

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

А. К. Ермаков, Н. В. Поваренкин

ЦИФРОВЫЕ СИСТЕМЫ
ОБРАБОТКИ ИНФОРМАЦИИ.
МЕТОДЫ ЦИФРОВОЙ ОБРАБОТКИ
Учебное пособие



Санкт-Петербург
2023

FreeBusta is knowledge
without borders!



УДК 004(075)
ББК 32.97я73
Е72

Рецензенты:
кандидат педагогических наук, доцент *А. В. Федоров*;
доктор технических наук, профессор *А. П. Лось*

Утверждено
редакционно-издательским советом университета
в качестве учебного пособия

Протокол № 5 от 3 октября 2022 г.

Ермаков, А. К.

Е72 Цифровые системы обработки информации. Методы цифровой обработки: учеб. пособие / А. К. Ермаков, Н. В. Поваренкин. – СПб.: ГУАП, 2023. – 110 с.

ISBN 978-5-8088-1836-1

Издание принадлежит циклу учебных пособий по дисциплинам, связанным с изучением систем цифровой связи, и содержит сведения, касающиеся обработки, передачи и приема сигналов с целью обмена информацией.

Предназначено для студентов бакалавриата, обучающихся по направлениям подготовки 11.03.01 «Радиотехника», 11.03.02 «Инфокоммуникационные технологии и системы связи», 12.03.05 «Лазерная техника и лазерные технологии», 03.03.03 «Радиофизика», а также для студентов специалитета по специальностям 11.05.01 «Радиоэлектронные системы и комплексы». Будет полезно студентам других направлений и специальностей, интересующимся практической реализацией устройств цифровой связи.

УДК 004(075)
ББК 32.97я73

ISBN 978-5-8088-1836-1

© Санкт-Петербургский государственный
университет аэрокосмического
приборостроения, 2023

FreeBusta is knowledge
without borders!



СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

4G – стандарт беспроводной широкополосной связи четвертого поколения

5G NR– 5G New Radio – стандарт связи пятого поколения

ARQ – Automatic repeat Request – автоматический запрос повторной передачи

AWGN – Additive White Gaussian Noise – аддитивный белый гауссов шум

BER – Bit Error Ratio – частота битовых ошибок

BPSK – Binary Phase Shift Keying – двоичная фазовая манипуляция

CDMA – Code-Division Multiple Access – множественный доступ с кодовым разделением

DSSS – Direct Sequence Spread Spectrum – метод прямой последовательности для расширения спектра

FDD – Frequenct Division Duplex – частотное разделение каналов

FDMA – Frequency Division Multiple Access – множественный доступ с разделением каналов по частоте

FEC – Forward Error Correction – прямая коррекция ошибок

FSK – Frequency Shift Keying – частотная манипуляция

GMSK – Gaussian Minimum Shift Keying – вид частотной манипуляции

GSM – the Global System for Mobile – глобальный стандарт цифровой связи второго поколения

LTE – Long-Term Evolution – стандарт беспроводной передачи данных для абонентских терминалов

LTI – Learning Tools Interoperability – стандартный протокол, описывающий взаимодействие учебных платформ между собой

NRZ – Non Return to Zero – код без возвращения к нулю

OFDM – Orthogonal Frequency-Division Multiplexing – множественный доступ с ортогональным частотным разделением каналов

PAM4 – 4 Pulse Amplitude Modulation – четырехпозиционная импульсно-амплитудная модуляция

PCM – Pulse-code modulation – импульсно-кодовая модуляция

QAM – Quadrature Amplitude Modulation – квадратурно-амплитудная модуляция



QPSK – Quadrature Phase Shift Keying – квадратурная фазовая манипуляция

SDMA – Spatial Division Access – множественный доступ с пространственным разделением каналов

SER – Symbol Error Ratio – частота символьных ошибок

SNR – Signal-to-Noise Ratio – отношение сигнал/шум

TDD – Time Division Duplex – разделение каналов по времени

TDMA – Time Division Multiple Access – множественный доступ с разделением по времени

WDMA – Wavelength Division Access – множественный доступ с разделением по длине волны
АБГШ – аддитивный белый гауссов шум

АМн – амплитудная манипуляция

АЦП – аналого-цифровой преобразователь

ГОСТ – государственный стандарт

ФМ – фазовая модуляция

ЦАП – цифро-аналоговый преобразователь

ЧМн – частотная манипуляция



ВВЕДЕНИЕ

«Цифровые системы обработки информации» – это устройства, которые обрабатывают цифровые сообщения, представляющие элементы конечного множества – символы, с целью получения и/или анализа исходного информационного сообщения.

Особенностью цифровых систем, в отличие от аналоговых, является способность передавать символы и биты с большей скоростью и дальностью, в том числе за счет введения помехоустойчивого кодирования, цифровых методов модуляции и множественного доступа.

В настоящем учебном пособии рассмотрены основные понятия систем обработки информации, такие как цифровое сообщение, цифровой, аналоговый и дискретный сигналы, бит, символ, скорость передачи информации, спектральная эффективность, битовая и символьная ошибки, пропускная способность и др.

Также были рассмотрены некоторые из основных элементов, участвующих в системе цифрового обмена информацией, такие как, например, канальное кодирование, модуляция, демодуляция, множественный доступ и т. д.

