскольку ошибки чаще происходят за счет переходов в области соседних сигналов, то кодовые блоки, соответствующие соседним сигналам, должны различаться наименьшим числом двоичных символов.

На рис. 4.3 приведены примеры кода Грея для ансамблей одномерных (АМ-4) и двумерных (ФМ-4, КАМ-16) сигналов.

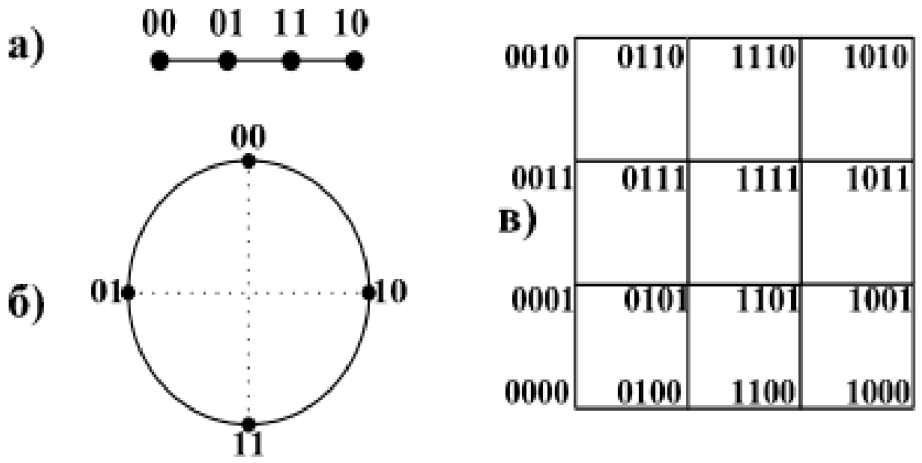


Рис. 4.3. Коды Грея ансамблей: одномерных: а) АМ-4; б) ФМ-4; двумерных: КАМ-16

Несмотря на достаточно высокие показатели энергетической эффективности при мягком решении в демодуляторе и декодировании алгоритмом Витерби, согласование кодом Грея не является оптимальным.

Двоичные коды, оптимальные по критерию максимума хэммингова расстояния, будут оптимальны и по критерию максимума свободного евклидова расстояния, если при отображении двоичных подблоков в сигнальные точки ансамбля выполняется принцип:

большему расстоянию Хэмминга *dh* max , соответствует большее расстояние по Евклиду *de* max .

Простейшие ансамбли сигналов АМн-2, ФМн-2, ФМн-4 этому условию для кода Грея удовлетворяют. В табл. 4.4 показаны комбинации (подблоки) двоичного кода длиной $n = 3$, а также расстояния *dh* и *de* при использовании кода Грея для ФМн-8 (см. рис. 4.4).

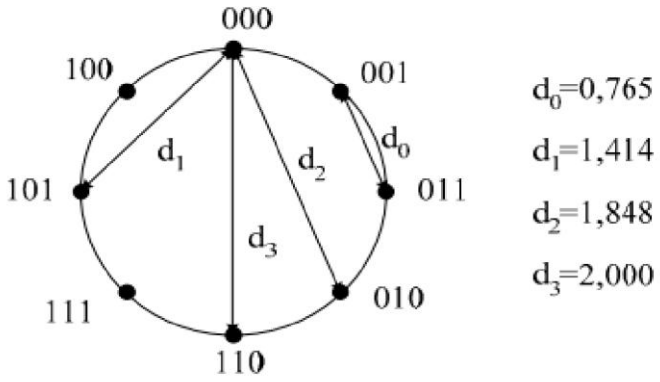


Рис. 4.4. Расстояние между сигнальными точками ФМн-8

Как следует из таблицы, сформулированный принцип соответствия большему расстоянию Хэмминга большего расстояния Евклида для всех вариантов отображения не выполняется. Например, для комбинации 111 большему расстоянию Хэмминга *dh* = 3 соответствует не самое большое расстояние Евклида *dh* = 1,848 , и т.д.

Соответствие расстояний Хэмминга и Евклида для сигналов ФМн-8

Кодовые комбинации	000	001	011	010	110	111	101	100
d_h	0	1	2	1	2	3	2	1
d_e	0	0,765	1,414	1,848	2,000	1,848	1,414	0,765

Таким образом поскольку манипуляционный код Грея для сложных сигналов не обеспечивает оптимального согласования кодека и модема, необходимо найти методы дальнейшего повышения свободного евклидова расстояния *def* и, соответственно, энергетической эффективности β.

Согласование на основе разбиения ансамбля на вложенные подансамбли

В начале 80 х гг. Унгербоек (Ungerboeck G.) опубликовал статью, в которой, анализируя СКК на базе ансамбля ФМн-8 и сверточного кода со скоростью *Rкк* = *k* / *k* +1 ,

сформулировал ряд правил построения СКК. Поэтому СКК построенные по этим правилам (Trellis-Coded Modulation – TCM), часто называют СКК Унгербоека.

По способу согласования модуляции и кодирования СКК Унгербоека относятся к конструкциям, полученным на основе разбиения ансамбля сигналов на вложенные подансамбли. Разбиение осуществляется таким образом, что подансамбли содержат равное количество сигналов, расстояния *de* между соседними сигналами подансамблей одинаковы, минимальные расстояния *de* min между сигналами подансамблей увеличиваются с каждым шагом разбиения; при этом левая ветвь разбиения кодируется символом «0», а правая «1». Считывание кодовой комбинации, соответствующей сигнальной точке на амплитудно-фазовой плоскости, осуществляется снизу вверх. Разбиение для ансамбля сигналов ФМн-8 представлено на рис. 4.5.

Как следует из рис. 4.5, исходный ансамбль разбивается на подансамбли при максимальном увеличении наименьших расстояний *de* min между сигналами внутри подансамблей $d0 < d1 < d2 < d3$. Разбиение осуществляется поэтапно.

В данном примере три этапа, за ключающихся в разбиении каждого из подансамблей пределы предыдущего этапа на 2 равноэлементных подансамбля.

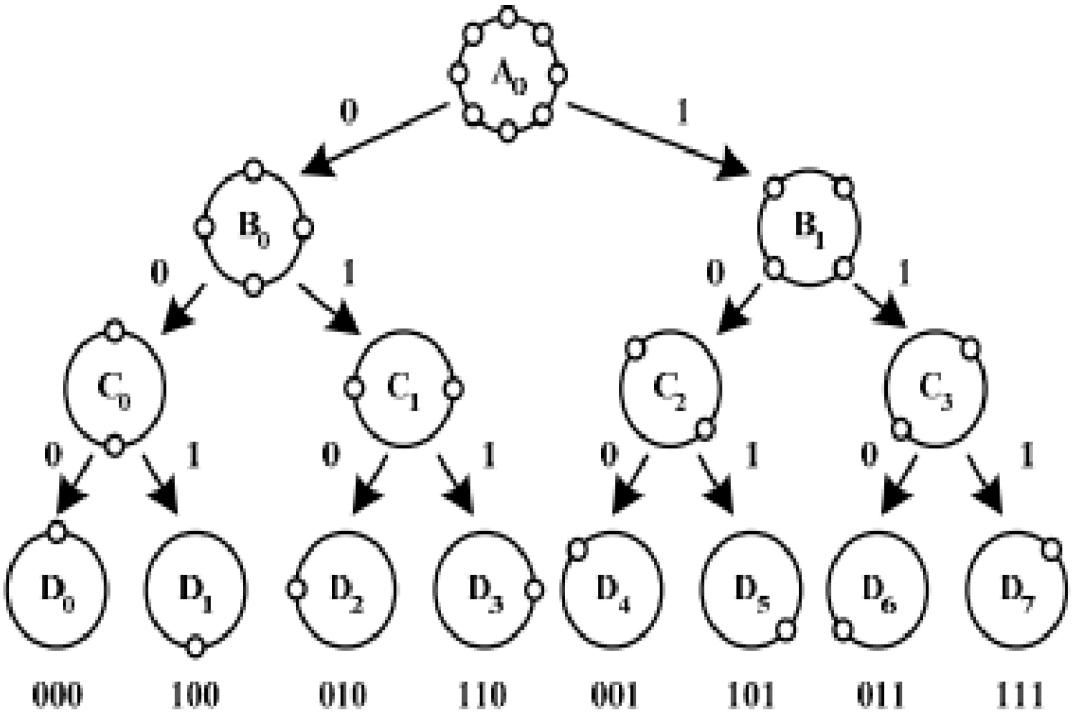


Рис.4.5. Разбиение ансамбля сигналов ФМн-8

В общем случае количество этапов *i* полного разбиения ансамбля из *m* сигналов на вложенные подансамбли определяется выражением: *i* max = log 2 *m*, (4.6), т. е. совпадает кратностью ансамбля *n* . В ансамбле из *m* сигналов кратности *n* каждой сигнальной точке соответствует блок двоичных символов *b* = [*b n*−1 , *b n*−2 ,..., *b* 0]. Соответствие между

кодовым блоком b и сигнальной точкой на плоскости определяет манипуляционный код.

Достижение наибольшей помехоустойчивости непосредственно связано с увеличением евклидова расстояния между передаваемыми сигнальными последовательностями. Решетчатая диаграмма сверточного кода (5.6.3), ребра которой промаркированы сигнальными точками, полностью отображает весь набор разрешенных сигнальных последовательностей.

Таким образом, величина свободного евклидова расстояния def зависит от маркировки ребер решетчатой диаграммы сигнальными точками (канальными символами).

Унгербоек на примере ансамбля сигналов ФМн-8 (см. рис. 4.6) сформулировал четыре необходимых правила маркировки ребер сигнальными точками: все сигнальные точки используемого ансамбля сигналов должны встречаться с одинаковой частотой и с определенной степенью регулярности и симметричности;

переходы из одного и того же состояния соответствуют сигналам из подансамблей $B0$ или $B1$;

переходы в одно и то же состояние соответствуют сигналам из подансамблей $B0$ или $B1$;

параллельные переходы между состояниями соответствуют сигналам из подансамблей $C0$ или $C1$, или $C2$, или $C3$. Как показывает анализ, СКК Унгербоека имеют несколько более высокие частотно энергетические характеристики по сравнению с традиционными СКК, при той же сложности реализации. Это определило их бурное внедрение в технике связи. Но известные правила построения СКК Унгербоека, хотя и снижают размерность переборной задачи синтеза, но не обеспечивают гарантированное построение СКК с максимальными частотно-энергетическими характеристиками. В то же время, основной целью работ в области синтеза систем сигналов и СКК является поиск таких способов их формирования и обработки, которые при заданных ограничениях на сложность устройств формирования и приема, временные задержки, позволяли бы приблизиться к известной шенноновской границе.

При построении многомерных СКК возникает проблема выбора манипуляционного кода, поскольку известные методы его построения (правила построения кодов Грея и разбиения ансамбля на вложенные подансамбли Унгербоека) не всегда позволяют согласовать евклидовы и хеммингговы расстояния. Именно с этим связаны многие проблемы построения многомерных СКК.

Синтез многопозиционных ансамблей сигналов и СКК, построенных на их основе, является одним из направлений решения более общей задачи статистического согласования

вероятностных характеристик передаваемого информационного сигнала и вероятностных характеристик канала. В рамках этих традиционных задач, такое согласование осуществляется на уровне канальных символов или их блоков (супербукв канала). При этом подходы к построению алфавита таких супербукв (ансамблей сигналов и СКК) могут существенно отличаться между собой, но направлены на решение этой общей проблемы.

Известно, что ансамбль сигналов, соответствующий полному двоичному коду длины n в пространстве соответствующей размерности n , построенный заменой «1» на «–1», а «0» на «+1», соответственно, обладает практически идеальным манипуляционным кодом. Минимальным хеммингловым расстояниям таких ансамблей соответствуют ребра n -мерного куба, которые характеризуются и минимальными евклидовыми расстояниями.

Кодовые комбинации и соответствующие им координаты сигнальных векторов приведены в табл. 4.2; графическое изображение ансамбля представлено на рис. 9.13. При приеме сигналов такого ансамбля минимальная ошибка (ошибочный прием одной координаты сигнальной точки) приводит к неправильному приему одного бита информации. Ошибочный прием двух координат сигнальной точки приводит к искажению двух бит информации и так далее. Однако, если рассмотреть зависимость между хеммингловыми $dh(i, j)$ и евклидовыми $de(\ )i, j$ расстояниями для такого ансамбля, то можно выявить следующую закономерность, связывающую эти две величины :

$$de(i, j) = 2r\sqrt{dh(i, j)}, \qquad (4.2)$$

где r – радиус сферы.

Таблица 4.2 - Взаимосвязь кодовых комбинаций манипуляционного кода и координат сигнальных векторов

Манипуляционный код	Координаты сигнальных векторов
000	+1,+1,+1
001	+1,+1,–1
010	+1,–1+1
011	+1,–1,–1
100	–1,+1,+1
101	–1,+1,–1
110	–1,–1,+1
111	–1,–1,–1

Таким образом, взаимосвязь между евклидовыми и хеммингловыми расстояниями в многомерном ансамбле сигналов нелинейная, хотя большему хемминговому расстоянию будет соответствовать большее евклидово расстояние. Если мощность и энергия сигналов

являются постоянными величинами, не зависящими от номера, то ансамбли таких сигналов считают сигналами поверхностно-сферической упаковки.

В противном случае ансамбли сигналов рассматривают как объемные упаковки. Сохранение манипуляционного кода, принятого для простого трехмерного куба, в значительной мере сохраняет пропорциональность между евклидовыми и хемминговыми расстояниями и поэтому будет наилучшим и для наиболее плотного ансамбля. Для других комбинаций манипуляционных кодов для сигнальных векторов изначально не будет соблюдаться взаимная пропорциональность между евклидовыми и хемминговыми расстояниями.

Таким образом, практически невозможно создать идеальный манипуляционный код и, следовательно, целесообразно строить манипуляционные коды, у которых хотя бы частично выполняется взаимосвязь между евклидовыми и хемминговыми расстояниями.

Практическая часть работы в MATLAB 2015 [21]

Перед началом выполнения работы, необходимо с папки ТСМ скопировать код в командную строку, который подгрузит необходимую схему для треллис модуляции.

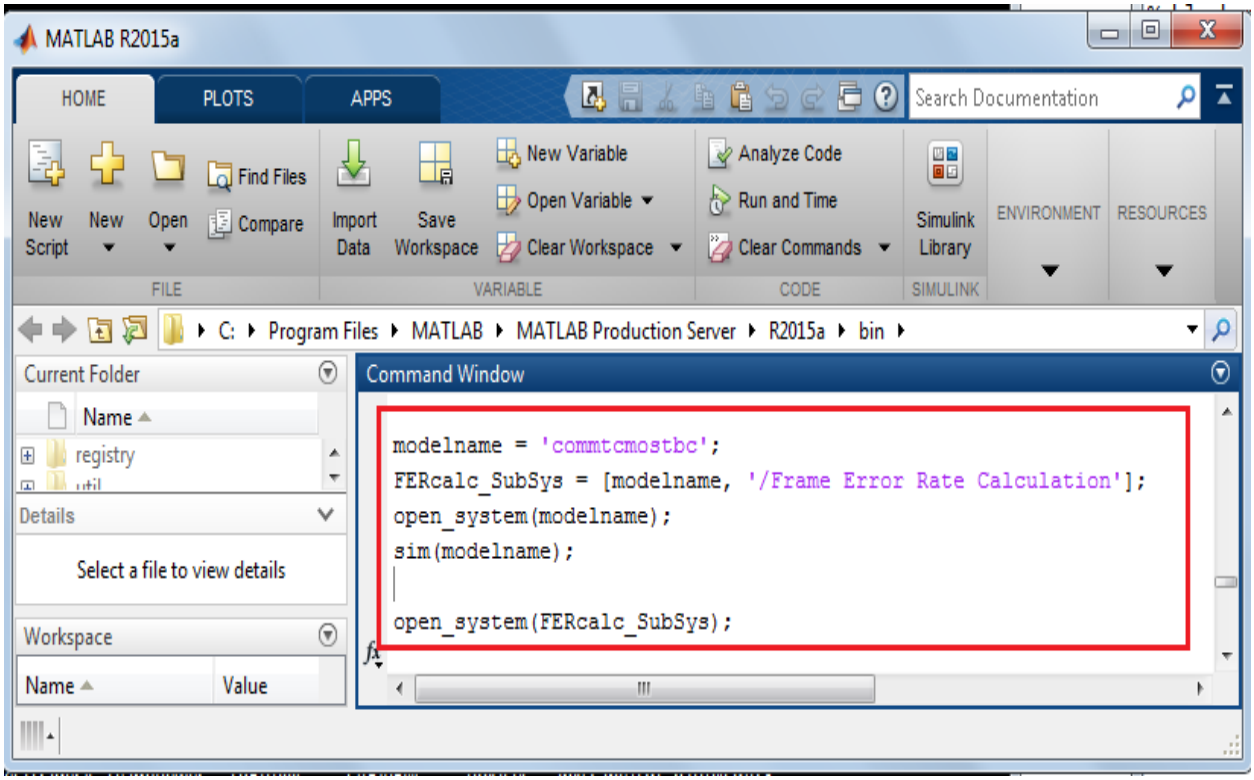


Рис. 4.6. Панель MATLAB 2015

После, откроется следующая схема:

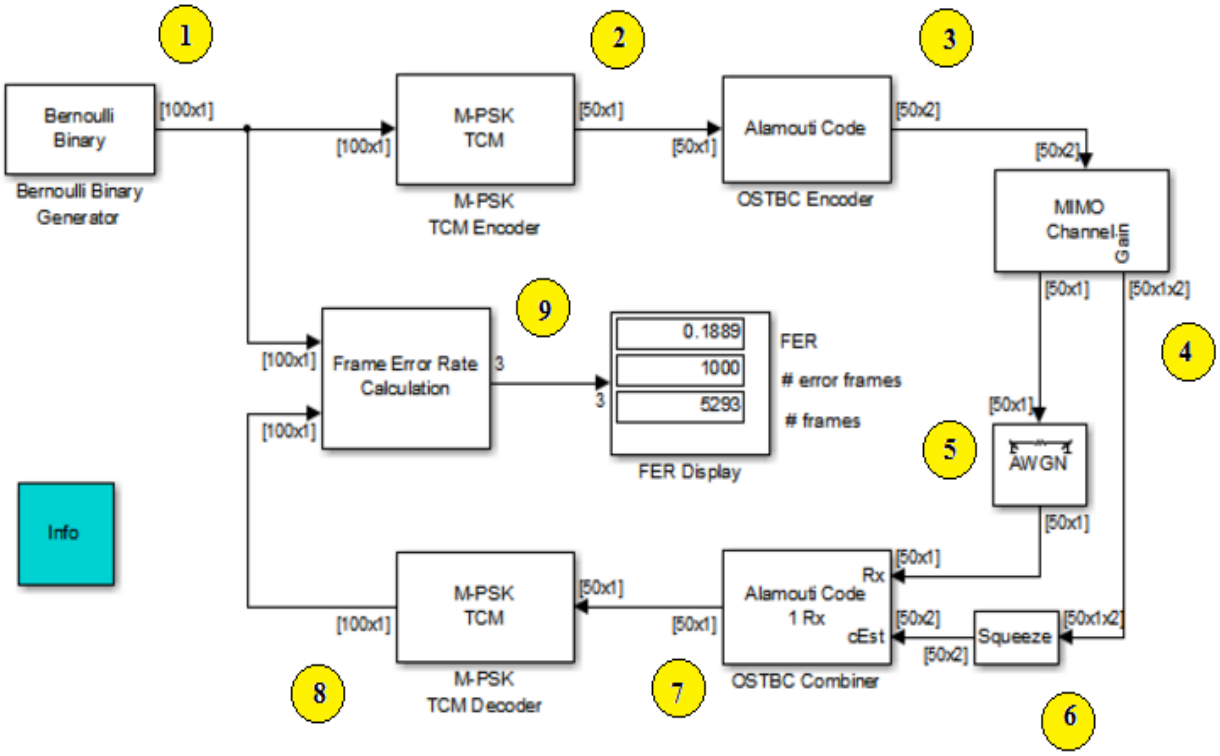


Рис. 4.7. Схема для исследования TCM в Simulink MATLAB 2015

1. Блок Двоичный генератор Бернулли генерирует случайные двоичные числа, используя распределение Бернулли. Выставьте ниже представленные параметры:

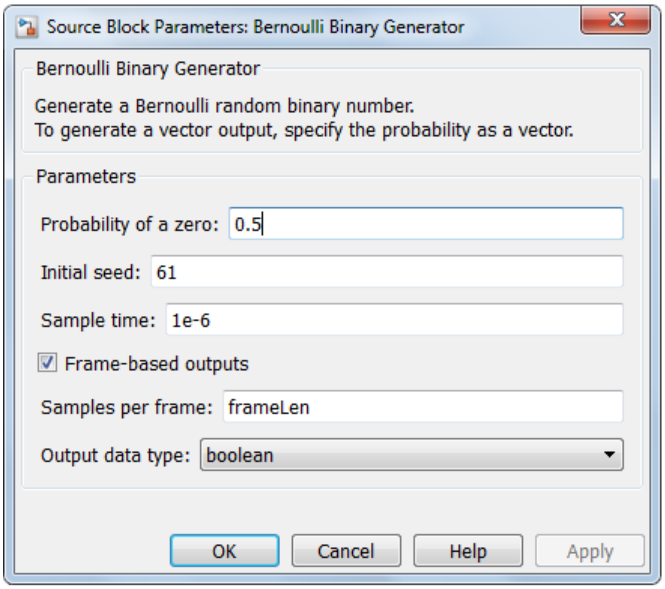


Рис. 4.8. Параметры источника (Source Block Parameters)

2. Блок M-PSK TCM кодер преобразовывает данные, полученные от блока Бернулли в PSK созвездия с заданной его средней энергией. В этом примере мы используем схему TCM 8 - PSK созвездия для 8 решетчатых состояний. Выставьте ниже представленные параметры:

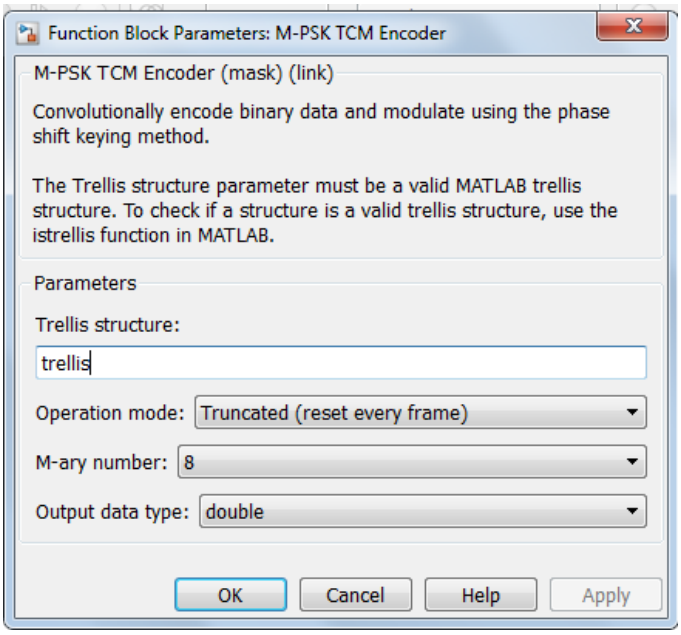


Рис. 4.9. Параметры блока: M-PSK TCM Encoder

После 2го блока нужно добавить диаграмму созвездий, который расположен в панели программы Matlab. Выставьте ниже представленные параметры:

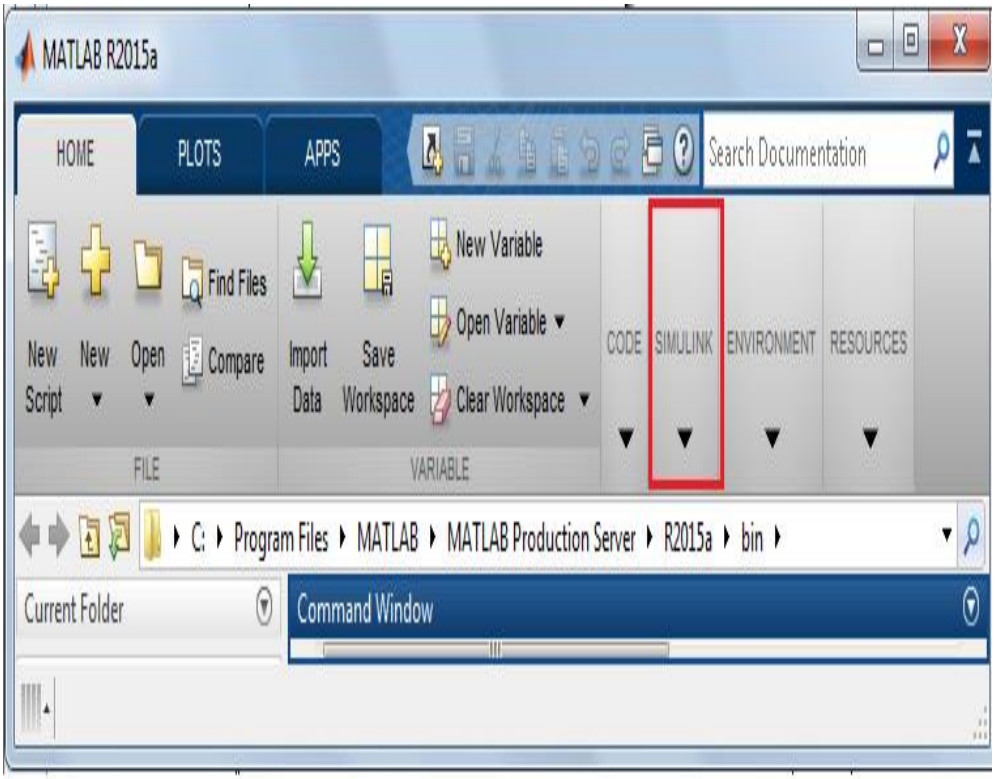


Рис. 4.10. Панель MATLAB

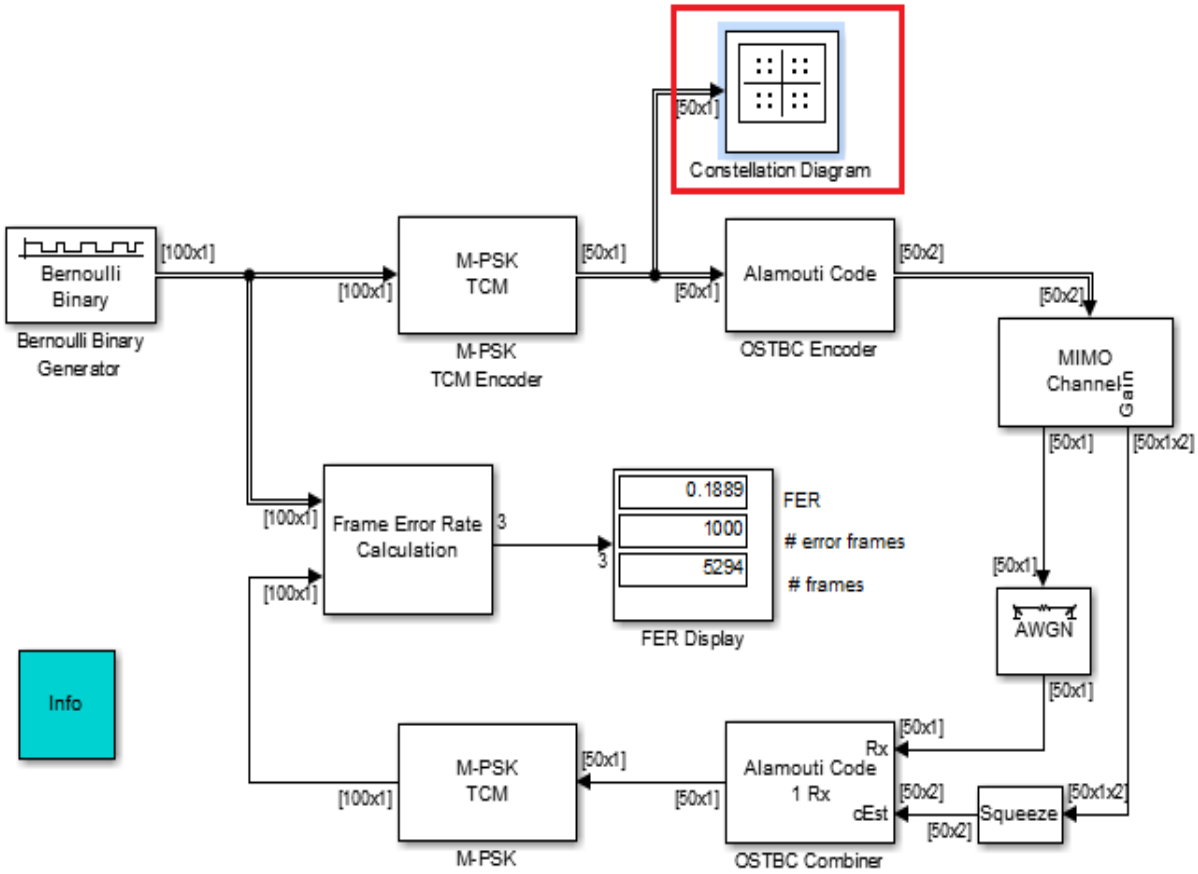


Рис. 4.11. Схема для исследования TCM в Simulink MATLAB 2015

3. Блок OSTBC (ортогонально пространственно-временные блочные коды) кодирует информационные символы из TCM кодировщика, используя код Alamouti для 2x передающих антенн. Выходом этого блока является матрица размером 50x2, элементы которой соответствуют данным колонки, передаваемых по одной антенне.

Блок OSTBC сочетает в себе полученные сигналы от приемной антенны с информацией о состоянии канала (CSI), которые затем подают в M-PSK TCM декодера. CSI известно на стороне приемника. Выставьте ниже представленные параметры:

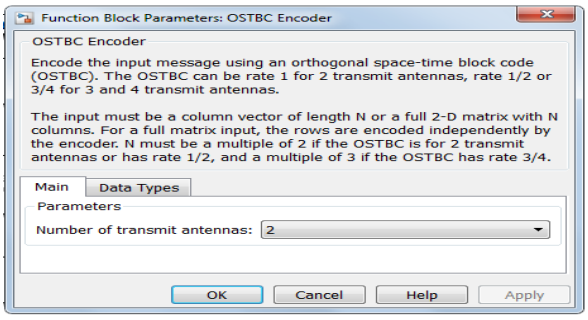


Рис. 4.12. Параметры блока: OSTBC Encoder

4. MIMO канал разносит передающие и приёмные антенны так, чтобы корреляция между соседними антеннами была слабой. Выставьте ниже представленные параметры:

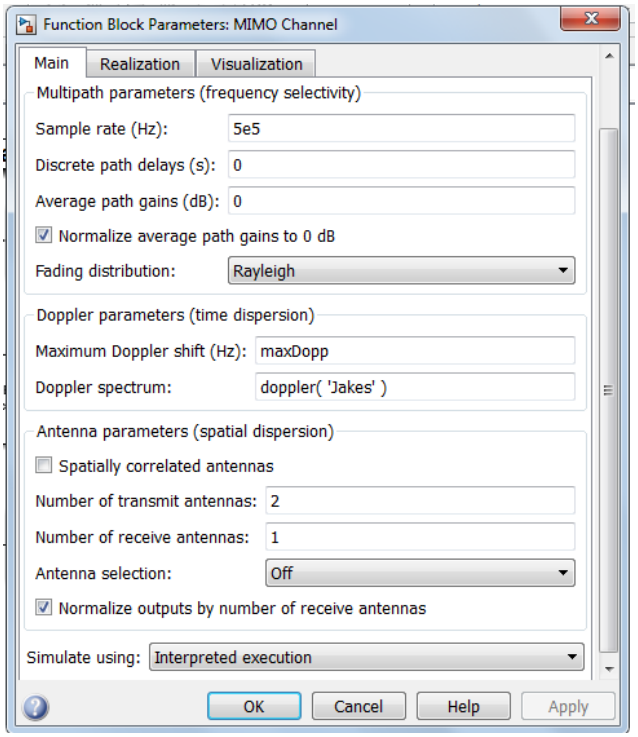


Рис. 4.13. Параметры блока: MIMO Channel

5. Блок AWGN добавляет белый гауссовский шум на приемной стороне. Выставьте ниже представленные параметры:

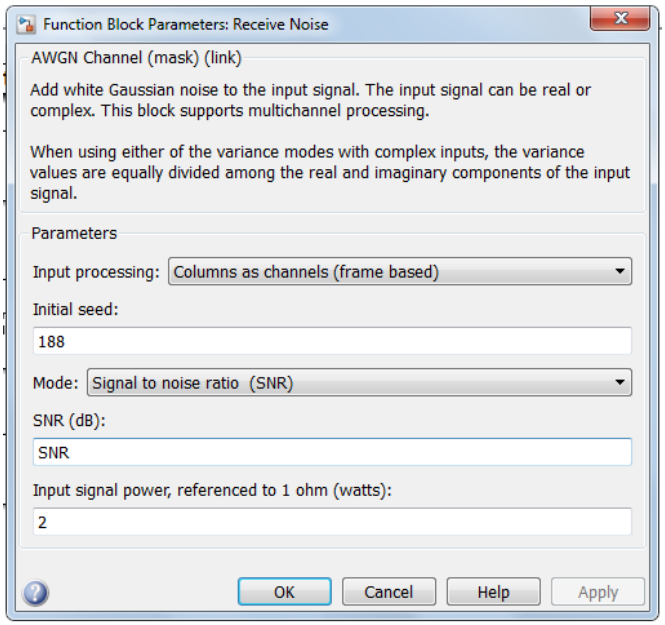


Рис. 4.14. Параметры блока: Receive Noise

6. Блок Сжатия. Выставьте ниже представленные параметры:

7. Блок OSTBC дает оценку канала сигнала. Выставьте ниже представленные параметры:

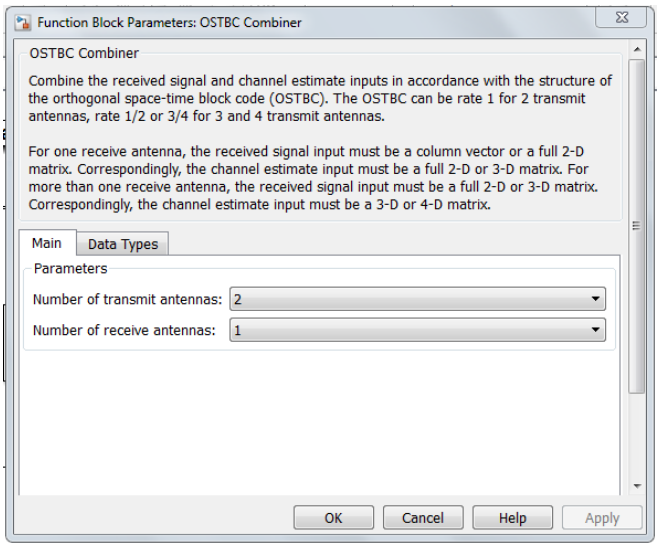


Рис. 4.15. Параметры блока: OSTBC Combiner

8. Блок M-PSK TCM декодер выполняет декодирование сигнала ранее модулированного с использованием PSK созвездия на входе. Для декодирования используется алгоритм Витерби. Выставьте ниже представленные параметры:

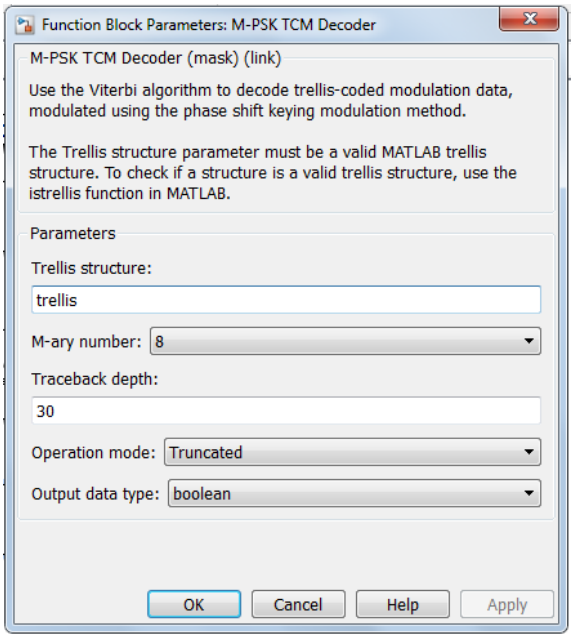


Рис. 4.16. Параметры блока: M-PSK TCM Decoder

9. Блок FER Display сравнивает декодированные биты с исходными битами в кадре для обнаружения ошибок и в онлайн режиме обновляет показатели. Этот блок состоит из трех показателей: количество ошибок в кадре, количество наблюдаемых кадров и количество обработанных ошибок в кадре. Выставьте ниже представленные параметры:

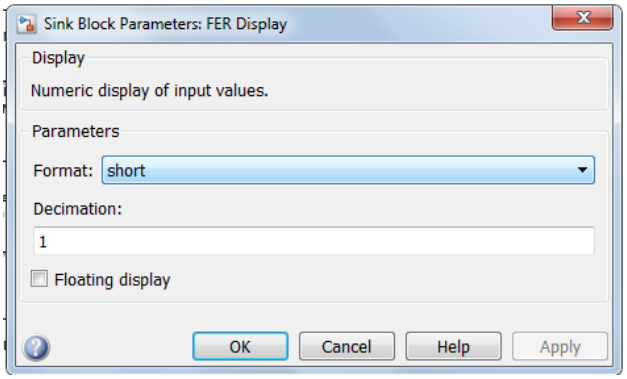


Рис. 4.17. Параметры блока: FER Display

Таблица 4.3 – Результаты работы

FER	1	0,9 94	0,8 873	0,6 289	0,2 813	0,1 88	0,0 8	0,038
SNR	0	2	4	6	8	10	12	14

Таблица 4.4 – Результаты работы

FER	1	0,9 95	0,8 85	0,6 1	0,2 7	0,1 89	0,0 8	0,0 45
SNR	0	2	4	6	8	10	12	14

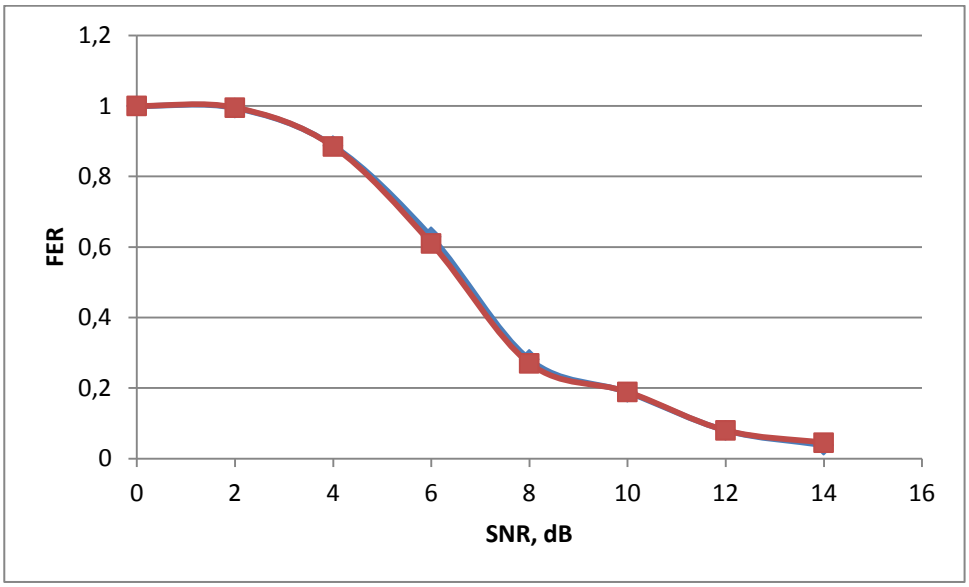


Рис. 4.18. Зависимость FER от отношения сигнал/шум

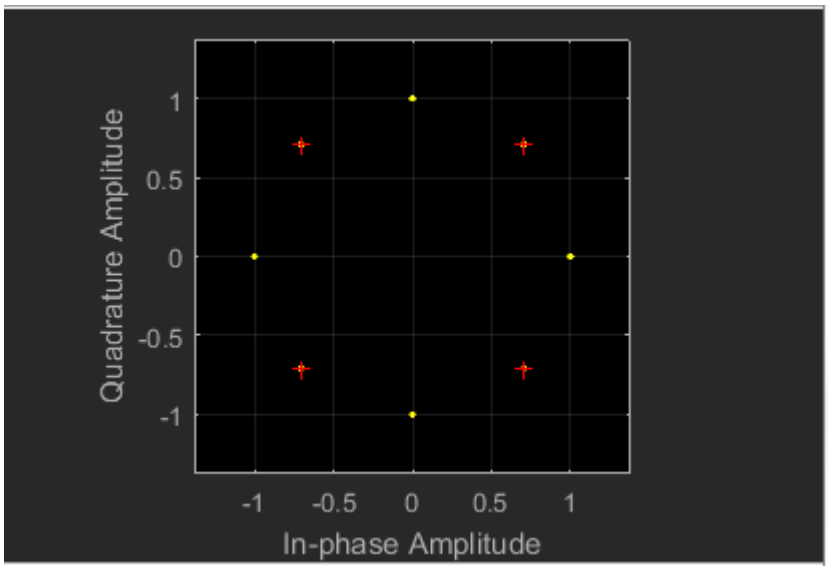


Рис. 4.19. Диаграмма созвездия 8-PSK

Таким образом, в данной работе на примере сверточного кодирования были рассмотрены преимущество и технология применения треллис-модуляции для повышения скорости передачи данных.