С точки зрения канального (помехоустойчивого кодирования) были рассмотрены основные классификационные признаки, а также наиболее используемые методы кодирования: блочные и сверточные. При рассмотрении каждого из методов были затронуты все особенности кодирования и декодирования, приведены примеры.

Немаловажным является то, что в учебном пособии присутствует практическая часть, то есть помимо текстовых примеров приведены конкретные листинги программ для среды Matlab. Это было сделано в тех пунктах, где затронута тема модуляции сигнала с целью его передачи. В качестве примеров рассмотрены основные сигнальные созвездия, в числе которых QAM, QPSK, BPSK и другие виды модуляций.



1. ОБЩИЕ ПОНЯТИЯ И ПРИНЦИПЫ ЦИФРОВЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ

1.1. Основные термины цифровых систем обработки информации

Очевидно, что понятие информации занимает центральное место в цифровых системах обработки информации. Но информация – это многозначное слово, подразумевающее семантические и философские понятия, которые не поддаются точному определению. Однако с точки зрения цифровых систем передачи, приема и обработки информации последняя понимается как физическое ее проявление. Что бы не требовалось от сообщения, цель цифровой системы связи состоит в том, чтобы воспроизвести в пункте назначения приемлемую копию исходного сообщения. Для изучения цифровых систем обработки информации в первую очередь необходимо определиться с основными понятиями.

Существует множество видов источников информации, включая машины, а также людей, в которых сообщения проявляются в различных формах. Тем не менее мы можем выделить два основных вида сообщений: аналоговые и цифровые. Это различие в свою очередь определяет критерий успешного обмена информацией или данными.

Аналоговое сообщение – это физическая величина, которая изменяется со временем, обычно плавно и непрерывно. Примерами аналоговых сообщений являются акустическое давление, создаваемое при разговоре, угловое положение авиационного гироскопа или интенсивность света в некоторой точке телевизионного изображения. Поскольку информация кодируется в параметрах аналогового сигнала, изменяющегося во времени, аналоговая система связи должна передавать его с определенной степенью точности соблюдения этих параметров.

Цифровое сообщение – это упорядоченная последовательность символов, выбранных из конечного набора дискретных элементов. Примерами цифровых сообщений являются буквы, напечатанные на этой странице, список почасовых показаний температуры или клавиши, которые вы нажимаете на клавиатуре компьютера. Поскольку информация хранится в дискретных символах, цифровая система связи должна передавать эти символы с заданной степенью точности за заданный промежуток времени.



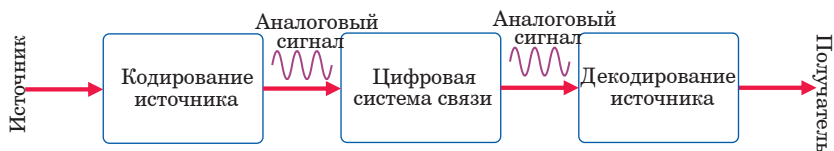


Рис. 1.1. Преобразование исходного сообщения в электрический сигнал

Будь то аналоговые или цифровые, немногие источники исходного сообщения по своей сути являются электрическими. Следовательно, большинство систем связи имеют входные и выходные преобразователи, как показано на рис. 1.1. Первый блок первичного кодирования исходного сообщения преобразовывает информацию в электрический сигнал (в напряжение или ток), в то время как блок декодирования источника в пункте назначения преобразовывает электрический сигнал в желаемую форму получения информации. Например, преобразователями в системе голосовой связи могут быть микрофон на входе и громкоговоритель на выходе.

Цифровой сигнал описывается уровнем напряжения или силой тока, сигнал (импульс (рис. 1.2а) – для низкочастотной передачи или синусоида (рис. 1.2б) – для полосовой передачи) и представляет собой цифровой символ. Для идентификации цифрового сигнала

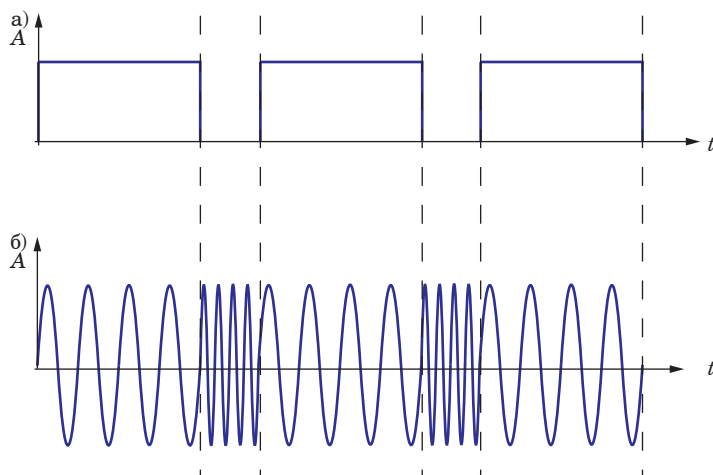


Рис. 1.2. Пояснения к определению цифрового сигнала:
а – импульсный сигнал; б – цифровой сигнал для полосовой передачи



ла как один из символов конечного алфавита используются его характеристики:

- амплитуда, длительность и положение для низкочастотной передачи;

- амплитуда, частота и фаза для высокочастотного сигнала.