Переход в режим треллис-модуляции позволил сохранить тот же уровень помехоустойчивости (3 дБ) при увеличении скорости передачи информации в 2 раза. Это достигается с помощью совместного использования сверточного кода со скоростью 2/3 и фазовой манипуляции 8PSK. Использование треллис-модуляции является простым и эффективным решением для повышения скорости передачи в цифровых высокоскоростных системах связи.

4.2. Исследование сигнально-кодовой конструкции на базе системы с ортогональным частотным мультиплексированием и пространственно-временным кодированием OFDM - MIMO

В последнее время все активнее ощущается рост беспроводных систем связи. Развитие технологий мобильных устройств, беспроводных локальных сетей (WLAN) и стремительный рост Интернет вызывают все возрастающую потребность в увеличении емкости мобильных сетей. Также наблюдается все большая интеграция сотовых сетей с сетями передачи данных, например GPRS в GSM сетях, а также сети 3G. Однако существующие технологии не могут удовлетворить новых потребностей по емкости сети, скорости передачи и стоимости услуг.

OFDM - ортогональное частотное мультиплексирование – это схема модуляции, которая позволяет быстро и эффективно передавать данные даже в каналах с многолучевым распространением сигнала. Передача ведется одновременно на большом количестве несущих частот. Эти несущие имеют небольшое разнесение по частоте и их спектры образуют групповой спектр OFDM сигнала. Частотное разнесение и синхронизация подобраны так,

чтобы несущие были ортогональны между собой, то есть не оказывали влияния друг на друга, несмотря на перекрытие по спектру. Другой важный подход, позволяющий существенно улучшить, по сравнению с традиционными системами, спектральную эффективность и помехоустойчивость системы - применение разнесения как на передающей, так и на приемной стороне путем использования нескольких передающих и нескольких приемных антенн (MIMO). Эта технология уже применяется в некоторых современных стандартах, например IEEE 802.16 и LTE.

Особенности распространения сигнала

В мобильной радиосвязи параметры канала изменяются во времени, поскольку перемещение абонента в пространстве приводит к изменению условий распространения сигнала. Скорость изменения условий распространения определяет скорость замираний. В большинстве случаев мобильный канал связи характеризуется отсутствием прямой видимости между передатчиком и приемником, особенно в условиях плотной городской застройки. Если имеется большое число многократно отраженных лучей и отсутствует прямая видимость, огибающая полученного сигнала может быть статистически описана с помощью релеевской функции плотности вероятности. Характер замираний в этом случае называют релеевским. На Рис. 4.20 показан многолучевой характер распространения сигнала вследствие отражений.

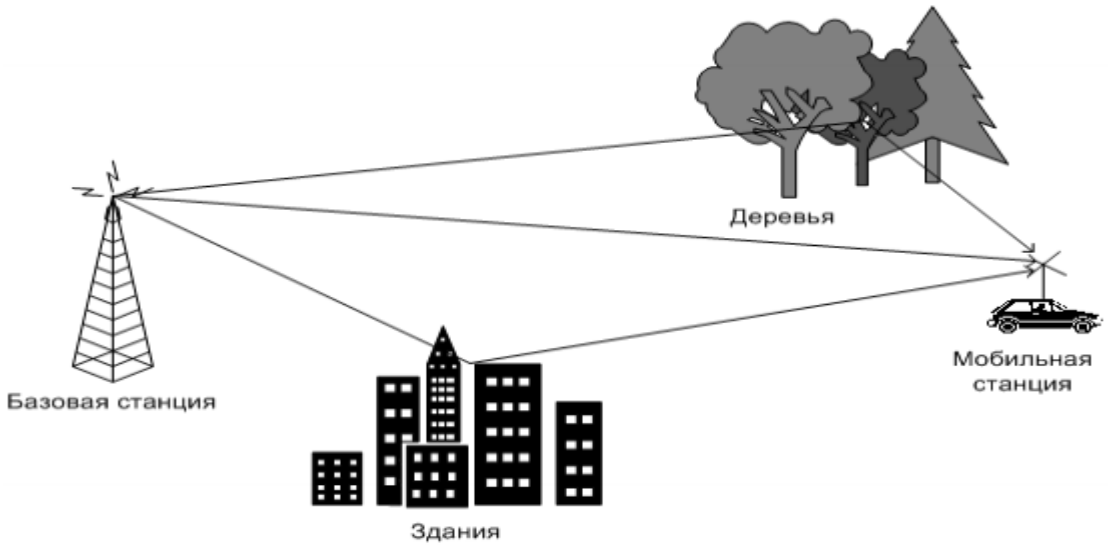


Рис. 4.20. Многолучевой характер распространения сигнала в мобильном канале связи.

В мобильном канале связи сигнал подвергается воздействию следующих основных типов искажений:

• Многолучевое распространение возникает в результате отражения, рассеяния и дифракции электромагнитных волн при взаимодействии с различными объектами в

пространстве. Таким образом, сигнал в приемной антенне содержит сумму волн с различными задержками, амплитудами и фазами. Суперпозиция этих волн приводит к

изменению амплитуды и фазы принимаемого сигнала.

• Допплеровский сдвиг возникает из-за перемещения абонента в пространстве. Возникает изменение амплитуды и фазы принимаемого сигнала во времени. Даже небольшие перемещения на расстояния, соизмеримые с длиной волны передаваемого сигнала, могут вызывать существенные изменения параметров принимаемого сигнала. Изменения параметров сигнала во времени, вызванные движением абонента и многолучевым распространением радиоволн называются быстрыми замираниями. Величина допплеровского сдвига пропорциональна частоте передачи и скорости движения. Чем меньше разнесение между несущими в сигнале OFDM, тем более восприимчива система к допплеровскому сдвигу частоты. Различные режимы передачи позволяют получить компромисс между уровнем восприимчивости к межсимвольной интерференции и допплеровскому сдвигу частоты.

• Затенение вызывается объектами, такими как здания, холмы, деревья и т.п., оказывающимися на пути сигнала и ограничивающими прямую видимость между передатчиком и приемником. Изменения параметров сигнала во времени, вызванные затенением обычно относят к медленным замираниям.

• Потери в тракте характеризуются как зависимость падения средней мощности сигнала от расстояния между передатчиком и приемником. В открытом пространстве средняя мощность уменьшается пропорционально квадрату расстояния между передатчиком и приемником. В мобильном радиоканале, где прямая видимость часто отсутствует, эта характеристика часто представляется как функция степени от 3 до 5.

Искажения, вызванные потерями в тракте и затенением (медленные замирания) обычно компенсируются с помощью систем управления мощностью. Борьба с быстрыми замираниями, вызванными движением и многолучевым распространением волн, является более сложной задачей и требует применения сложной обработки сигнала как на приемной, так и на передающей стороне.

Следствием многолучевого распространения радиоволн является искажение формы принимаемого сигнала. Многолучевая интерференция присуща любому типу сигналов, но особенно негативно она сказывается на широкополосных сигналах. Дело в том, что при использовании широкополосного сигнала в результате интерференции определенные частоты складываются синфазно, что приводит к увеличению уровня сигнала, а некоторые, наоборот, противофазно, вызывая ослабление сигнала на данной частоте.

Говоря о многолучевой интерференции, возникающей при передаче сигналов, различают два крайних случая. В первом из них максимальная задержка между различными сигналами не превышает времени длительности одного символа и интерференция возникает в пределах одного передаваемого символа. Во втором случае максимальная задержка между различными сигналами больше длительности одного символа, в результате интерференции складываются сигналы, представляющие разные символы, и возникает так называемая межсимвольная интерференция (InterSymbol Interference, ISI).

Межсимвольная интерференция наиболее существенно влияет на искажение сигнала. Поскольку символ - это дискретное состояние сигнала, характеризующееся значениями частоты несущей, амплитуды и фазы, то для различных символов меняются амплитуда и фаза сигнала, а значит, восстановить исходный сигнал крайне сложно. Чтобы хотя бы частично компенсировать эффект многолучевого распространения, используются частотные эквалайзеры, однако по мере роста скорости передачи данных либо вследствие увеличения символьной скорости, либо из-за усложнения схемы кодирования эффективность использования эквалайзеров падает.

В традиционных системах с одной несущей борьба с межсимвольной интерференцией обычно ведется путем адаптивного выравнивания. Этот процесс использует адаптивную фильтрацию для аппроксимации импульсного отклика канала. Затем инверсный фильтр используется для воссоздания копий искаженных символов. Этот процесс, однако, довольно сложный, ввиду высокой сложности адаптивного эквалайзера. В случаях, когда межсимвольная интерференция становится довольно высокой, процесс также теряет эффективность.

Модель системы связи с ортогональным частотным мультиплексированием (OFDM)

При высоких скоростях передачи применяется метод передачи данных, который состоит в том, что поток передаваемых данных распределяется по множеству частотных подканалов и передача ведется параллельно на всех этих подканалах. При этом высокая скорость передачи достигается именно за счет одновременной передачи данных по всем каналам, а скорость передачи в отдельном подканале вполне может быть невысокой. Поскольку в каждом из частотных подканалов скорость передачи можно сделать не слишком высокой, это создает предпосылки для эффективного подавления межсимвольной интерференции.

При частотном разделении каналов необходимо, чтобы ширина отдельного канала была, с одной стороны, достаточно узкой для минимизации искажения сигнала в пределах

отдельного канала, а с другой - достаточно широкой для обеспечения требуемой скорости передачи. Кроме того, для экономного использования всей полосы канала, разделяемого на подканалы, желательно как можно более плотно расположить частотные подканалы, но при этом избежать межканальной интерференции, чтобы обеспечить полную независимость каналов друг от друга. Частотные каналы, удовлетворяющие перечисленным требованиям, называются ортогональными. Несущие сигналы всех частотных подканалов (а точнее, функции, описывающие эти сигналы) ортогональны друг другу. Важно, что хотя сами частотные подканалы могут частично перекрывать друг друга, ортогональность несущих сигналов гарантирует независимость каналов друг от друга, а следовательно, и отсутствие межканальной интерференции.

Рассмотренный способ деления широкополосного канала на ортогональные частотные подканалы называется ортогональным частотным мультиплексированием (OFDM). Сигнал в системе с ортогональным частотным мультиплексированием имеет разбиение на множество несущих, что обеспечивает небольшое количество символов на одну несущую и снижает межсимвольную интерференцию. На рисунке 4.21 представлен спектр сигнала при модуляции OFDM.

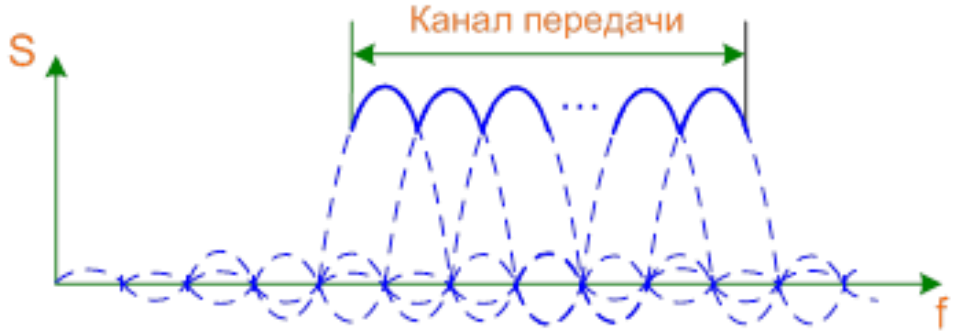


Рис. 4.21. Спектр сигнала при модуляции OFDM

Дополнительно применяется защитный интервал - циклический префикс, добавляемый в начало каждого символа. Для эффективной работы системы, использующей такой подход, максимальная задержка в канале не должна превышать длину циклического префикса. Высокая эффективность систем OFDM при работе в каналах с многократными отражениями делает их пригодными для высокоскоростных систем передачи данных в наземных системах связи.

На рисунке 4.22 представлена структура OFDM символа.

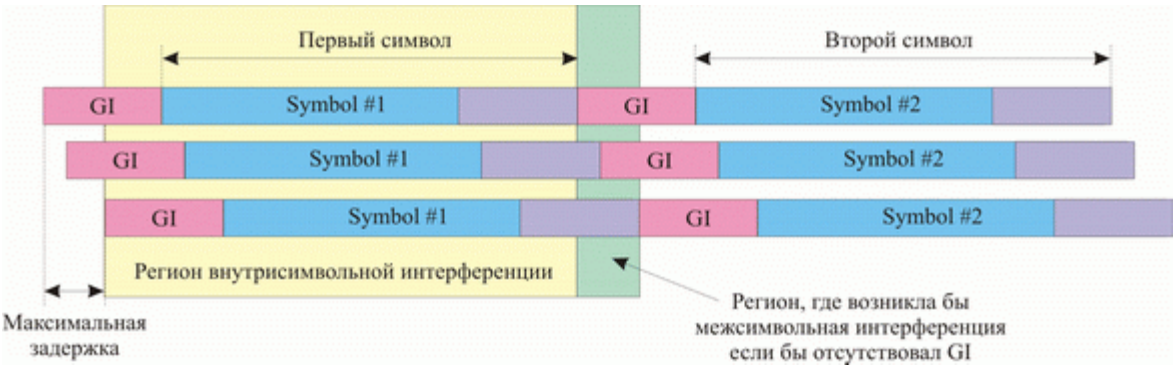


Рис. 4.22. Структура OFDM символа.

На рисунке 4.24 показана обобщенная структура системы связи с ортогональным частотным мультиплексированием. Данные пользователя вначале поступают в блок помехоустойчивого кодирования, а затем на модулятор. После модуляции данные в виде комплексных символов поступают в блок Обратного быстрого преобразования Фурье (ОБПФ), где происходит формирование сигнала OFDM с использованием M поднесущих. На рисунке 4.23 представлена временная форма сигнала после прохождения ОБПФ.

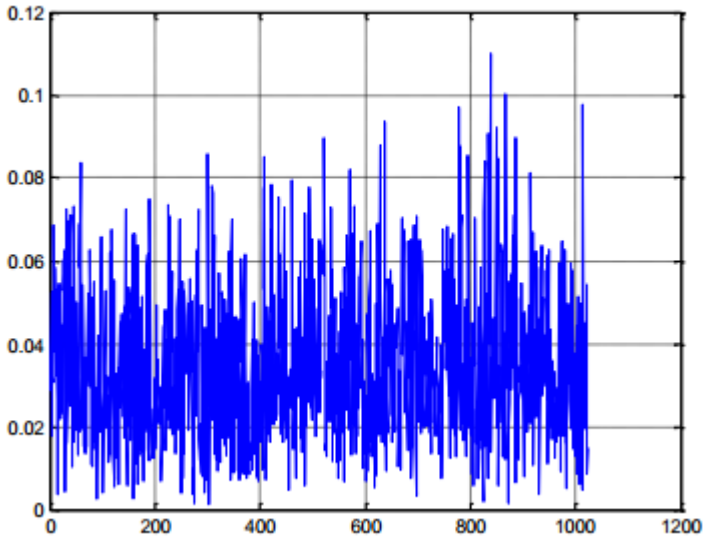


Рис. 4.23. Временная форма сигнала

Символ OFDM состоит из основной информационной части и циклического префикса, который формируется путем копирования последних L отсчетов в начало кадра. Далее последовательность символов OFDM преобразуется в аналоговый сигнал и предается по каналу связи. Длительность циклического префикса выбирается таким образом, чтобы быть больше, чем длительность импульсного отклика канала связи. Таким образом, символ OFDM имеет длительность $(M+L)T_0$, где T_0 - период дискретизации в системе.

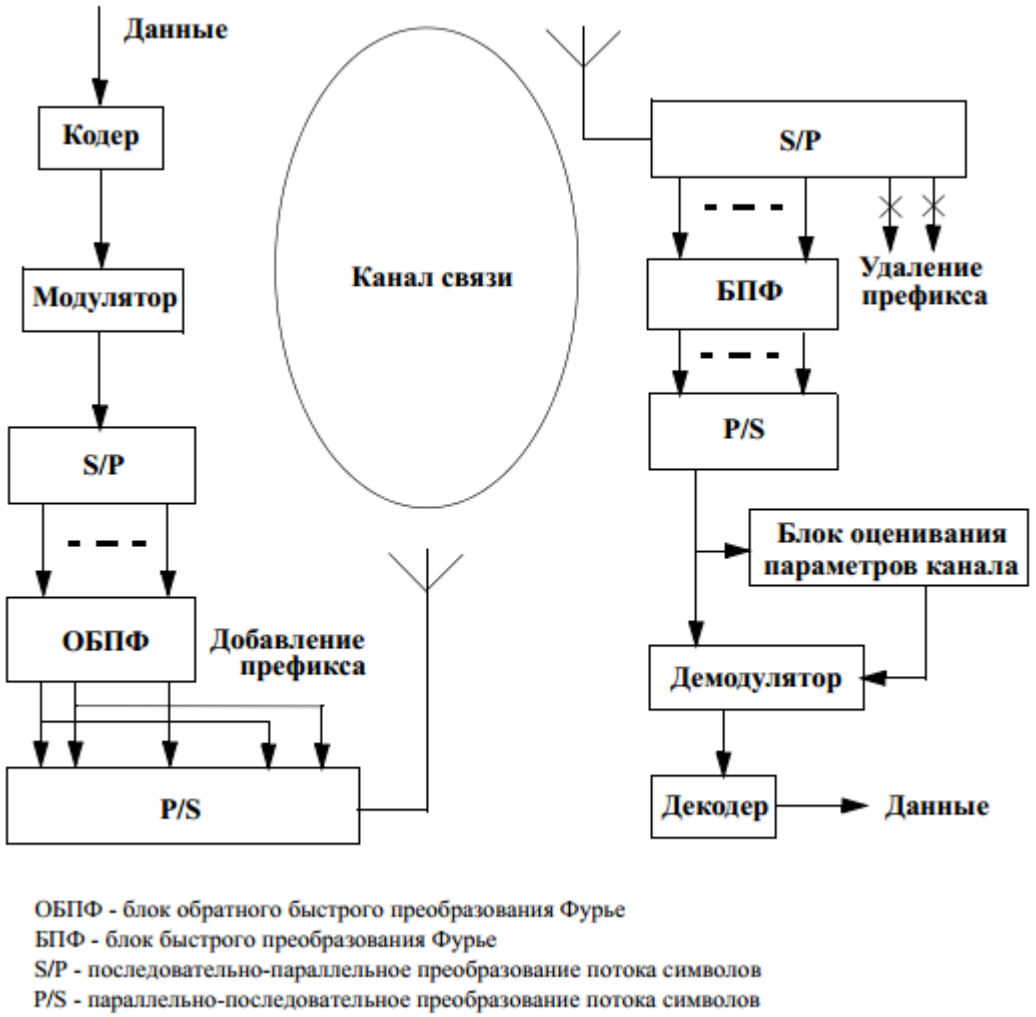


Рис. 4.24. Структурная схема системы связи с ортогональным частотным мультиплексированием

На приемной стороне, сигнал после дискретизации и удаления префикса поступает в блок Быстрого преобразования Фурье (БПФ), где осуществляется демодуляция сигнала OFDM. После преобразования параллельного потока символов в последовательный, символы поступают в демодулятор, а также в блок оцениваня параметров канала. Оценки параметров канала необходимы для работы демодулятора, поэтому выход блока оценивания параметров канала соединен с демодулятором. Дальнейшие преобразования сигнала зависят от структуры конкретной системы и не относятся к основной концепции OFDM. В реальных условиях работы системам беспроводной подвижной связи с FDM приходится работать в условиях, когда в канале связи сигнал подвергается воздействию замираний. Поэтому для получения приемлемой помехоустойчивости требуется использование помехоустойчивого кодирования.

В процессе развития сетей передачи данных постоянно увеличиваются требования к скорости передачи информации и качеству предоставляемых сервисов. Данный механизм можно наблюдать и для сетей передачи данных, где средой передачи является радиоканал.

Из-за этого в значительной степени усложняются сигналы, которые используют для передачи информации. Одним из перспективных видов является ортогональное частотное мультиплексирование OFDM.

В современных системах связи, например, в сотовых системах связи, высокоскоростных локальных вычислительных сетях и др., существует необходимость повышения пропускной способности. Пропускная способность может быть увеличена путём расширения полосы частот или повышения излучаемой мощности. Тем не менее, применимость этих методов ограничена из-за требований биологической защиты, ограниченной мощности источника питания (в мобильных устройствах) и электромагнитной совместимости. Поэтому если в системах связи эти подходы не обеспечивают необходимую скорость передачи данных, то эффективным может оказаться применение адаптивных антенных решёток со слабо коррелированными антенными элементами. Системы связи с такими антеннами получили название систем MIMO.

Технология OFDM

OFDM основан на разделении потока входных данных на множество параллельных потоков, каждый из которых передается на своей несущей (ортогональной) частоте. Это обеспечивает высокие скорость и помехоустойчивость передачи информации, в частности, по отношению к провалам в спектре передаваемых сигналов, так как узкополосное затухание может исключить только одну или несколько несущих частот из их большого числа (сотни - тысячи). Поскольку модуляция OFDM использует для передачи ортогональные несущие колебания, то возможна демодуляция модулированных сигналов даже в условиях частичного перекрытия полос отдельных несущих. Наличие большого числа несущих не позволяет реализовать модуляцию OFDM непосредственно, т.е. с использованием нескольких тысяч синтезаторов несущих колебаний и нескольких тысяч модуляторов. Поэтому для уменьшения объема оборудования учитывают, что модуляция OFDM представляет собой обратное преобразование Фурье, а демодуляция прямое преобразование Фурье, и применяют быстрые алгоритмы двух этих преобразований, допускающие более простую аппаратную реализацию по сравнению с непосредственной реализацией алгоритмов модуляции OFDM. Модуляция OFDM используется в системах цифрового телевидения, системах сотовой связи WiMAX, MobileWiMAX, MBWA, автоматизированных системах контроля и учета электроэнергии, системах типа "интеллектуальный дом" и др. На ней базируются стандарты беспроводной связи IEEE 802.11a,e,g,n; 802.16a,d,e; 802.20.

Ортогональное частотное разделение каналов

При беспроводной передаче сигналов один и тот же сигнал в результате многократных отражений может поступать в приемник различными путями. Поэтому в точке приема результирующий сигнал представляет собой суперпозицию (интерференцию) многих сигналов с различными амплитудами и начальными фазами. Применительно к многолучевой интерференции, возникающей при передаче сигналов, различают два крайних случая. В первом случае максимальная задержка между различными сигналами не превышает длительности одного символа, и интерференция возникает в пределах одного передаваемого символа. Во втором случае максимальная задержка между различными сигналами больше длительности одного символа, и в результате интерференции складываются сигналы, представляющие разные символы. Вследствие этого возникает межсимвольная интерференция, которая наиболее сильно сказывается на искажении сигнала. Для того, чтобы частично компенсировать эффект многолучевого распространения, применяют частотные эквалайзеры, однако по мере роста скорости передачи данных либо за счет увеличения символьной скорости, либо за счет усложнения схемы кодирования, эффективность их применения падает. Поэтому для достижения высокой скорости передачи данных используют другой подход, состоящий в том, что поток передаваемых данных распределяется по множеству частотных подканалов и передача ведется параллельно на всех этих подканалах. При этом достигается высокая скорость передачи за счет одновременной передачи данных по всем каналам, причем скорость передачи в отдельном подканале может быть и невысокой. Это создает предпосылки для эффективного подавления межсимвольной интерференции. При частотном разделении каналов необходимо, чтобы ширина каждого канала была, с одной стороны, достаточно узкой для минимизации искажения сигнала 3 в его пределах, а с другой - достаточно широкой для обеспечения требуемой скорости передачи. Кроме того, для экономного использования всей полосы канала, разделяемого на подканалы, желательно как можно плотнее расположить частотные подканалы, но при этом избежать межканальной интерференции для обеспечения полной независимости каналов друг от друга. Перечисленным требованиям удовлетворяют ортогональные частотные каналы. Функции, описывающие несущие сигналы всех этих каналов, ортогональны друг другу, т.е. для них выполняется условие:

$$\int_0^T \sin 2\pi f_l(t) \cdot \sin 2\pi f_m(t) dt = 0, \; k \neq l$$

,

где T - длительность передаваемого символа,
fl и fk - частоты l-го и k-го несущих сигналов соответственно.

Ортогональность несущих сигналов обеспечивает частотную независимость каналов друг от друга и, следовательно, отсутствие межканальной интерференции. Рассмотренный способ деления широкополосного канала на ортогональные частотные подканалы называется ортогональным частотным разделением с мультиплексированием или OFDM- модуляцией.

Ортогональность несущих сигналов обеспечивается только тогда, когда за время длительности одного символа T несущий сигнал будет совершать целое число колебаний. Так как каждый символ длительности T передается ограниченной по времени синусоидальной функцией (рис. 2.1.1), то ее спектр описывается функцией вида

$$\frac{\sin 2\pi(f - f_i)}{2\pi(f - f_i)},$$

где fi- центральная (несущая) частота i-го канала.

Такой же функцией описывается и форма частотного подканала.

Несмотря на частичное перекрытие частотными подканалами друг друга (рис.1.1), ортогональность несущих сигналов обеспечивает их частотную независимость каналов друг от друга и, следовательно, отсутствие межканальной интерференции.

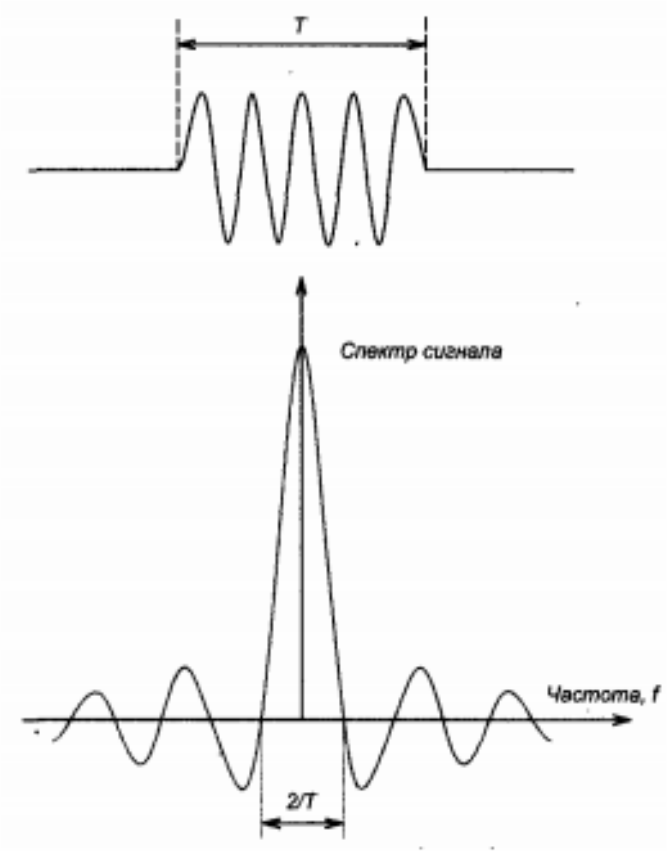


Рисунок 4.5 - Символ длительностью T и его спектральное представление.

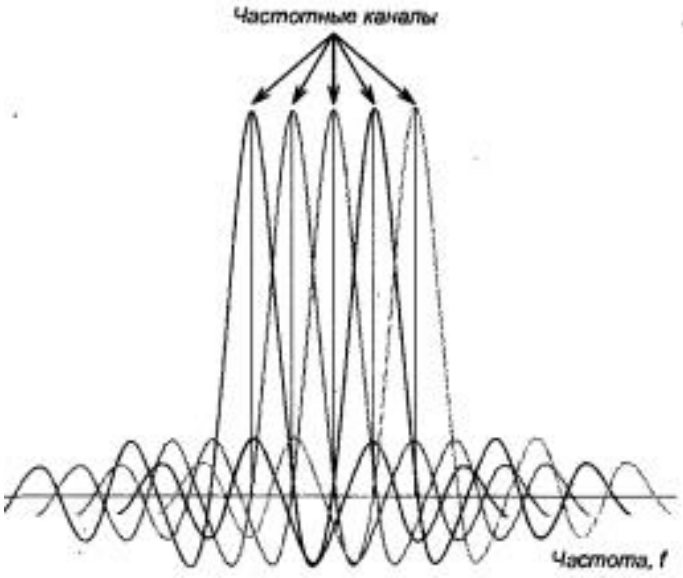


Рисунок 4.26 - Частотное разделение каналов с ортогональными несущими сигналами.

Принцип организации канала OFDM на примере DVB-T

Характеристики канала передачи, к сожалению, не остаются постоянными во времени, но в течение короткого промежутка времени эти характеристики для наземного канала можно считать постоянными. Используя эту особенность, в системе OFDM имеется возможность применить расщепление наземного канала передачи во времени и по частоте (см. рис. 1.3). В результате радиочастотный канал организуется в виде набора узких частотных полос и в виде коротких во времени смежных «временных сегментов»

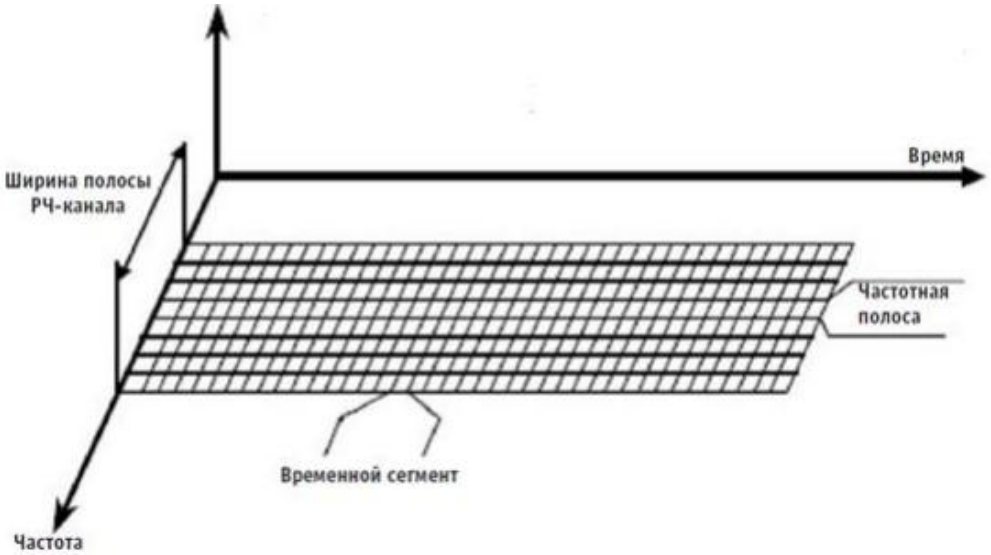


Рисунок 4.27 - Расщепление канала

Каждая частотно-временная ячейка имеет свою собственную поднесущую (см. рис. 4.28). Набор поднесущих в определенном временном сегменте называется символом OFDM. Для устранения взаимных помех между поднесущими расстояние (промежуток) между ними

выбирается равным обратной величине длительности символа: в этом случае поднесущие являются ортогональными.

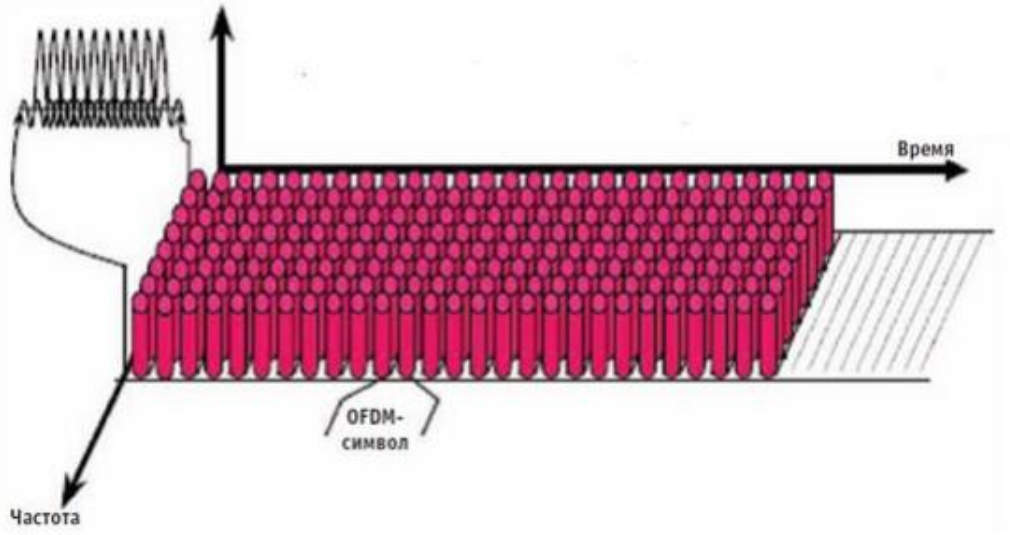


Рисунок 4.28 - Ввод поднесущих частот.

Поскольку эхо-сигналы представляют собой задержанные во времени копии основного сигнала, начало данного символа OFDM подвергается «загрязнению» задержанным окончанием предыдущего (взаимные помехи между символами). Для устранения этого эффекта между двумя соседними символами OFDM вводится защитный интервал (см. рис. 4.29). Во время защитного интервала приемные устройства игнорируют поступающий сигнал, что приводит к снижению пропускной способности канала передачи.

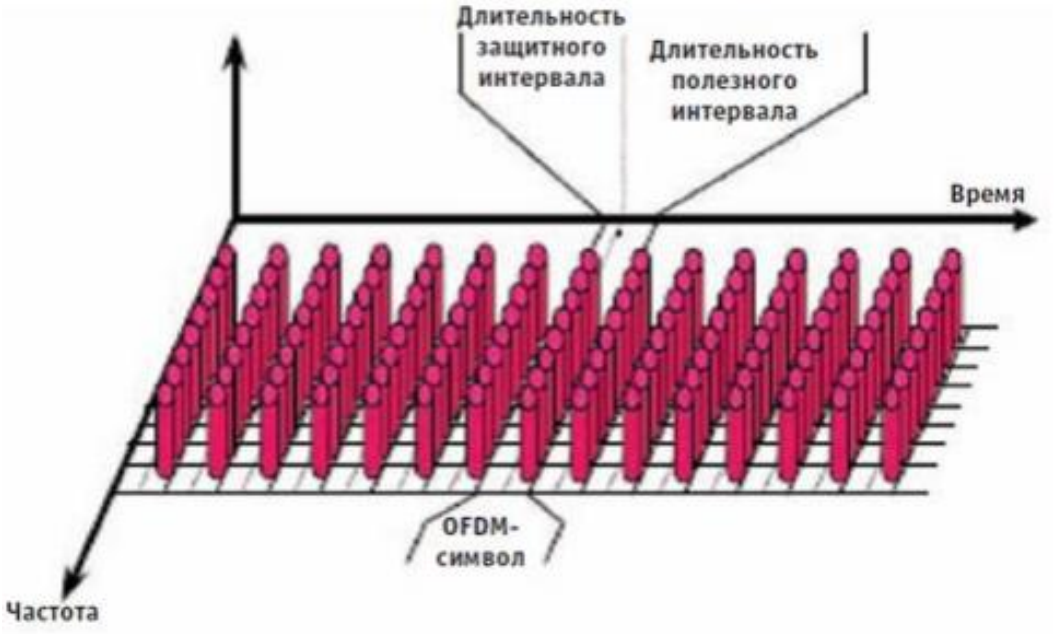


Рисунок 4.30 - Ввод защитного интервала.

Чтобы осуществить надлежащим образом демодуляцию сигнала, приемные устройства должны произвести его выборку во время полезного периода символа OFDM (но не во время защитного интервала). Но тогда необходимо ввести временнòе окно по отношению к моменту, когда передается в эфир каждый символ OFDM. В системе DVB-T используются «пилотные» поднесущие, равномерно распределенные в канале передачи в виде маркеров синхронизации (см. рис. 4.31)

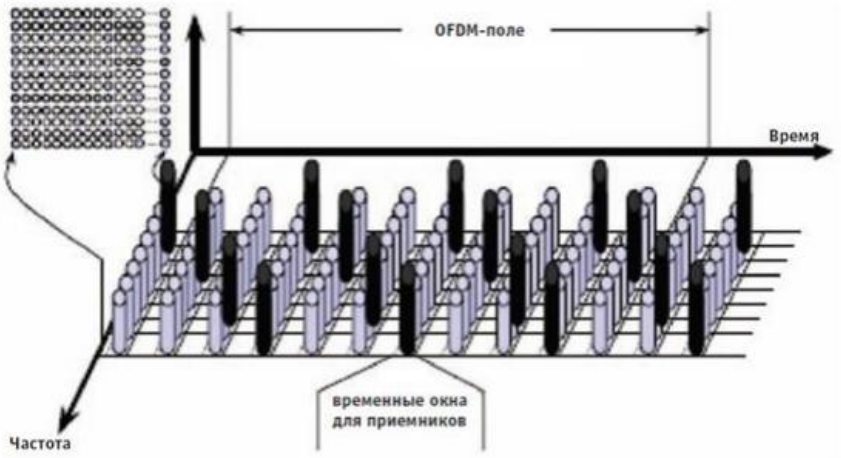


Рисунок 4.31 - Маркеры синхронизации.

Итак, общая длительность Ts OFDM-символа представляет собой сумму длительностей полезной части Tu и защитного интервала Tg. Расстояние по частоте между соседними несущими частотами OFDM-сигнала равно 1/Tu (поскольку чтобы осуществить надлежащим образом демодуляцию сигнала, приемные устройства должны произвести его выборку во время полезного периода символа OFDM, но не во время защитного интервала). Защитный интервал располагается перед полезной частью OFDM-символа.

Структурная схема модулятора

Принцип OFDM-модуляции заключается в следующем. В полосе канала связи передается множество несущих, каждая из которых модулируется, например, с использованием QAM-модуляции, частью общего цифрового потока. До преобразования спектра такого сигнала цифровой поток разбивается на последовательности, каждая из которых соответствует передаче kRa битов информации, где Ra — число активных несущих, k — коэффициент используемой QAM-модуляции (или число битов информации, передаваемой на каждой активной несущей). Длительность T0 интервала, на котором передаются все указанные kRa битов информации, определяет минимальную частоту несущей fU = 1/TU и интервал между несущими, т. е. частотный интервал (Ra + Rn)fU, где Rn - число пассивных несущих, характеризует групповой спектр мощности радиосигнала.