На рис. 1.2б приведен пример полосового цифрового сигнала. Хотя сигнал является синусоидальным и, следовательно, имеет аналоговый вид, все же он именуется цифровым, поскольку кодирует цифровую информацию. На данном рисунке цифровое значение указывает определенную частоту передачи в течение каждого интервала времени T .

Стоит отметить, что понятия и физический смысл аналогового и дискретного сигналов, а также примеры реализации в среде Matlab будут раскрыты в 5 разделе, посвященном основам цифровой обработки сигналов. Здесь же отметим, что аналоговый сигнал – это сигнал, распространяющийся в пространстве и непрерывный в моменте времени, в котором он распространяется. Дискретный же сигнал – это цифровая копия аналогового сигнала, однако он дискретизирован по отсчетам и частота появления этих отсчетов может влиять на ошибку, которая может привести к тому что тот сигнал, который мы хотели получить, будет искажен, то есть, например, значение его фазы или частоты будут неверны.

В соответствии с Постановлением Правительства РФ от 31 октября 2009 года «Об утверждении Положения о единицах величин, допускаемых к применению в Российской Федерации» в области информационных технологий и связи минимальной единицей измерения количества информации является *бит* (рис. 1.3а). Бит может принимать только два значения – 0 (сигнала нет) или 1 (сигнал есть).

В цифровых системах обработки информации с целью увеличения скорости передачи информации стараются передавать информацию параллельно. Например, в цифровых устройствах по 8-разрядной шине данных можно передавать одновременно 8 бит информации. Совокупность переданных бит одновременно называется *символом* (рис. 1.3б), а временной интервал, через который символы следуют друг за другом, называется *длительностью передачи символа* и обозначается буквой T (рис. 1.3в). Количество бит в одном символе обозначается буквой m , а число возможных значений символа буквой M и определяется формулой

$$M = 2^m. \quad (1.1)$$



Так, к примеру, в случае с 4-разрядным цифровым устройством, где шина данных (символ m) равен 4 бит, количество возможных значений M будет равно 16.

Также в цифровых системах обработки информации принято говорить о скорости передачи информации, которая делится на:

– битовую, определяемую числом бит, переданных за единицу времени, – R , бит/с;

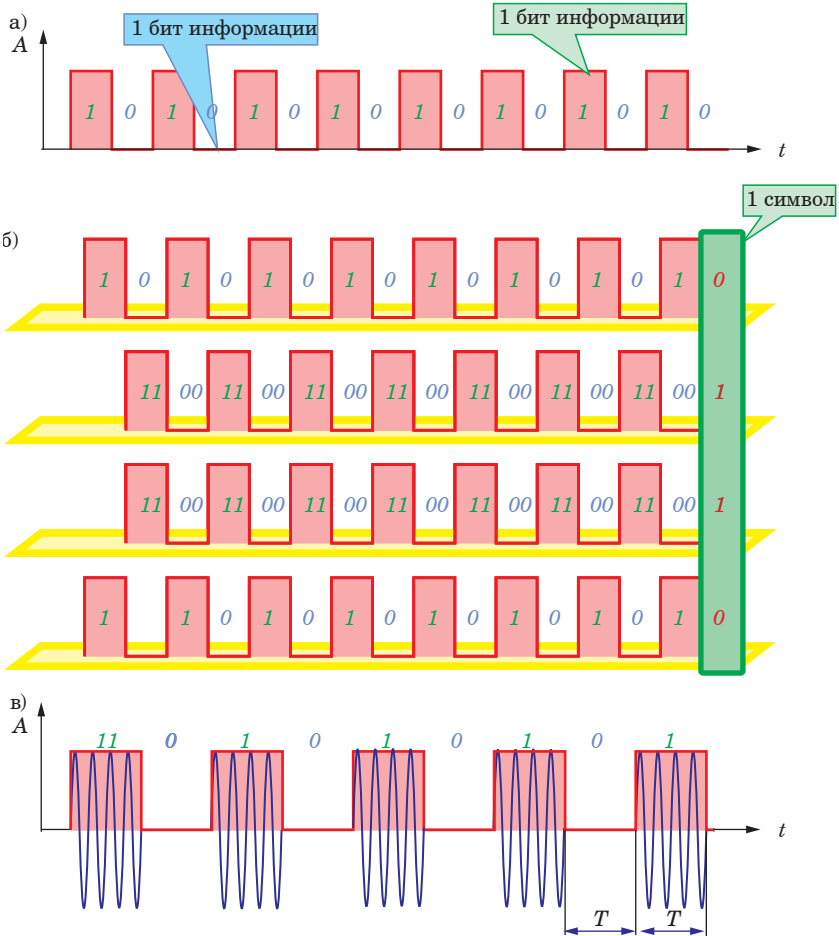


Рис. 1.3. Единицы измерения в цифровых системах обработки информации: а – бит; б – символ; в – длительность передачи символа