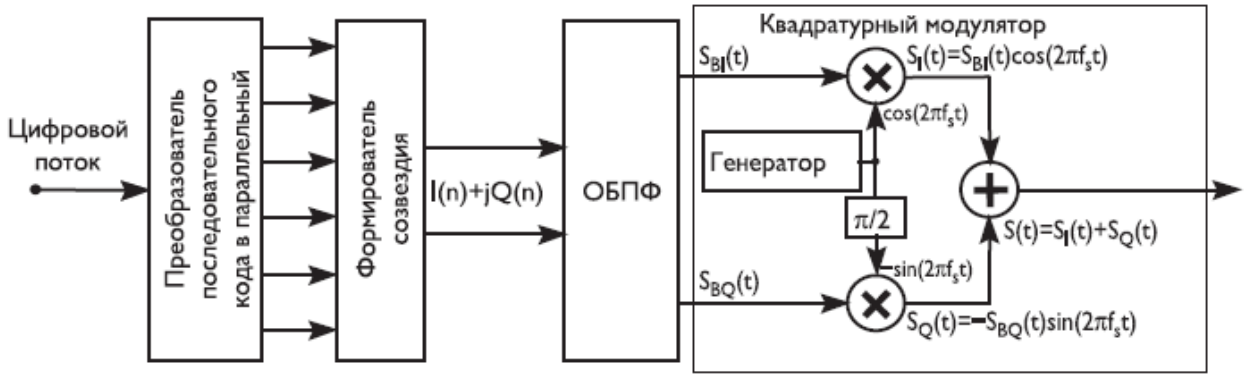


Рисунок 4.32 - Структурная схема идеального OFDM-модулятора.

Цифровой поток поступает на вход преобразователя последовательной информации в параллельную. На выходе этого преобразователя формируется код, состоящий из k битов и соответствующий используемой QAM-модуляции несущих (k = 2 при QPSK, k = 4 при QAM-16, k = 6 при QAM-64 и т.д.). Последовательно каждые k битов преобразуются в параллельный код, который подается на формирователь созвездия, преобразующий этот код в значения соответствующих векторов звездной диаграммы, как показано, например, на рис. 4.33.

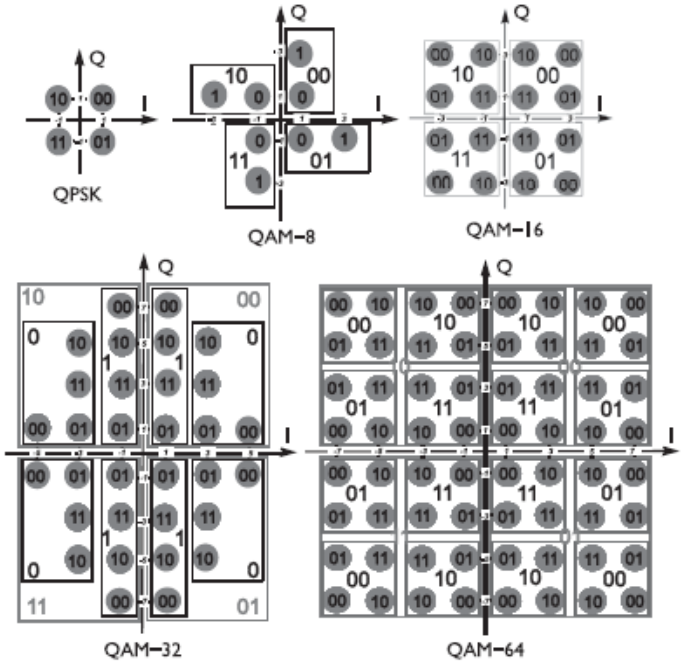


Рисунок 4.33 - Варианты сигнальных созвездий: QPSK, QAM-8, QAM-16, QAM-32 и QAM-64.

Поток битов трансформируется в формирователе созвездия в поток I и Q сигналов: S(n) = I(n) + jQ(n), 1 _ n _ 2N + 1, где 2N + 1 = Ra + Rn. Блок обратного быстрого преобразования Фурье (ОБПФ) преобразует последовательности во временную комплексную функцию: SB(t) =2_N+1 n=1

$$[I(n) + jQ(n)] \exp(2\pi jnfUt) = SBI(t) + jSBQ(t),$$

где SBI(t) =2_N+1 n=1

[I(n) · cos(2πnfUt) − Q(n) · sin(2πnfUt)] —синфазная составляющая сигнала, SBQ(t) =2_N+1 n=1 [I(n) · sin(2πnfUt) + Q(n) · cos(2πnfUt)] —квадратурная составляющая сигнала (преобразование синфазной составляющей по Гильберту).

Связь между частотой fU и количеством несущих частот 2N + 1 сигнала в используемой полосе частот можно представить графиком, приведенным на рис.1.9, на котором изображены амплитуды отсчетов составляющих частот nf0 (n = 1, 2, . . . , 2N + 1).

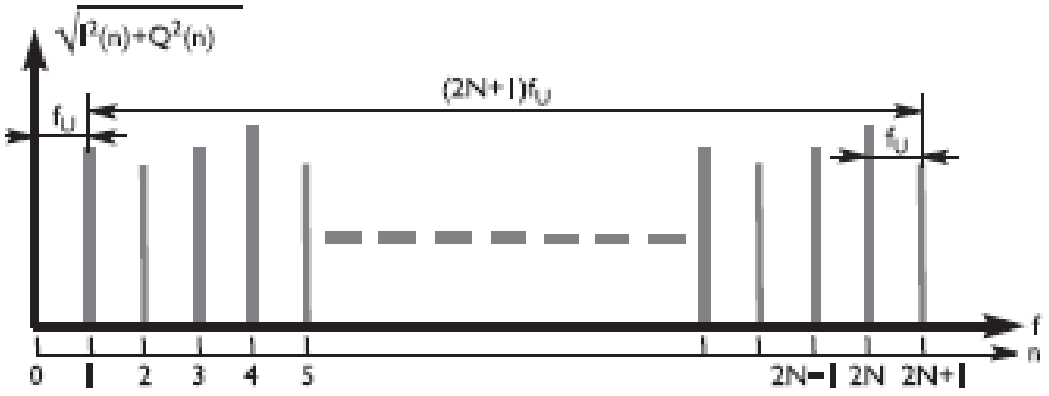


Рисунок 4.34 - График спектральных составляющих ОБПФ.

В соответствии со схемой квадратурного модулятора эти два сигнала SBI(t) и SBQ(t) перемножаются соответственно на сдвинутые по фазе на 90◦ синусоидальные сигналы. На выходах перемножителей формируются две составляющие:

$$S_I(t) = S_{BI}(t) \cdot \cos(2\pi f_s t) = \sum_{n=1}^{2N+1} \left\{ \frac{1}{2} I(n) \cdot \left[\cos 2\pi (nf_U + f_s) + \right. \right.$$

$$\left. \left. + \cos 2\pi (nf_U - f_s) \right] - \frac{1}{2} Q(n) \cdot \left[\sin 2\pi (nf_U + f_s) + \sin 2\pi (nf_U - f_s) \right] \right\},$$

$$S_Q(t) = -S_{BQ}(t) \cdot \sin(2\pi f_s t) = \sum_{n=1}^{2N+1} \left\{ \frac{1}{2} I(n) \cdot \left[\cos 2\pi (nf_U + f_s) - \right. \right.$$

$$\left. \left. - \cos 2\pi (nf_U - f_s) \right] - \frac{1}{2} Q(n) \cdot \left[\sin 2\pi (nf_U + f_s) - \sin 2\pi (nf_U - f_s) \right] \right\}.$$

В результате суммирования этих двух составляющих окончательно формируется выходной сигнал OFDM-модулятора. При этом разностные частоты (nf0 − fs) взаимно исключаются.

Структурная схема демодулятора

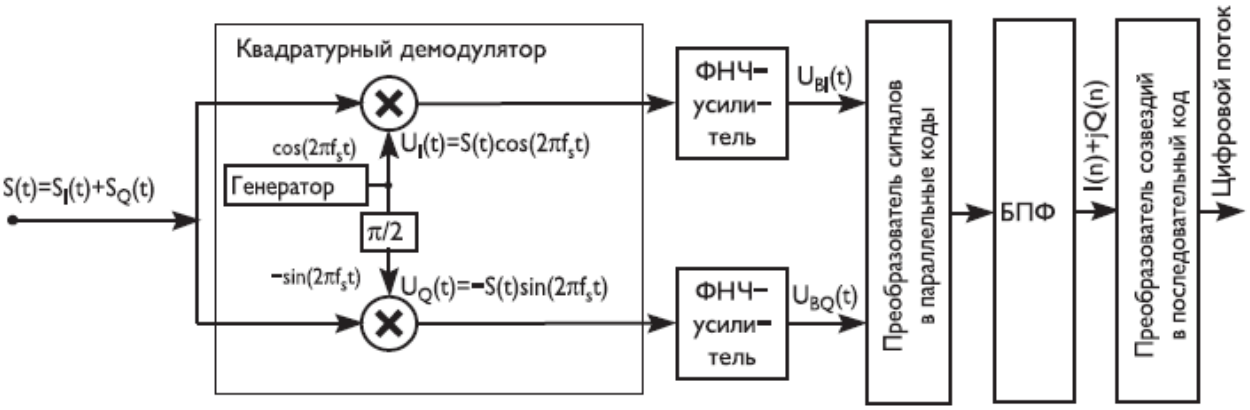


Рисунок 4.35 - Структурная схема идеального демодулятора.

Если частота генератора fs выбрана равной fs = fc + (N + 1) · fU, где fc -центральная частота радиоканала, то выходной сигнал определяется следующим соотношением:

$$S(t) = \sum_{n=1}^{2N+1} \{I(n) \cdot \cos[2\pi(f_s + nf_U)t] - Q(n) \cdot \sin[2\pi(f_s + nf_U)t]\} =$$

$$= \sum_{n=-N}^{N} \{I(n) \cdot \cos[2\pi(f_c + nf_U)t] - Q(n) \cdot \sin[2\pi(f_c + nf_U)t]\}.$$

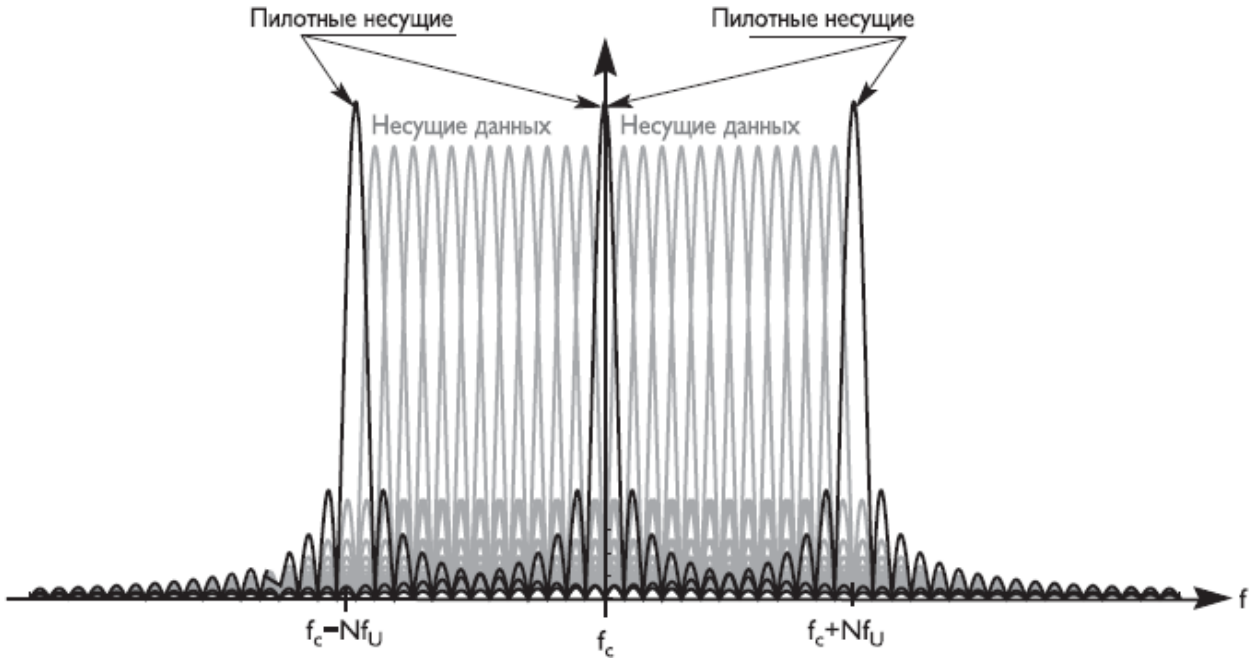


Рисунок 4.36 - Групповой спектр несущих частот.

На рис. 4.36 изображена структура группового спектра несущих частот OFDM-сигнала. Здесь условно показаны составляющие синусоидальные сигналы, промодулированные соответствующими данными созвездий дискретной QAM-модуляции. В

составе сигнала имеются также специальные пилотные несущие, на которых передается информация о параметрах системы. Эти пилотные несущие используются также для обеспечения устойчивой синхронизации и коррекции характеристик в демодуляторе.

Таким образом, в системах с OFDM модуляцией передаваемая цифровая информация разделена на большое число низкоскоростных подканалов, длительность тактового интервала передачи каждой несущей весьма велика. Такое построение системы преобразует широкополосный канал с одной несущей в большое число независимых узкополосных каналов с частотным разделением, что упрощает коррекцию параметров затухающего сигнала. Более того, ряд групп несущих может быть полностью подавлен при приеме, если дополнительно вводится корректирующее кодирование данных в сочетании с временным и частотным перемежением. При таком построении системы модуляцию часто называют COFDM (Coded Orthogonal Frequency DivisionMultiplexing — кодированное ортогональное частотное мультиплексирование, разновидность технологии OFDM, сочетающая канальное кодирование и OFDM).

На рис. 4.36 приведена структурная схема идеального демодулятора, содержащего квадратурный демодулятор и ряд преобразователей, формирующих передаваемую цифровую информацию.

В квадратурном демодуляторе входной сигнал S(t) перемножается с двум сдвинутыми на 90◦ синусоидальными сигналами опорной частоты fs. На двух выходах выделяются сигналы:

$$U_I(t) = S(t) \cdot \cos(2\pi f_s t) = \sum_{n=1}^{2N+1} \left\{ \frac{1}{2} I(n) \cdot [\cos 2\pi n f_U t + \cos 2\pi (2f_s + n f_U) t] - \right.$$
$$\left. - \frac{1}{2} Q(n) \cdot [\sin 2\pi n f_U t + \sin 2\pi (2f_s + n f_U) t] \right\},$$

$$U_Q(t) = S(t) \cdot [-\sin(2\pi f_s t)] = \sum_{n=1}^{2N+1} \left\{ \left(\frac{1}{2} I(n) \cdot [\sin 2\pi n f_U t - \sin 2\pi (2f_s + n f_U) t] + \right. \right.$$
$$\left. \left. + \frac{1}{2} Q(n) \cdot [\cos 2\pi n f_U t - \cos 2\pi (2f_s + n f_U) t] \right\}.$$

Эти два сигнала поступают соответственно на два фильтра нижних частот и усиливаются в два раза. На входы преобразователя в параллельные коды при этом подаются два сигнала:

$$U_{BI}(t) = \sum_{n=1}^{2N+1} [I(n) \cdot \cos(2\pi n f_U t) - Q(n) \cdot \sin(2\pi n f_U t)],$$

$$U_{BQ}(t) = \sum_{n=1}^{2N+1} [I(n) \cdot \sin(2\pi n f_U t) + Q(n) \cdot \cos(2\pi n f_U t)].$$

Таким образом, на вход схемы быстрого преобразования Фурье подается сигнал, который может быть представлен в следующем виде

$$U_B(t) = U_{BI}(t) + jU_{BQ}(t) = \sum_{n=1}^{2N+1} [I(n) + jQ(n)] \exp(2\pi j n f_U t).$$

Очевидно: после преобразования Фурье формируется последовательность, определяющая векторы звездной диаграммы:

$$S(n) = I(n) + jQ(n), \quad 1 \leqslant n \leqslant 2N + 1,$$

которые выходным преобразователем преобразуются в передаваемую цифровую последовательность.

Достоинства и недостатки OFDM

Плюсы

• Высокая эффективность использования радиочастотного спектра, объясняемая почти прямоугольной формой огибающей спектра при большом количестве поднесущих.

• Простая аппаратная реализация: базовые операции реализуются методами цифровой обработки.

• Хорошее противостояние межсимвольным помехам (ISI – intersymbol interference) и интерференции между поднесущими (ICI – intercarrier interference).

• Возможность применения различных схем модуляции для каждой поднесущей, что позволяет адаптивно варьировать помехоустойчивость и скорость передачи информации.

Минусы

• Необходима высокая синхронизация частоты и времени.

• Чувствительность к эффекту Доплера, ограничивающая применение OFDM в мобильных системах.

• Не идеальность современных приёмников и передатчиков вызывает фазовый шум, что ограничивает производительность системы.

• Защитный интервал, используемый в OFDM для борьбы с многолучевым распространением, снижает спектральную эффективность сигнала.

Технология MIMO

MIMO (Multiple Input Multiple Output – множественный вход множественный выход) – это технология, используемая в беспроводных системах связи (WIFI, WI-MAX, сотовые сети связи), позволяющая значительно улучшить спектральную эффективность системы, максимальную скорость передачи данных и емкость сети. Главным способом достижения указанных выше преимуществ является передача данных от источника к получателю через несколько радио соединений, откуда данная технология и получила свое название. Рассмотрим предысторию данного вопроса, и определим основные причины, послужившие широкому распространению технологии MIMO.

Необходимость в высокоскоростных соединениях, предоставляющих высокие показатели качества обслуживания (QoS) с высокой отказоустойчивостью растет от года в год. Этому в значительной мере способствует появление таких сервисов как VoIP (Voice over Internet Protocol), видеоконференции, VoD (Video on Demand) и др. Однако большинство беспроводных технологий не позволяют предоставить абонентам высокое качество обслуживания на краю зоны покрытия. В сотовых и других беспроводных системах связи качество соединения, также как и доступная скорость передачи данных стремительно падает с удалением от базовой станции (BTS). Вместе с этим падает и качество услуг, что в итоге приводит к невозможности предоставления услуг реального времени с высоким качеством на всей территории радио покрытия сети. Для решения данной проблемы можно попробовать максимально плотно установить базовые станции и организовать внутреннее покрытие во всех местах с низким уровнем сигнала. Однако это потребует значительных финансовых затрат что в конечном счете приведет к росту стоимости услуги и снижению конкурентоспособности. Таким образом, для решения данной проблемы требуется оригинальное нововведение, использующее, по возможности, текущий частотный диапазон и не требующее строительства новых объектов сети.

Особенности распространения радиоволн

Для того чтобы понять принципы действия технологии MIMO необходимо рассмотреть общие принципы распространения радио волн в пространстве. Волны, излучаемые различными системами беспроводной радиосвязи в диапазоне свыше 100 МГц, во многом ведут себя как световые лучи. Когда радиоволны при распространении встречают какую-либо поверхность, то в зависимости от материала и размера препятствия часть энергии поглощается, часть проходит насквозь, а оставшаяся – отражается. На соотношение долей поглощенной, отраженной и прошедшей насквозь частей энергий влияет множество внешних факторов, в том числе и частота сигнала. Причем отраженная и прошедшая

насквозь энергии сигнала могут изменить направление своего дальнейшего распространения, а сам сигнал разбивается на несколько волн.

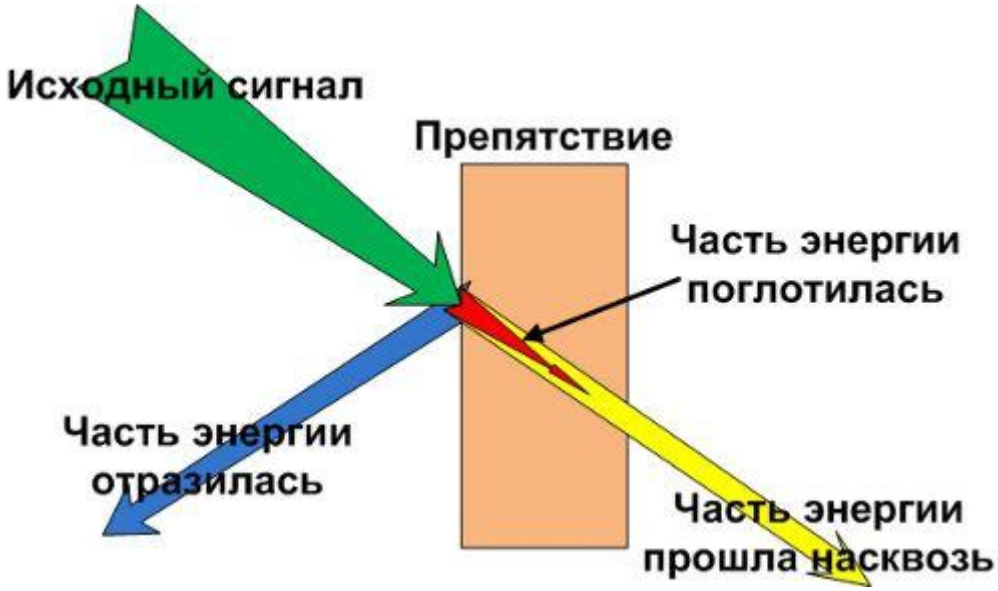


Рисунок 4.37 - Распределение энергии сигнала при взаимодействии с препятствием.