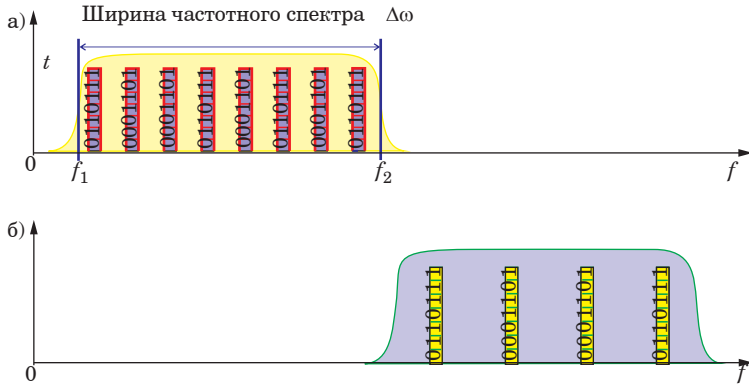


Рис. 1.4. Пояснения к понятиям «ширина частотного спектра» и «спектральная эффективность»:

- а – ширина частотного спектра с символом равным 8 бит;
- б – ширина частотного спектра с символом равным 4 бит

– *символьную*, определяемую числом символов, переданных в единицу времени, – $R_{\text{сим}} = R/m = 1/T$, сим/с.

Говоря о передаче цифровой информации с помощью аналогового сигнала, важными параметрами являются:

- *ширина частотного спектра сигнала* – $\Delta\omega$, Гц;
- *спектральная эффективность* – $R/\Delta\omega$, (бит/с)/Гц, которая показывает эффективность использования ширины частотного спектра с целью передачи цифровой информации. Как видно из рис. 1.4, в верхней его части спектр используется эффективнее, чем на нижней. Распределение энергии сигнала по частотам называется спектром сигнала (рис. 1.5).

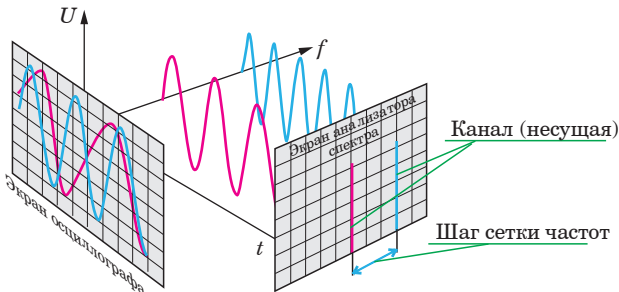


Рис. 1.5. Пояснения к понятию «спектр сигнала»



Вероятность того, что принятая цифровая информация не совпадает с переданной, называется вероятностью ошибки и подразделяется на две составляющие:

– *битовая ошибка* $P_b \in [0, 0,5]$ – параметр, определяющий вероятность несовпадения принятого бита переданному;

– *символьная ошибка* $P_{sym} \in [0, (M-1)/M]$ – параметр, определяющий вероятность несовпадения принятого символа переданному. В связи с тем, что символ – это совокупность (группа) бит, в случае символьной ошибки, вероятнее всего, исказятся не все биты, а лишь часть из них. Так, становится ясно, что вероятность возникновения битовой ошибки P_b не может превышать символьную ошибку $P_b \leq P_{sym}$. Исключением является только бинарная система, в которой 1 символ равен одному биту, тогда вероятности их ошибок совпадают $P_b = P_{sym}$.

Также вероятность ошибки в цифровых системах обработки информации называется частотой битовой ошибки *BER* (Bit Error Rate) и частотой символьной ошибки *SER* (Symbol Error Rate) соответственно, что с точки зрения физического смысла нагляднее характеризует этот параметр.

Еще один параметр, определяющий качество передачи информации, называется «*отношение сигнал/шум*». Для измерения отношения сигнал/шум в цифровых системах обработки информации может использоваться два подхода:

1) применительно к аналоговым системам основным является измерение отношения сигнал/шум по мощности сигналов – SNR, дБ (рис. 1.6). В связи с тем, что спектр шума является бесконечным,

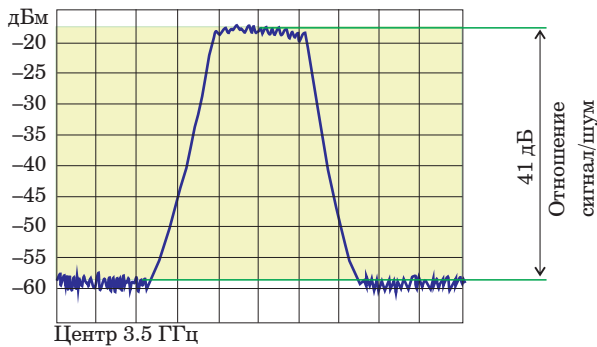


Рис. 1.6. Пояснение к определению измерения отношения сигнал/шум по мощности сигнала



необходимо определить, в какой полосе частот будет измеряться шум. Так, зачастую для измерения мощности шума выбирают такую полосу частот, которая совпадает с полосой частот, занимаемой полезным сигналом. Исходя из этого, отношение сигнал/шум, можно определить по формуле

$$SNR = \frac{P_s}{P_n} = \frac{P_s}{N_0 \Delta \omega}, \quad (1.2)$$

где P_s – мощность полезного сигнала; N_0 – спектральная плотность мощности (СПМ) белого шума; $\Delta \omega$ – ширина спектра полезного сигнала.

Если выразить мощность полезного сигнала и мощность шума из (1) в децибелах относительно 1 мВт, то выражение можно упростить до

$$SNR_{dB} = |P_{сигн} - P_{шума}|; \quad (1.3)$$

2) применительно же к цифровым системам используют понятие «отношение сигнал/шум на бит». Физически эти понятия схожи с тем исключением, что в данном случае отношение вычисляет путем взятия энергии сигнала, на которую приходится только 1 бит передаваемой информации, к спектральной плотности мощности белого шума

$$SNR_b = \frac{E_b}{N_0 \Delta \omega}, \quad (1.4)$$

где E_b – энергия сигнала, приходящегося на 1 бит информации.

Также можно представить отношение сигнал/шум на символ

$$SNR_b = \frac{E_m}{N_0 \Delta \omega}, \quad (1.5)$$

где E_m – энергия сигнала, приходящегося на 1 символ.

В связи с тем, что $P_s = E_b R$, видна связь между отношением сигнал/шум по мощности и отношением сигнал/шум на бит

$$SNR = \frac{P_s}{N_0 \Delta \omega} = \frac{E_b R}{N_0 \Delta \omega}. \quad (1.6)$$



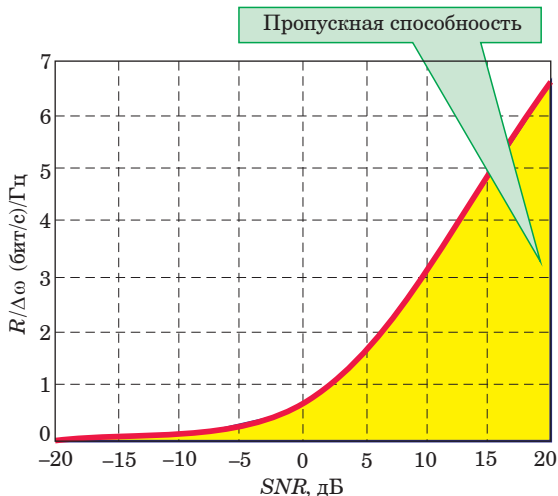


Рис. 1.7. Пояснение к понятию «пропускная способность»

Таким образом, становится понятно, что отношение сигнал/шум по мощности определяется произведением отношения энергии сигнала на бит к спектральной плотности мощности и спектральной эффективности системы.

Пропускная способность. Пропускная способность канала передачи информации показывает границу скорости передачи информации, с которой возможно обеспечить наименьшую вероятность символьной или битовой ошибки (для определенных методов кодирования информации). Измеряется пропускная способность в количестве битов или символов, передаваемых в единицу времени. На рис. 1.7 представлено пояснение к понятию «пропускная способность», где желтым цветом показана область допустимых значений.

1.2. Основные функциональные элементы цифровых систем обработки информации

Системы цифровой обработки информации предназначены для передачи информации через канал связи и приема информации, полученной через канал связи.

Решение указанных задач осуществляется с использованием множества преобразований, например, таких как ограничений по-





Рис. 1.8. Общая схема системы цифрового обмена информацией

лучаемых сигналов по ширине спектра. Обобщенная схема, позволяющая решать обозначенные задачи, представлена на рис. 1.8.

Для передачи информации от ее источника первым делом необходимо произвести начальное кодирование, которое направлено на проведение аналого-цифрового преобразования исходного сообщения.

Следующим шагом является так называемое *канальное кодирование*. Оно является неотъемлемой частью всех систем обмена цифровой информацией, так как вносит свои корректировки в сообщение с целью учета уровня шума и, например, изменчивости во времени. Общепринятым термином, которым называют канальное кодирование, является *помехоустойчивое кодирование*. При помехоустойчивом кодировании отправляемой информации в сигнал



закладывается код, необходимый для обеспечения на приемной стороне возможности обнаружения и исправления ошибок, появившихся в канале связи из-за влияния преднамеренных и непреднамеренных помех. Преднамеренные помехи вносятся специальными системами и (или) устройствами с целью ухудшения (уменьшения отношения сигнал\шум) или прекращения работы цифровой системы обмена информацией в штатном режиме. Непреднамеренным помехами могут являться погодные условия либо локальные особенности места расположения канала связи, например, горы, здания, лес и др.

Известно, что в радиоканале связи невозможно передавать цифровой сигнал. В связи с этим дальше сигнал с примененным к нему помехоустойчивым кодированием поступает на модулятор, где передаваемая информация кодируется в аналоговый сигнал, изменяя его параметры в соответствии с одним из методов модуляции радиосигнала. То есть, предположим, что нам нужно передать два информационных бита: 1 и 0. Примем, что «1» соответствует сигналу $s(t)$ с фазой φ равной 0, а «0» второму сигналу $s_1(t)$ с фазой φ равной $\pi/2$. Таким образом, мы преобразовали цифровую информацию в битах в два сигнала, которые мы передаем одновременно на разных частотах или разделенные по времени. Также в модуляторе происходит перенос частотного спектра в область более высоких частот.

Блоки *расширение спектра* и *множественный доступ* являются не обязательным атрибутом цифровой системы обмена информацией. Однако они позволяют достичь ряда преимуществ перед системами без них. Так, в современных стандартах связи, например, *5G New Radio* невозможно обойтись без таких методов расширения спектра, как, к примеру, DSSS, и методов множественного доступа, например, FDMA, который в свою очередь необходим для частотного разделения каналов, увеличивающего количество пользователей в рамках одного частотного диапазона (рис. 1.9).