Распространяющийся по вышеуказанным законам сигнал от источника к получателю после встречи с многочисленными препятствиями разбивается на множество волн, лишь часть из которых достигнет приемник. Каждая из дошедших до приемника волн образует так называемый путь распространения сигнала. Причем из-за того, что разные волны отражаются от разного числа препятствий и проходят разное расстояние, различные пути имеют разные временные задержки.

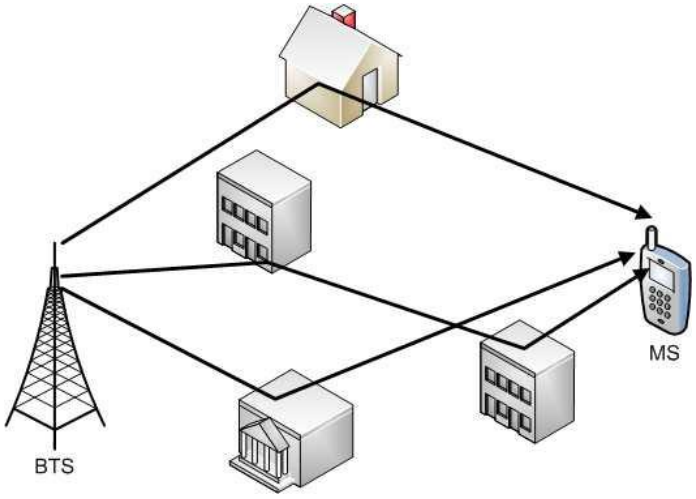


Рисунок 4.38 - Пример многолучевого распространения сигнала.

В условиях плотной городской постройки, из-за большого числа препятствий, таких как здания, деревья, автомобили и др., очень часто возникает ситуация, когда между абонентским оборудованием (MS) и антеннами базовой станции (BTS) отсутствует прямая видимость. В этом случае, единственным вариантом достижения сигнала приемника являются отраженные волны. Однако, как отмечалось выше, многократно отраженный

сигнал уже не обладает исходной энергией и может прийти с запозданием. Особую сложность также создает тот факт, что объекты не всегда остаются неподвижными и обстановка может значительно измениться с течением времени. В связи с этим возникает проблема многолучевого распространения сигнала – одна из наиболее существенных проблем в беспроводных системах связи.

Для борьбы с многолучевым распространением сигналов применяется несколько различных решений. Одной из наиболее распространенных технологий является Receive Diversity – разнесенный прием. Суть его заключается в том, что для приема сигнала используется не одна, а сразу несколько антенн (обычно две, реже четыре), расположенные на расстоянии друг от друга. Таким образом, получатель имеет не одну, а сразу две копии переданного сигнала, пришедшего различными путями. Это дает возможность собрать больше энергии исходного сигнала, т.к. волны, принятые одной антенной, могут не быть принятыми другой и наоборот. Также сигналы, приходящие в противофазе к одной антенне, могут приходить к другой синфазно. Эту схему организации радио интерфейса можно назвать Single Input Multiple Output (SIMO), в противовес стандартной схеме Single Input Single Output (SISO). Также может быть применен обратный подход: когда используется несколько антенн на передачу и одна на прием. Благодаря этому также увеличивается общая энергия исходного сигнала, полученная приемником. Эта схема называется Multiple Input Single Output (MISO). В обеих схемах (SIMO и MISO) несколько антенн устанавливаются на стороне базовой станции, т.к. реализовать разнесение антенн в мобильном устройстве на достаточно большое расстояние сложно без увеличения габаритов самого оконечного оборудования.

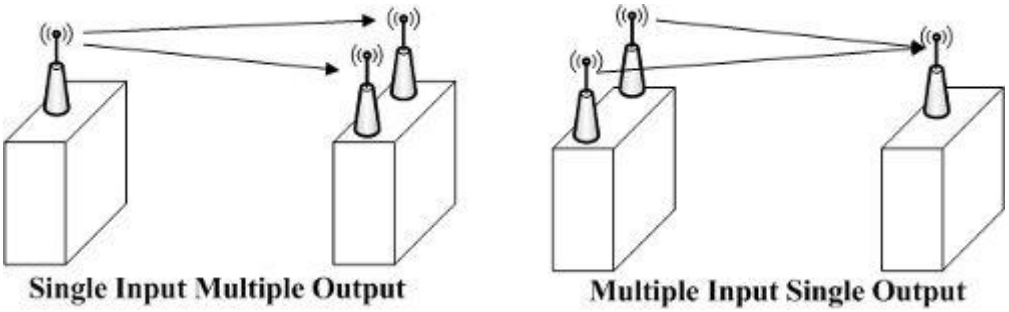


Рисунок 4.39 – Применение нескольких антенн

В результате дальнейших рассуждений мы приходим к схеме Multiple Input Multiple Output (MIMO). В этом случае устанавливаются несколько антенн на передачу и прием. Однако в отличие от указанных выше схем эта схема разнесения позволяет не только бороться с многолучевым распространением сигнала, но и получить некоторые дополнительные преимущества. За счет использования нескольких антенн на передаче и приеме каждой паре, передающей/приемной антенне можно сопоставить отдельный тракт

для передачи информации. При этом разнесенный прием будет выполняться оставшимися антеннами, а данная антенна также будет выполнять функции дополнительной антенны для других трактов передачи. В результате, теоретически, можно увеличить скорость передачи данных во столько раз, сколько дополнительных антенн будет использоваться. Однако существенное ограничение накладывается качеством каждого радио тракта.

Принцип работы и структурная схема MIMO

Как уже отмечалось выше, для организации технологии MIMO необходима установка нескольких антенн на передающей и на приемной стороне. Обычно устанавливается равное число антенн на входе и выходе системы, т.к. в этом случае достигается максимальная скорость передачи данных. Чтобы показать число антенн на приеме и передаче вместе с названием технологии «MIMO» обычно упоминается обозначение «AxB», где A – число антенн на входе системы, а B – на выходе. Под системой в данном случае понимается радио соединение.

Для работы технологии MIMO необходимы некоторые изменения в структуре передатчика по сравнению с обычными системами. Рассмотрим лишь один из возможных, наиболее простых, способов организации технологии MIMO. В первую очередь, на передающей стороне необходим делитель потоков, который будет разделять данные, предназначенные для передачи на несколько низкоскоростных подпорогов, число которых зависит от числа антенн. Например, для MIMO 4x4 и скорости поступления входных данных 200 Мбит/сек делитель будет создавать 4 потока по 50 Мбит/сек каждый. Далее каждый их данных потоков должен быть передан через свою антенну. Обычно, антенны на передаче устанавливаются с некоторым пространственным разнесением, чтобы обеспечить как можно большее число побочных сигналов, которые возникают в результате переотражений. В одном из возможных способов организации технологии MIMO сигнал передается от каждой антенны с различной поляризацией, что позволяет идентифицировать его при приеме. Однако в простейшем случае каждый из передаваемых сигналов оказывается промаркированным самой средой передачи (задержкой во времени, затуханием и другими искажениями).

На приемной стороне несколько антенн принимают сигнал из радиоэфира. Причем антенны на приемной стороне также устанавливаются с некоторым пространственным разнесением, за счет чего обеспечивается разнесенный прием, обсуждавшийся ранее. Принятые сигналы поступают на приемники, число которых соответствует числу антенн и трактов передачи. Причем на каждый из приемников поступают сигналы от всех антенн системы. Каждый из таких сумматоров выделяет из общего потока энергию сигнала только того тракта, за который он отвечает. Делает он это либо по какому-либо заранее

предусмотренному признаку, которым был снабжен каждый из сигналов, либо благодаря анализу задержки, затухания, сдвига фазы, т.е. набору искажений или «отпечатку» среды распространения. В зависимости от принципа работы системы (Bell Laboratories Layered Space-Time - BLAST, Selective Per Antenna Rate Control (SPARC) и т.д.), передаваемый сигнал может повторяться через определенное время, либо передаваться с небольшой задержкой через другие антенны.

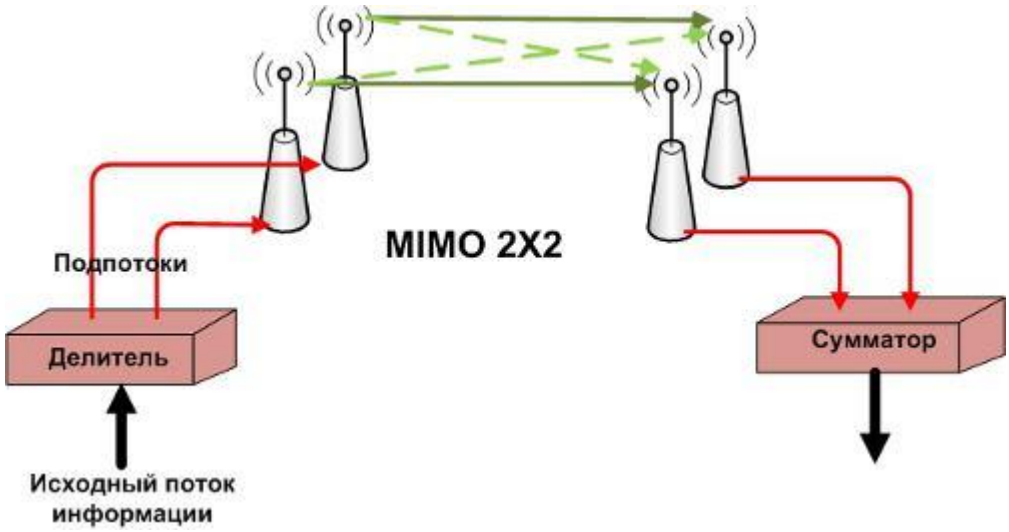


Рисунок 4.40 – Принцип организации технологии MIMO

В системе с технологией MIMO может возникнуть необычное явление, которое заключается в том, что скорость передачи данных в системе MIMO может снизиться в случае появления прямой видимости между источником и приемником сигнала. Это обусловлено в первую очередь уменьшением выраженности искажений окружающего пространства, который маркирует каждый из сигналов. В результате на приемной стороне становится проблематичным разделить сигналы, и они начинают оказывать влияние друг на друга. Таким образом, чем выше качество радио соединения, тем меньше преимуществ можно получить от MIMO.

Применение MIMO

Технология MIMO в последнее десятилетие является одним из самых актуальных способов увеличения пропускной способности и емкости беспроводных систем связи. Рассмотрим некоторые примеры использования MIMO в различных системах связи.

Стандарт WiFi 802.11n – один из наиболее ярких примеров использования технологии MIMO. Согласно ему, он позволяет поддерживать скорость до 300 Мбит/сек. Причем предыдущий стандарт 802.11g позволял предоставлять лишь 50 Мбит/сек. Кроме увеличения скорости передачи данных, новый стандарт благодаря MIMO также позволяет обеспечить лучшие характеристики качества обслуживания в местах с низким уровнем сигнала. 802.11n используется не только в системах точка/многоточка (Point/Multipoint) – наиболее

привычной нише использования технологии WiFi для организации LAN (Local Area Network), но и для организации соединений типа точка/точка которые используются для организации магистральных каналов связи со скоростью несколько сотен Мбит/сек и позволяющих передавать данные на десятки километров (до 50 км).

Стандарт WiMAX также имеет два релиза, которые раскрывают новые возможности перед пользователями с помощью технологии MIMO. Первый – 802.16e – предоставляет услуги мобильного широкополосного доступа. Он позволяет передавать информацию со скоростью до 40 Мбит/сек в направлении от базовой станции к абонентскому оборудованию. Однако MIMO в 802.16e рассматривается как опция и используется в простейшей конфигурации – 2x2. В следующем релизе 802.16m MIMO рассматривается как обязательная технология, с возможной конфигурацией 4x4. В данном случае WiMAX уже можно отнести к сотовым системам связи, а именно четвертому их поколению (за счет высокой скорости передачи данных), т.к. обладает рядом присущих сотовым сетям признаков: роуминг, хэндовер, голосовые соединения. В случае мобильного использования, теоретически, может быть достигнута скорость 100 Мбит/сек. В фиксированном исполнении скорость может достигать 1 Гбит/сек.

Наибольший интерес представляет использование технологии MIMO в системах сотовой связи. Данная технология находит свое применение, начиная с третьего поколения систем сотовой связи. Например, в стандарте UMTS, в Rel. 6 она используется совместно с технологией HSPA с поддержкой скоростей до 20 Мбит/сек, а в Rel. 7 – с HSPA+, где скорости передачи данных достигают 40 Мбит/сек. Однако в системах 3G MIMO так и не нашла широкого применения.

Системы 4G, а именно LTE, также предусматривают использование MIMO в конфигурации до 8x8. Это в теории может дать возможность передавать данные от базовой станции к абоненту свыше 300 Мбит/сек. Также важным положительным моментом является устойчивое качество соединения даже на краю соты. При этом даже на значительном удалении от базовой станции, или при нахождении в глухом помещении будет наблюдаться лишь незначительное снижение скорости передачи данных.

Таким образом, технология MIMO находит применение практически во всех системах беспроводной передачи данных. Причем потенциал ее не исчерпан. Уже сейчас разрабатываются новые варианты конфигурации антенн, вплоть до 64x64 MIMO. Это в будущем позволит добиться еще больших скоростей передачи данных, емкости сети и спектральной эффективности.

1. Задание на лабораторную работу

Для исследования OFDM:

1. Изменяя вероятность нуля (probability of a zero) от 0 до 1 с шагом 0,1 снять зависимость частоты ошибок от вероятности нуля. Построить график.
2. Изменяя отношение сигнал/шум от 0 дБ до 20 дБ с шагом 4 снять зависимость частоты ошибок от отношения сигнал/шум.
3. Снять созвездие «IFFT + white noise on amplitude Constellation» для значений отношения сигнал/шум 5 дБ, 10 дБ, 20 дБ.

Схема исследования OFDM представлена на рисунке 3.1:

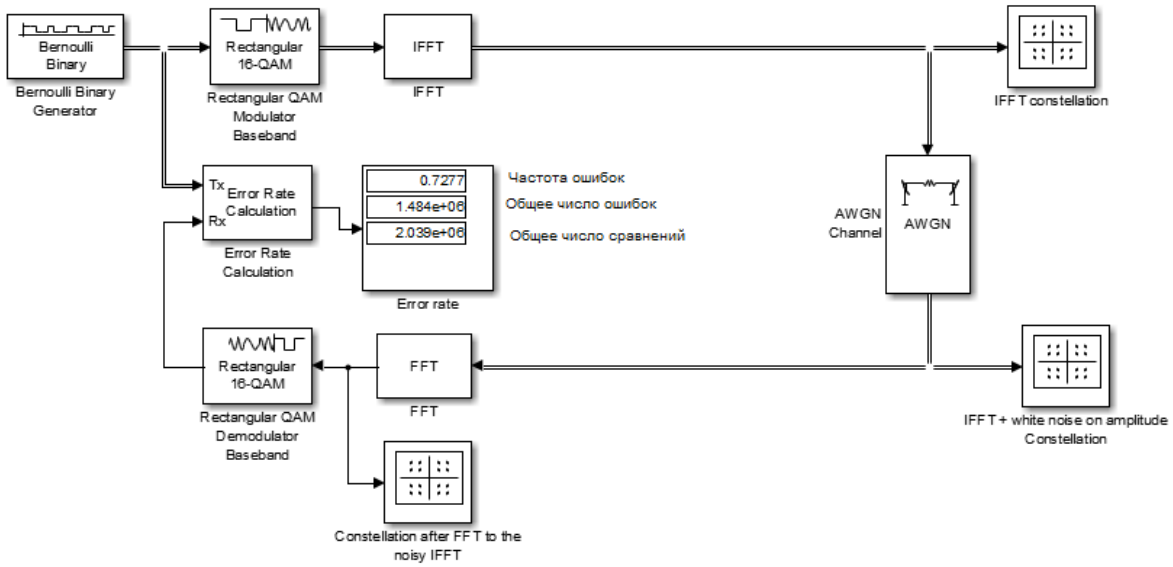


Рисунок 4.41 – Схема исследования системы OFDM.

В работе использованы следующие блоки:

1) Bernoulli Binary Generator – генератор ПСП.
2) QAM – модулятор.
3) Блок обратного быстрого преобразования Фурье.
4) Блок AWGN (канал с шумом).
5) Блок QAM-демодулятора.
6) Блок подсчёта ошибок.

Настройки блоков представлены на рисунках 3.2 – 3.7.

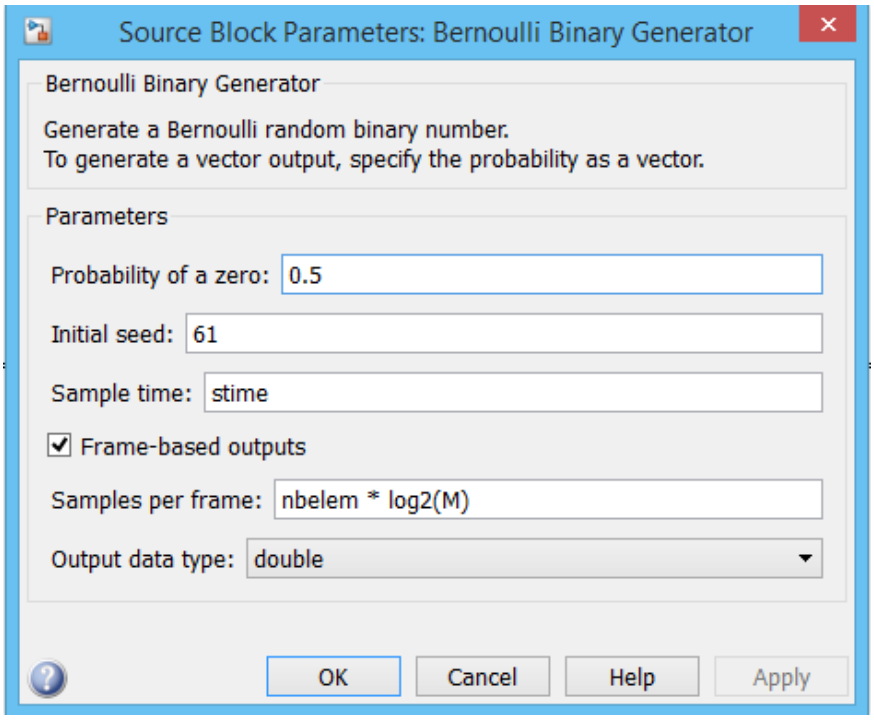


Рисунок 4.42 – Настройки блока генератора ПСП.