Дальше, имея уже аналоговый сигнал, его можно передавать в радиоканал. Однако в системе, формирующей передаваемый сигнал, его мощность слишком низкая для передачи в воздушном пространстве, при такой мощности сигнал затухнет и не поступит на систему-получатель. Так, для передачи сигнала необходимо его усилить, а также определить направление, в которое будет направлено передаваемое сообщение. Отсюда возникает необходимость *передатчика*. Этот блок является аналоговой частью цифровой си-



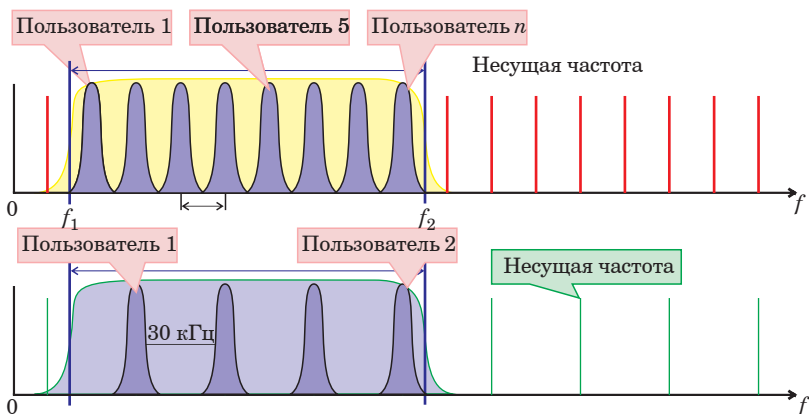


Рис. 1.9. Разделение пользователей радиоканалом по частоте

стемы передачи информации, содержит в себе усилитель радиосигнала и антенну.

Как уже было сказано ранее, при распространении сигнала в радиоканале, он подвергается воздействию преднамеренных и непреднамеренных помех. Таким образом, на вход приемника поступаем искаженный радиосигнал.

Для обеспечения приема и обработки принятого сигнала с целью получения информации в приемной части необходимо:

- максимум диаграммы направленности направить в противоположную сторону вектора направления максимума диаграммы направленности на передающей антенне;
- произвести фильтрацию сигналов и помех, находящихся вне спектрального диапазона, в котором была передана информация;
- усилить принятый сигнал;
- перенести его спектр в область низких частот для удобства работы с сигналом, поскольку высокочастотное оборудование является очень сложным для производства и, соответственно, дорогостоящим.

Таким образом, приемника по своему функциональному смыслу является зеркальным отражением передатчика.

В случае если использовались технологии множественного доступа и расширения спектра, в цифровую систему приема и обработки информации включают соответствующие блоки для выделения сигнала и сужения спектра.

Следующим шагом аналогово-модулированный сигнал подвергается демодуляции. В ходе демодуляции происходит выделение



огибающей радиосигнала, совпадающей с переданным цифровым сообщением.

В соответствии с пониманием о зеркальности построения приемной части цифровой системы обработки информации необходимо произвести канальное (помехоустойчивое) декодирование. В ходе него и благодаря корректирующим свойствам кода происходит обнаружение и исправление ошибок, вызванных из-за распространения сигнала в радиоканале. Затем сигнал после демодулятора поступает на последний блок, который является завершающим этапом цифровой системы приема и обработки информации и производит восстановление исходного сообщения.



2. КАНАЛ ПЕРЕДАЧИ ДАННЫХ

Канал передачи данных – это самый важный вопрос для любого вида системы связи. Эффективность канала связи зависит от уровня шума. Аддитивный белый гауссов шум исходит от многих естественных источников, таких как вибрация атомов в проводнике, шум линий электропередач, излучение от земли и других теплых объектов, а также от небесных источников, таких как Солнце.

2.1. Канал связи с аддитивным гауссовым шумом

Канал связи с аддитивным белым гауссовым шумом (АБГШ) AWGN является простейшей моделью канала и хорошо подходит для проводной связи. Этот канал является линейным и инвариантным по времени (LTI). Канал AWGN добавляет белый гауссов шум к сигналу, когда сигнал проходит через него. Амплитудно-частотная характеристика этого канала плоская, а фазовая характеристика линейна для всех частот. Модулированный сигнал проходит