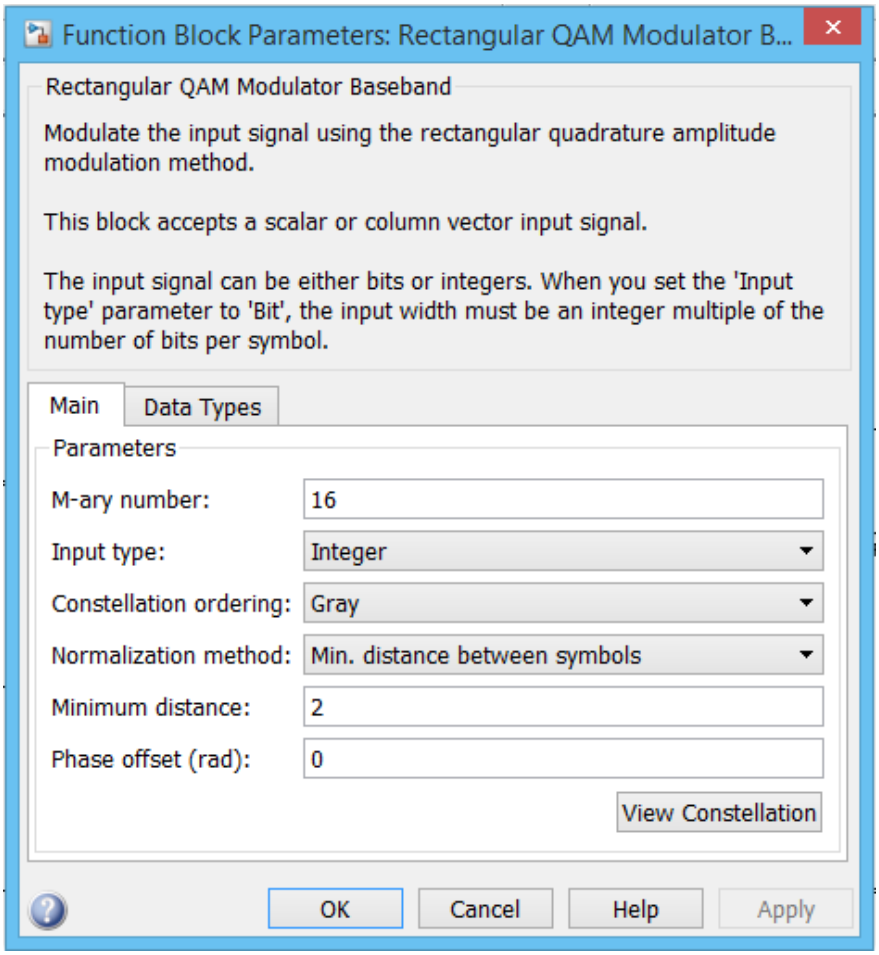


Рисунок 4.43 – Настройки QAM – модулятора.

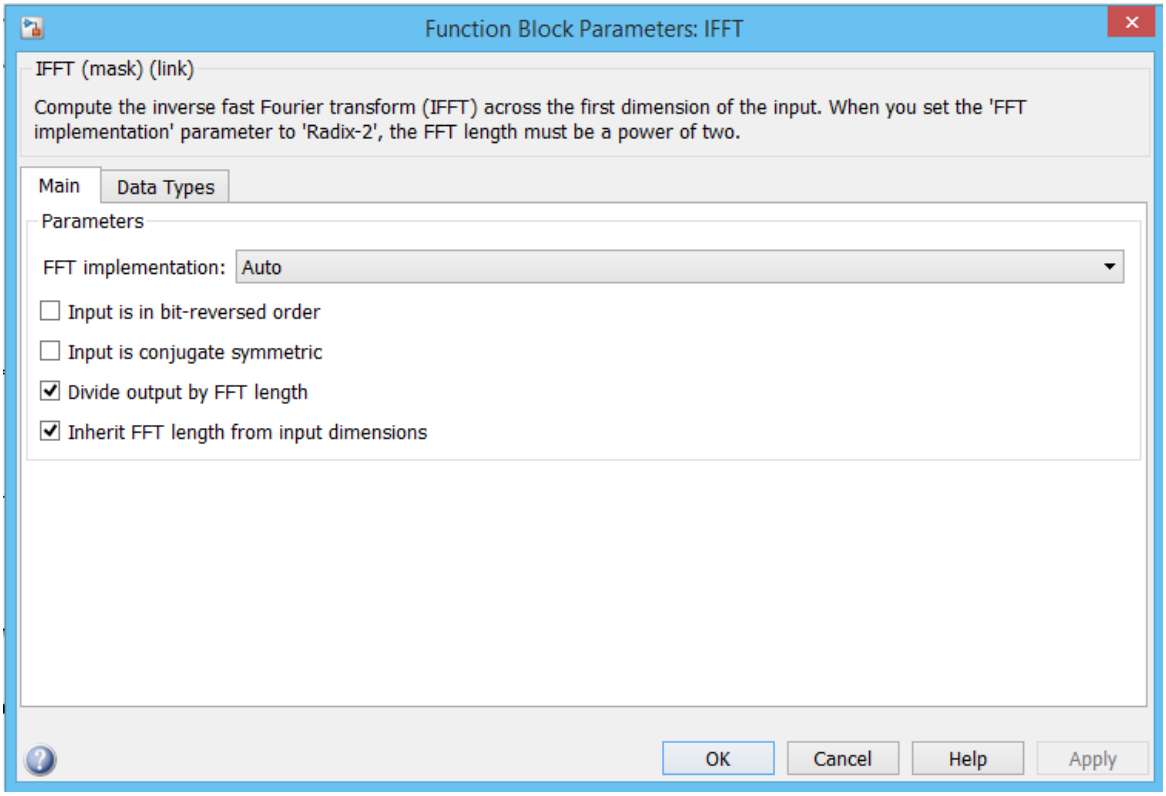


Рисунок 4.44 – Настройки блока преобразования Фурье.

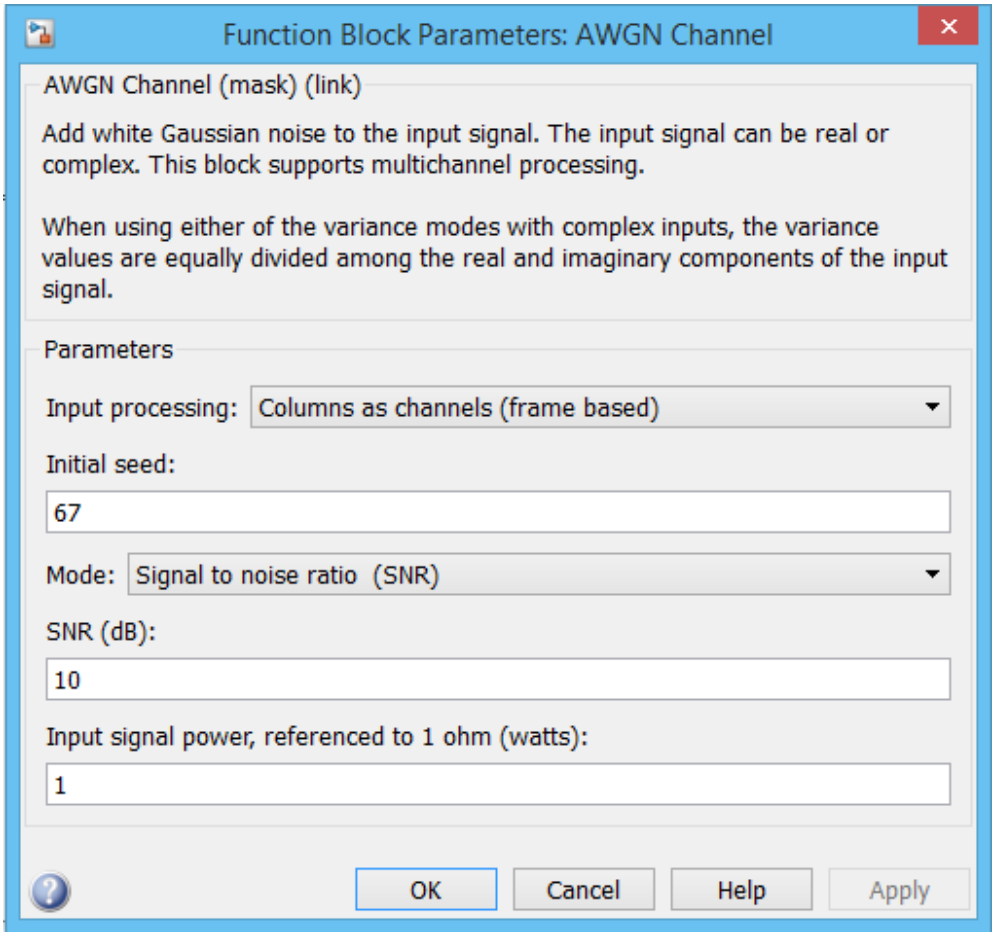


Рисунок 4.45 – Настройки блока канала с шумами.

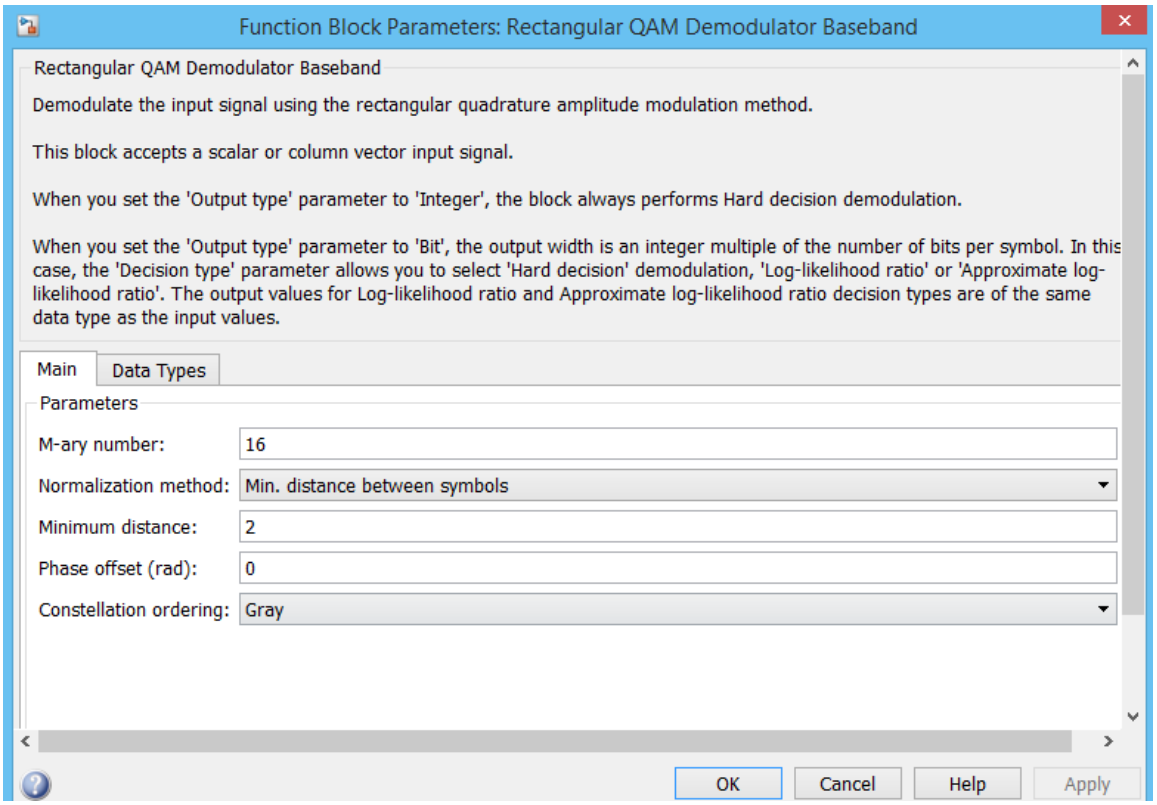


Рисунок 4.46 – Настройки QAM-демодулятора.

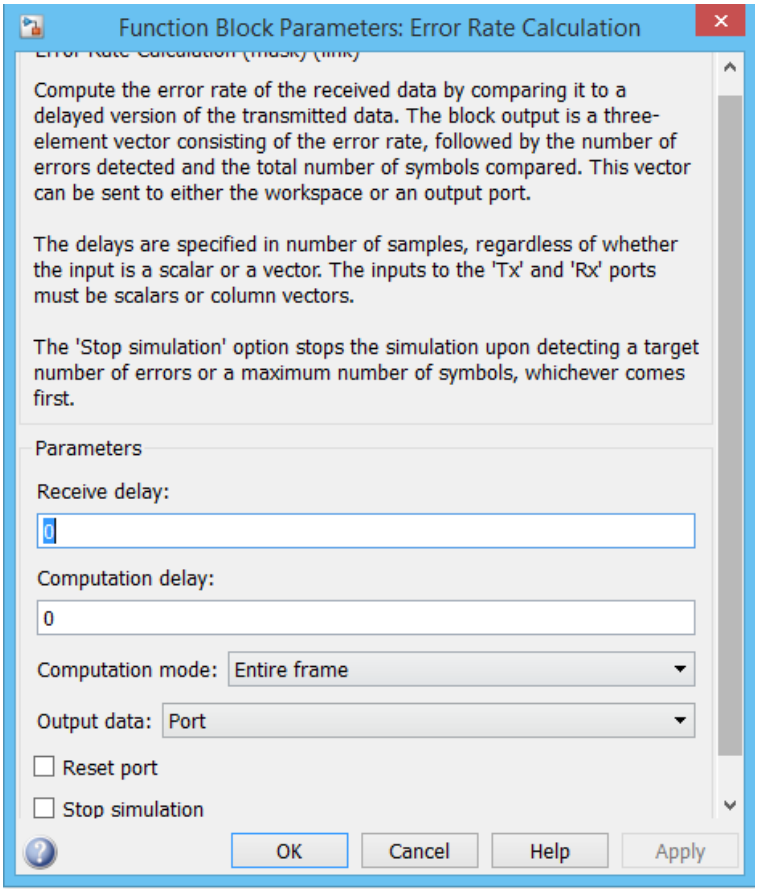


Рисунок 4.47 – Настройки блока подсчёта ошибок.

Для построения зависимости частоты ошибок от вероятности нуля нужно в генераторе ПСП изменять значение «probability of a zero» от 0 до 1.

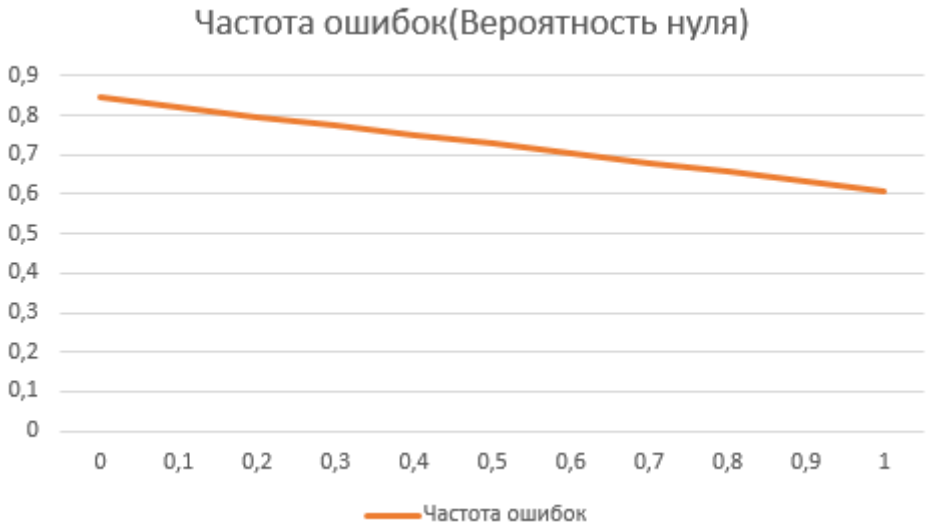


Рисунок 4.48 – Зависимость частоты ошибок от вероятности нуля.

Для построения зависимости частоты ошибок от отношения сигнал/шум нужно в блока канала с шумами изменять значение «probability of a zero» от 0 дБ до 20 дБ.

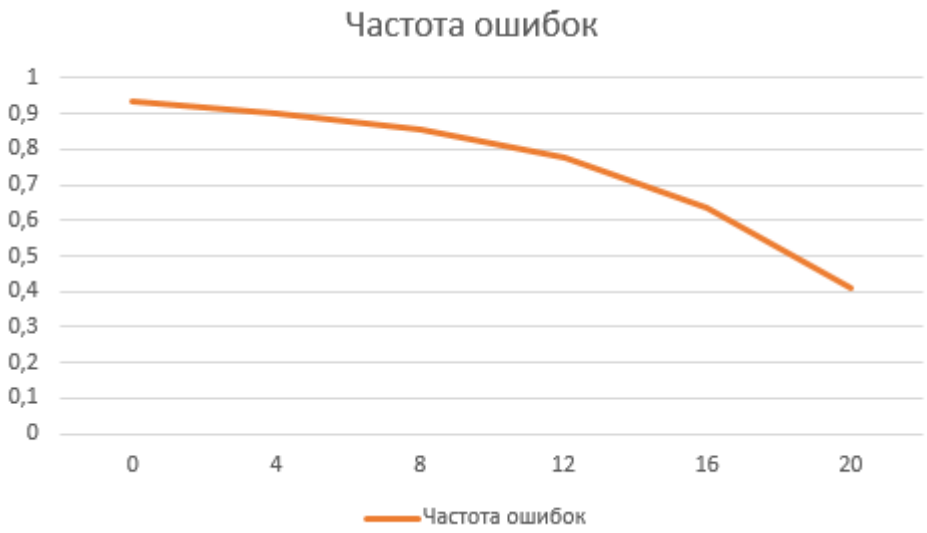


Рисунок 4.49 – Зависимость частоты ошибок от отношения сигнал/шум.

На рисунках ниже представлены созвездия для различных значений отношения сигнал/шум:

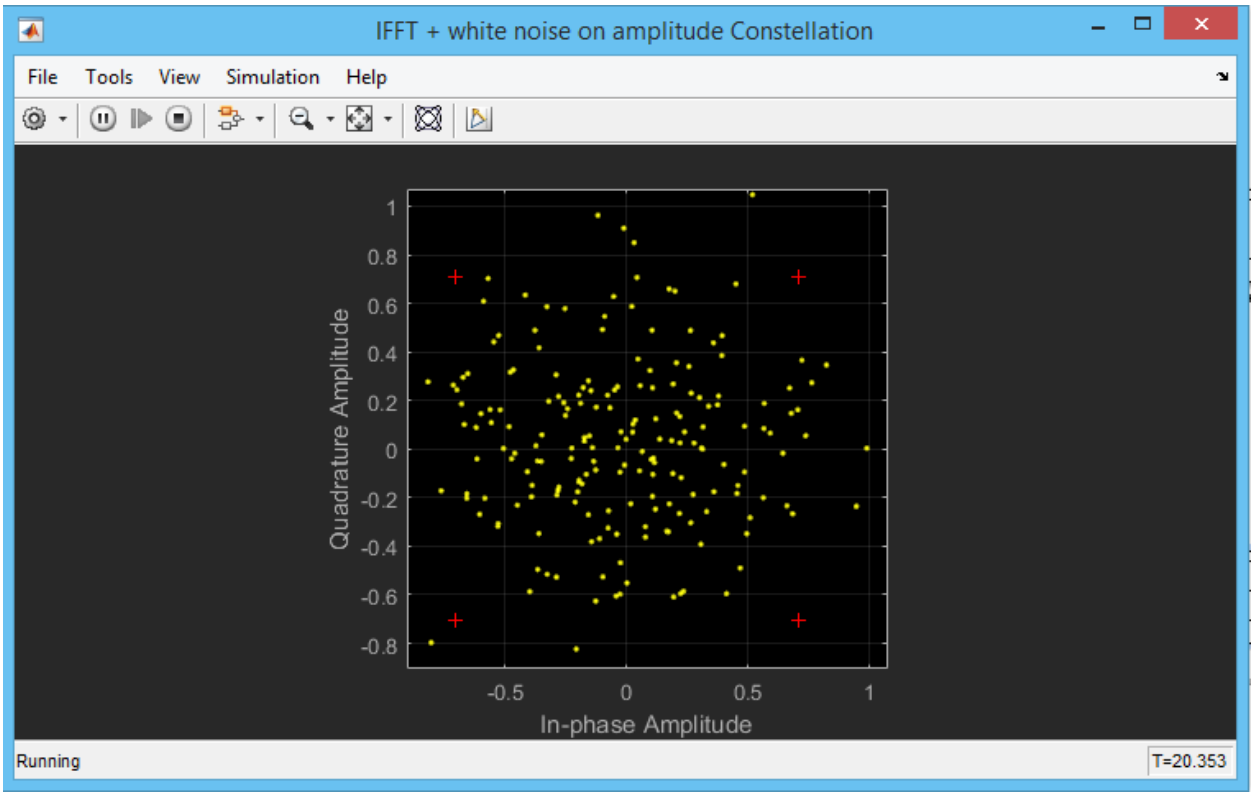


Рисунок 4.50 – Созвездие для значения отношения сигнал/шум 5 дБ.