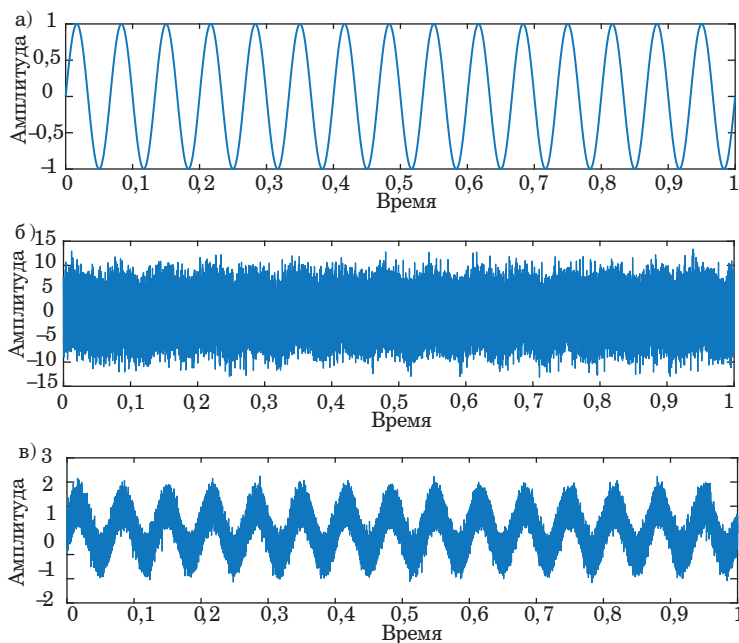


Рис. 2.1. Пояснение к выражению (2.1): а – сигнал без шума; б – сигнал с аддитивным гауссовым шумом при $SNR=$ минус 10 дБ; в – сигнал с аддитивным гауссовым шумом при $SNR=10$ дБ



через него без каких-либо потерь амплитуды и фазовых искажений, затухания не существует, единственное искажение вносится AWGN. Сигнал в канале с AWGN (рис. 2.1) определяется выражением

$$S(t)=s(t)+n(t), \quad (2.1)$$

где $s(t)$ – полезный сигнал; $n(t)$ – белый гауссов шум с двусторонней спектральной плотностью $N_0/2$.

Пример программного кода для среды Matlab, результатом работы которого является радиосигнал с аддитивным гауссовым шумом:

```
clc
clear all
close all
%Амплитуда сигнала
a=1;
%Частота сигнала
f=15;
%Частота дискретизации
fd=10000*f;
%Время наблюдения
t=0:1/(fd):1;
%Исходный сигнал
s=a*sin(2*pi*f*t);
%Сигнал с аддитивным гауссовым шумом, SNR = -10
s1=awgn(s,-10);
%Сигнал с аддитивным гауссовым шумом, SNR = 10
s2=awgn(s,10);
%Построение графика для вектора s
subplot(3,1,1)
plot(t,s);
xlabel('Время')
ylabel('Амплитуда')
title('Сигнал без шума')
%Построение графика для вектора s1
subplot(3,1,2)
plot(t,s1)
xlabel('Время')
ylabel('Амплитуда')
title('Сигнал с аддитивным гауссовым шумом, SNR = -10')
%Построение графика для вектора s2
subplot(3,1,3)
plot(t,s2)
xlabel('Время')
ylabel('Амплитуда')
title('Сигнал с аддитивным гауссовым шумом, SNR = 10')
```



Основную роль в нем играет функция `awgn`, аргументами которой являются:

- вектор-строка, содержащая амплитуды радиосигнала;
- значение отношения сигнал/шум по мощности, дБ.

Для определения пропускной способности AWGN канала воспользуемся формулой Клода Элдвуда Шенона

$$R \leq \Delta\omega \log_2 \left(1 + \frac{P_s}{P_n} \right) = \Delta\omega \log_2(1 + SNR). \quad (2.2)$$

Так, в соответствии с (2.1) и рис. 2.3 видно, что пропускная способность прямо пропорционально зависит от ширины канала связи (спектральной эффективности) и отношения сигнал/шум (2.3).

Для значений отношения сигнал/шум больше нуля можно записать следующее выражения для описания функции

$$\frac{R}{\Delta\omega} \leq \log_2(1 + SNR) \approx \log_2 SNR = \frac{\log_2 10}{10} 10 \lg SNR \approx 0,33 SNR_{\text{дБ}}, \quad (2.3)$$

где $SNR_{\text{дБ}}$ – отношение сигнал/шум по мощности в дБ.

Стоит отметить, что в (2.1) и (2.2) используется отношение сигнал/шум по мощности, в то время как ранее было определено, что в цифровых системах обработки применяется понятие *отношение сигнал/шум на бит*. Тогда с учетом (1.6) преобразуем выражение (2.2)

$$\frac{R}{\Delta\omega} \leq \log_2 \left(1 + \frac{E_b}{N_0} \frac{R}{\Delta\omega} \right). \quad (2.4)$$

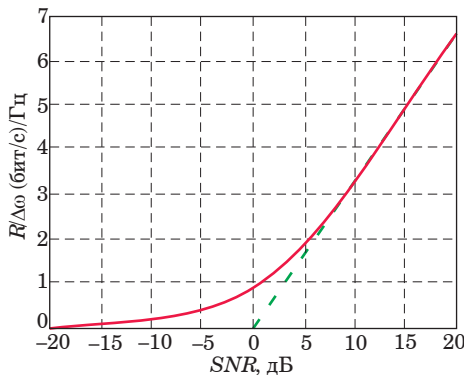


Рис. 2.3. Граница Шенона для отношения сигнал/шум по мощности



Для понимания физики (2.4) для теории цифровых систем обработки информации запишем неравенство для связи спектральной эффективности и отношения сигнал/шум на бит

$$\frac{E_b}{N_0} \geq \frac{2^{R/\Delta\omega} - 1}{R / \Delta\omega}. \quad (2.5)$$

На рис. 2.4 показана пропускная способность (зависимость спектральной эффективности от отношения сигнал/шум на бит). Допустимые значения показаны под границей Шенона (красная сплошная линия) желтым цветом.