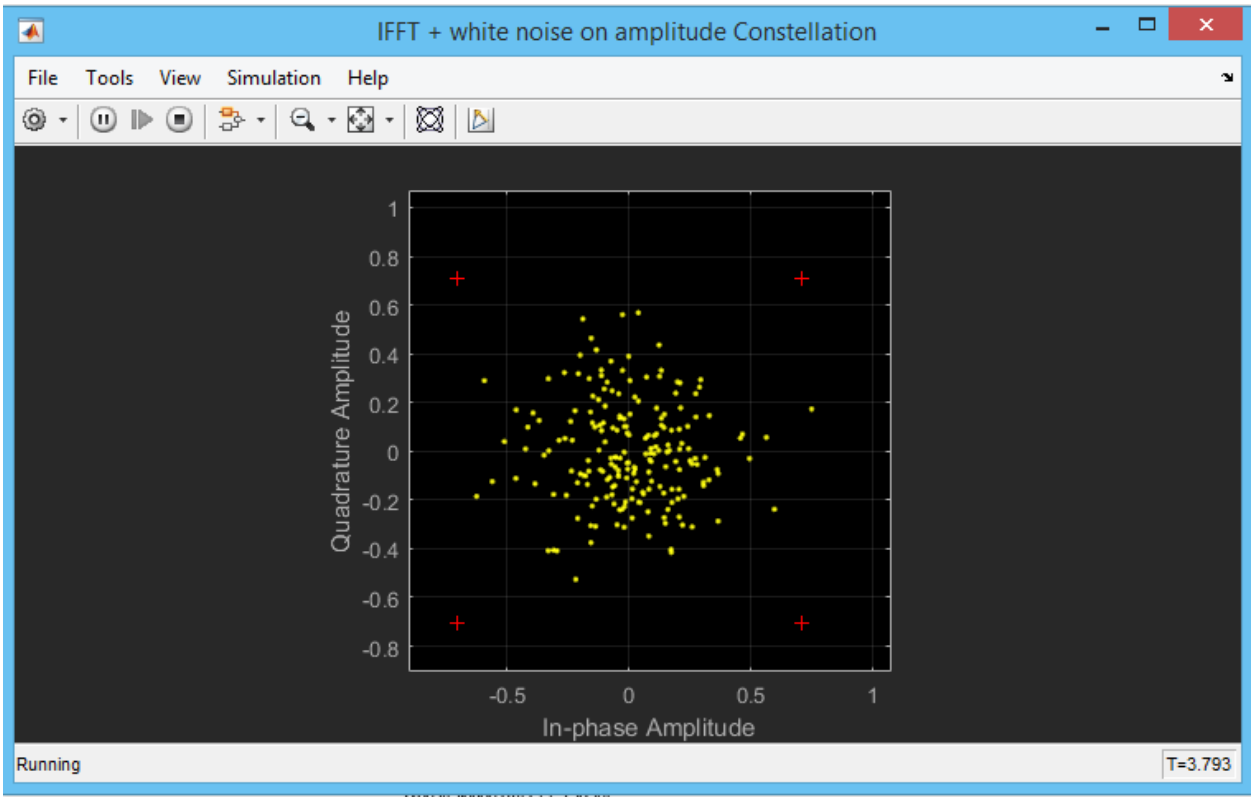


Рисунок 4.51 - Созвездие для значения отношения сигнал/шум 10 дБ.

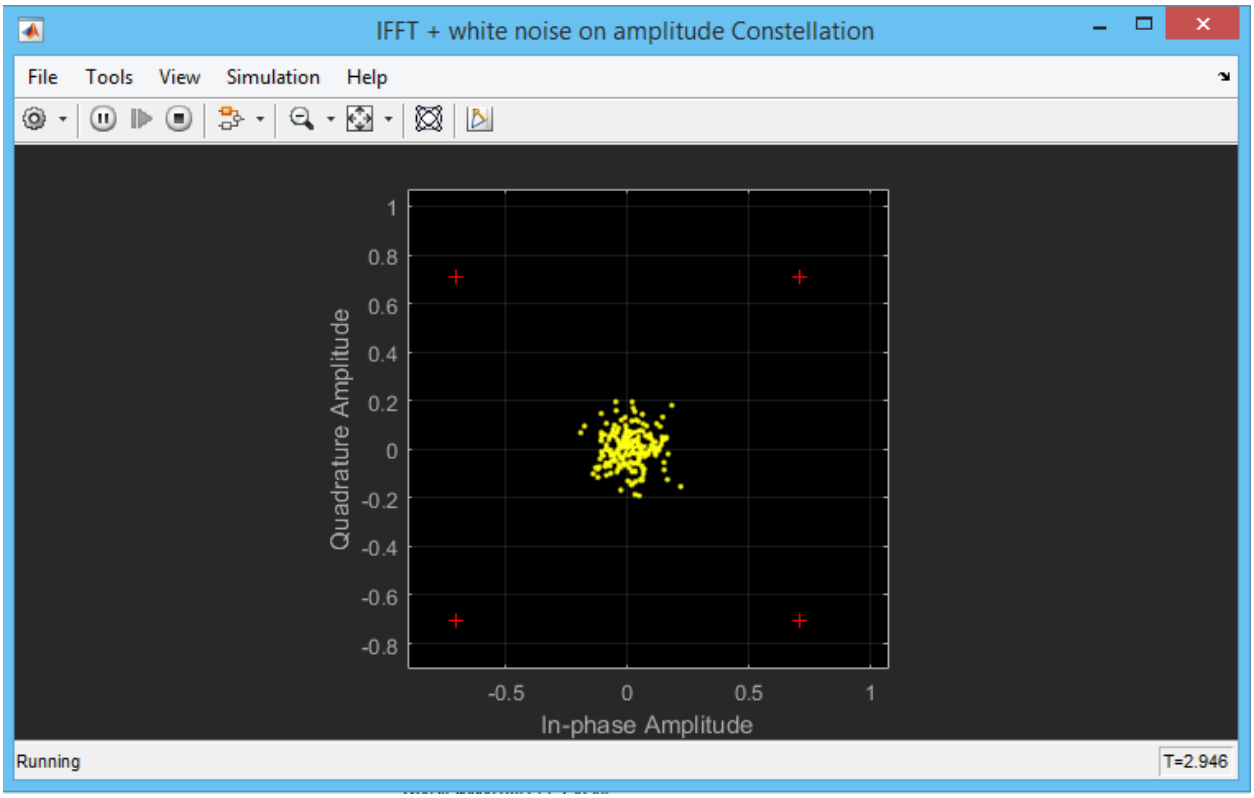


Рисунок 4.52 - Созвездие для значения отношения сигнал/шум 20 дБ.

Для исследования MIMO:

1. Изменяя SNR, dB (signal noise rate) в блоке Model Parameters от 0 до 30 с шагом 5, снять зависимость FER (частоты ошибок в кадре) от SNR. Построить график для адаптивного случая MIMO. Зафиксировать характеристики сигнала из блока Signal Visualization (двойной щелчок, рисунок 3.16).

Справка. В адаптивном случае система автоматически подстраивается под условия в канале под заданное в блоке Model Parameters значение FER. Для включения адаптивного случая в блоке Adaptive Control Panel нужно выставить переключатели в соответствующие положения.

2. Перевести переключатели в блоке Adaptive Control Panel в положения, соответствующие ручному вводу количества антенн (рисунок 3.15).

Изменяя SNR, dB (signal noise rate) в блоке Model Parameters от 0 до 30 с шагом 5, снять зависимость FER (частоты ошибок в кадре) от SNR. Построить график. Зафиксировать характеристики сигнала из блока Signal Visualization.

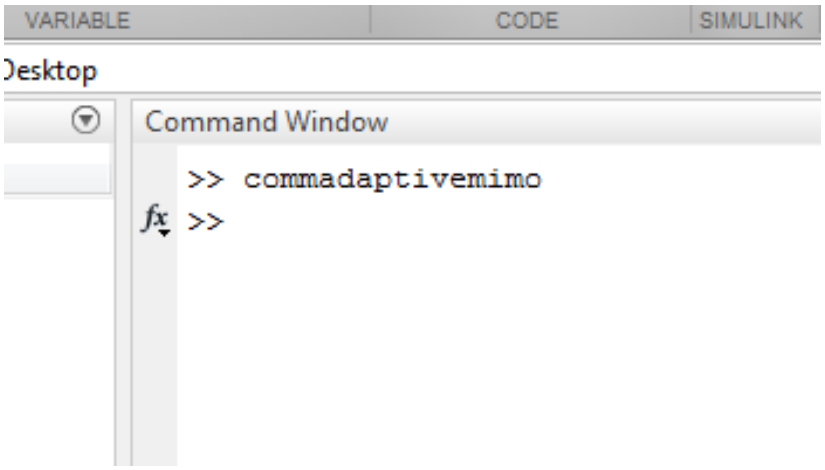


Рисунок 4.53 – Команда для создания схемы исследования MIMO.

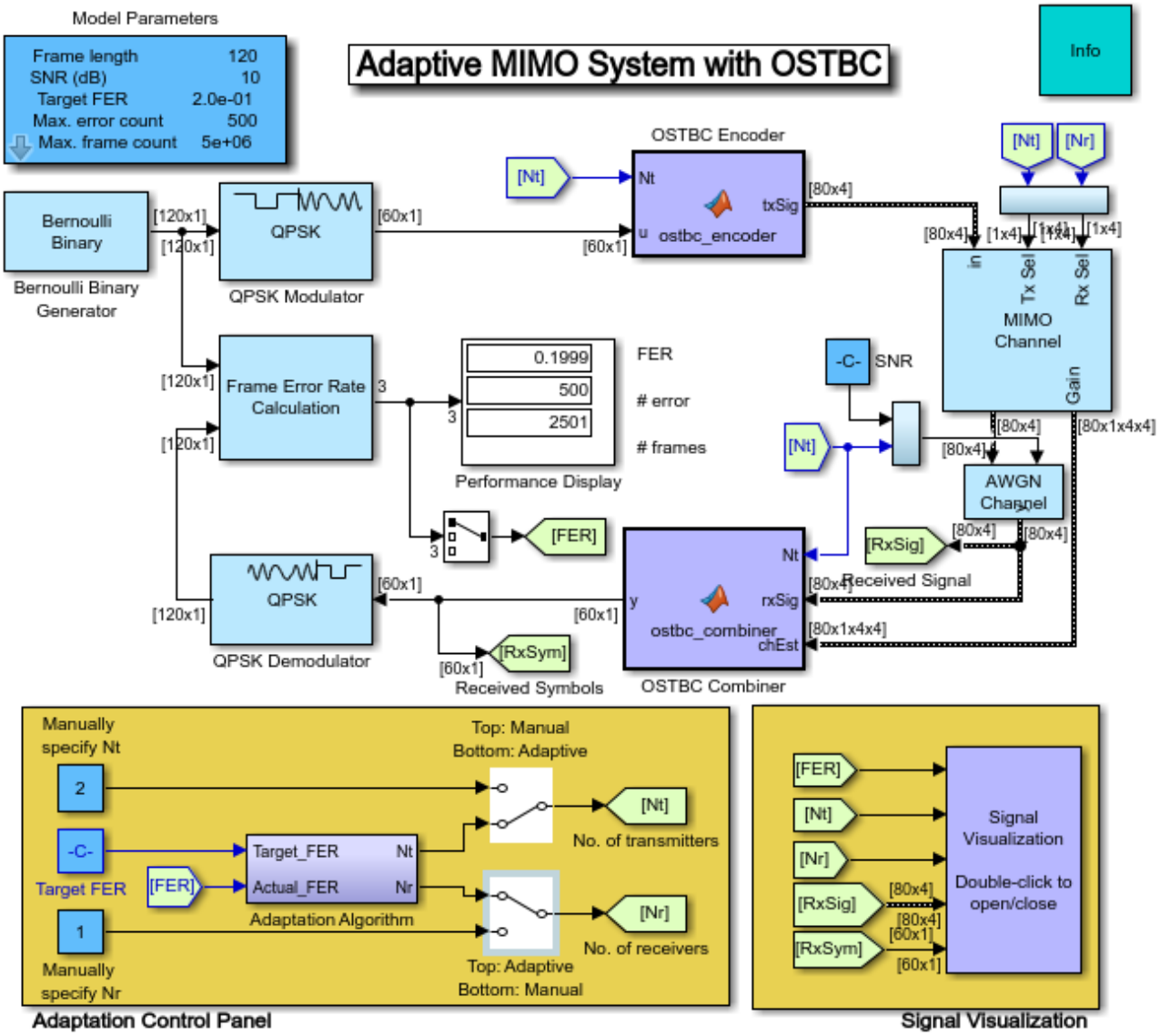


Рисунок 4.54 – Схема для исследования MIMO.

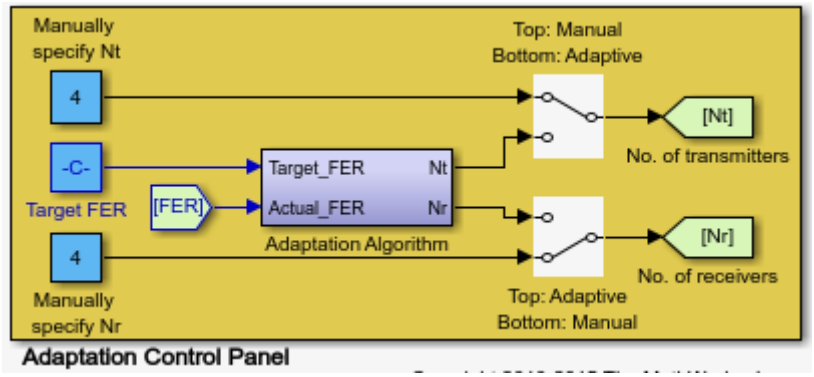


Рисунок 4.55 – Настройка блока Adaptation Control Panel для исследования случая MIMO 4x4.

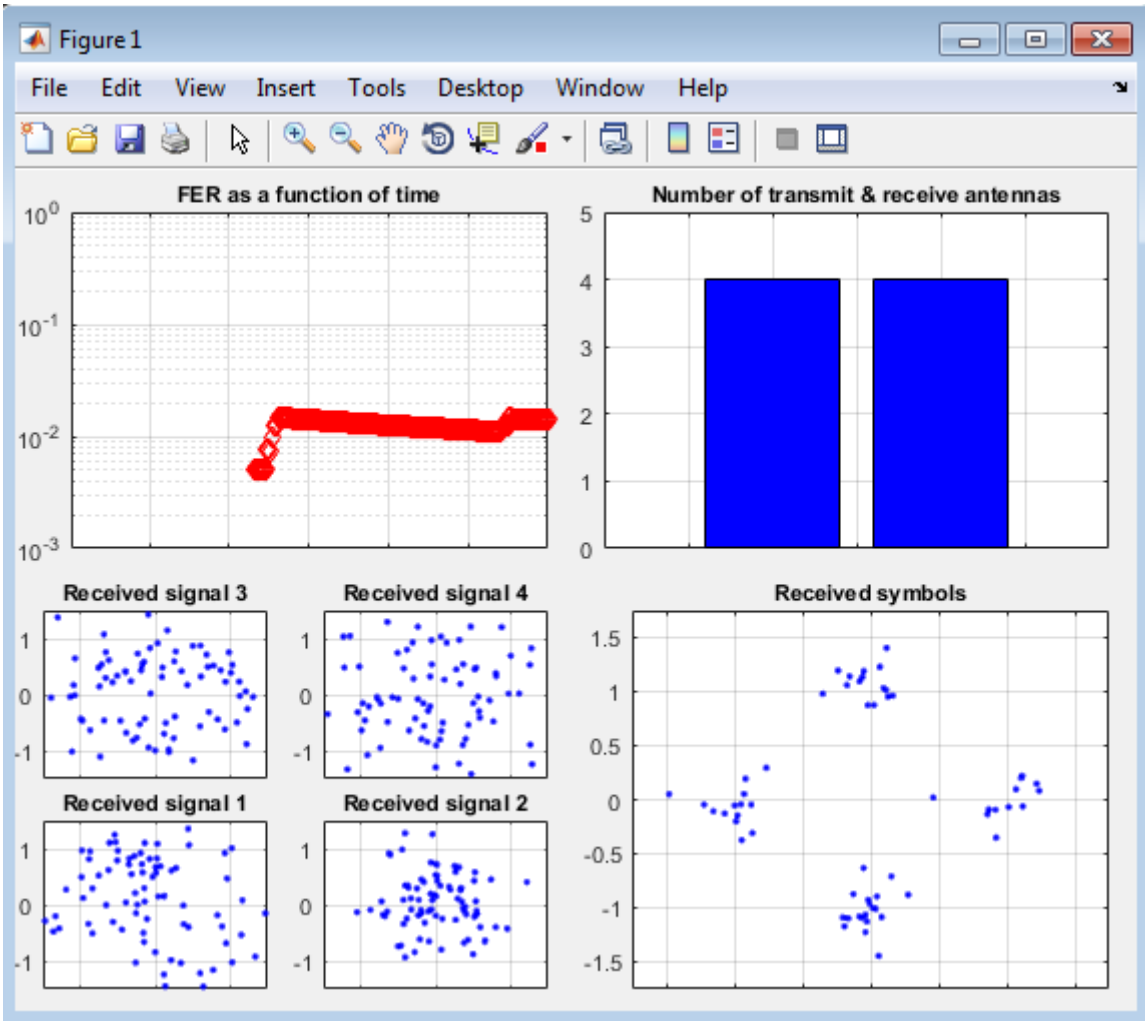


Рисунок 4.56 – Пример работы блока Signal Visualization.

ЛИТЕРАТУРА

1. Банкет В.Л. Помехоустойчивое кодирование в телекоммуникационных системах: учебн. пособие. - Одесса: ОНАС им А.С. Попова, 2011. - 104 с.ё

2. Зюко А.Г., Фалько А.И., Панфилов И.П., Банкет В.Л., Иващенко П.В. Помехоустойчивость и эффективность систем передачи информации. М.:Радио и связь. 1985.

3. Методы повышения энергетической и спектральной эффективности цифровой радиосвязи: учеб. пособие / В. А. Варгаузин, И. А. Цикин. — СПб.: БХВ-Петербург, 2013. — 352 с.

4. Банкет В.Л. Сигнально-кодовые конструкции в телекоммуникационных системах. - Одесса: Фешкс, 2009. - 180 с.

5. Мелихов С.В. Аналоговое и цифровое радиовещание: Учебное пособие. Издание второе, исправленное. - Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2012. – 233 с.

6. Голиков А.М., Уваровский В.Д. Исследование многоуровневых методов модуляции сигналов, используемых в космических системах связи, на базе аппаратуры и ПО labVIEW 2010. Методические указания по лабораторным работам – Томск: Том. гос. ун-т систем управления и радиоэлектроники, 2011. – 50 с.

7. Галкин В.А. Цифровая мобильная радиосвязь. Учебное пособие для вузов. – М.: Горячая линия-Телеком, 2007. – 432 с..

8. Федосов В. П., Нестеренко А. К. Цифровая обработка сигналов в LabVIEW: учеб. пособие / под ред. В. П. Федосова. – М.: ДМК Пресс, 2007. – 456 с.

9. Теория и техника передачи информации : учебное пособие /Ю. П. Акулиничев, А. С. Бернгардт. — Томск: Эль Контент, 2012. — 210 с.

10. Скляр Б. Цифровая связь. — М.: Издательский дом Вильямс. 2003 — 1104с

11. Феер К.: Беспроводная цифровая связь. М.: Радио и связь, 2000. - 520 с.

12. Крейнделин В.Б., Колесников А.В. Оценивание параметров канала в системах связи с ортогональным частотным мультиплексированием. Учебное пособие / МТУСИ.-М., 2010. -29 с.

13. Д. Ватолин, М. Смирнов «Методы сжатия данных: Сжатие изображений» // http://www.compression.ru/book/part2/part2__3.htm

14. С. Уэлстид. “Фракталы и вейвлеты для сжатия изображений в действии”. Москва. “Издательство ТРИУМФ” 2003. 360 .

15. https://sites.google.com/site/szatieinformacii/lekcii/tema13

16. Дворкович В.П., Дворкович А.В. Цифровые видеоинформационные системы (теория и практика) Москва: техносфера, 2012. – 1008 с.