При сравнении рис. 2.2 и 2.3 становится понятно, что при определении пропускной способности отношение сигнал/шум на бит практически не может находиться в области отрицательных значений. Это можно объяснить следующим пределом

$$\lim_{R/\Delta\omega \rightarrow 0} \frac{2^{R/\Delta\omega}}{R / \Delta\omega} = \ln 2 \approx 0,7 \approx -1,6 \text{ дБ}. \quad (2.6)$$

Отсюда при отношении сигнал/шум на бит менее минус 1,6 дБ невозможно осуществить передачи без битовых и символьных ошибок при любом помехоустойчивом кодировании и спектральной эффективности. Такое значение отношения сигнал/шум на бит в цифровых системах обработки информации называется *предел Шенона*.

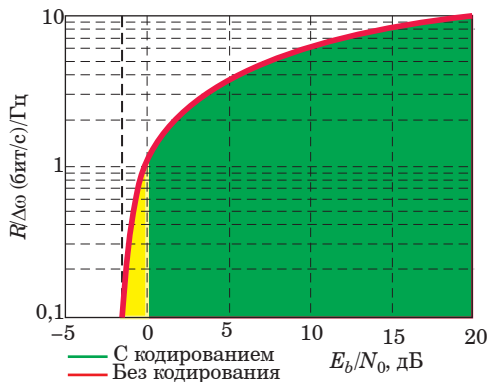


Рис. 2.4. Граница Шенона для отношения сигнал/шум на бит



3. КАНАЛЬНОЕ КОДИРОВАНИЕ

3.1. Общие принципы и терминология

Кодирование – это операция обработки информации (символов) для повышения качества систем цифровой связи, когда информация является цифровой или может быть аппроксимирована в виде дискретных символов. При этом для надежной передачи цифрового сообщения на большие расстояния могут потребоваться как канальное кодирование, так и методы модуляции радиосигнала.

Операция кодирования преобразовывает цифровую информацию в новую последовательность символов. Декодирование преобразует закодированную последовательность обратно в исходное сообщение, возможно, с некоторыми ошибками, вызванными преднамеренными и непреднамеренными помехами в радиоканале. Рассмотрим компьютер или другой цифровой источник, имеющий $M \gg 2$ символа. Для некодированной передачи сообщения из этого источника потребовалось бы M различных форм сигнала, по одной для каждого символа. В качестве альтернативы каждый символ может быть представлен двоичным кодовым словом, состоящим из K двоичных цифр. Поскольку существует 2^K возможных кодовых слов, состоящих из K двоичных цифр, нам нужно $K \geq \log_2 M$ цифр на кодовое слово для кодирования M исходных символов. Если источник выдает r символов в секунду, двоичный код будет содержать Kr цифр в секунду, а требуемая пропускная способность передачи в K раз будет превышать пропускную способность некодированного сигнала.

В обмен на увеличенную пропускную способность двоичное кодирование множества исходных символов дает два преимущества. Во-первых, для обработки двоичного сигнала, состоящего всего из двух разных форм сигнала, требуется менее сложное аппаратное обеспечение. Во-вторых, при прочих равных условиях загрязняющий шум оказывает меньшее влияние на двоичный сигнал, чем на сигнал, состоящий из M различных форм сигнала, поэтому ошибок, вызванных шумом, будет меньше. Следовательно, этот метод кодирования, по существу, является цифровым методом широкополосного шумоподавления. Исключением из вышеуказанного правила было бы, если бы каждая из M различных форм сигнала передавалась на разной частоте, в разном пространстве или была взаимно ортогональной.



Канальное кодирование (помехоустойчивое кодирование) – это метод, используемый для введения контролируемой избыточности для дальнейшего повышения надежности работы в условиях преднамеренных и непреднамеренных помех в радиоканале. Кодирование с контролем ошибок идет дальше в направлении снижения широкополосного шума. Добавляя дополнительные контрольные цифры к каждому двоичному кодовому слову, мы можем обнаружить или даже исправить большинство возникающих ошибок. Кодирование с контролем ошибок увеличивает как пропускную способность, так и аппаратную сложность, но это окупается с точки зрения почти безошибочной цифровой связи, несмотря на низкое отношение сигнал/шум.

Теперь давайте рассмотрим другое фундаментальное системное ограничение – пропускную способность. Многие системы связи полагаются на телефонную сеть для передачи данных. Поскольку пропускная способность системы передачи ограничена разработанными десятилетиями техническими условиями, для увеличения скорости передачи данных необходимо уменьшить полосу пропускания сигнала. Высокоскоростные модемы (модулятор/демодуляторы данных) являются одним из приложений, требующих такого сокращения объема данных. Методы кодирования источника используют преимущества статистических знаний об исходном сигнале для обеспечения эффективного кодирования. Таким образом, исходное кодирование можно рассматривать как двойственное канальному кодированию в том смысле, что оно уменьшает избыточность для достижения желаемой эффективности.

Наконец, преимущества цифрового кодирования могут быть включены в аналоговую связь с помощью метода аналого-цифрового преобразования, такого как импульсно-кодовая модуляция (PCM). Сигнал PCM генерируется путем выборки аналогового сообщения, оцифровки (квантования) значений выборки и кодирования последовательности оцифрованных выборок. Ввиду надежности, универсальности и эффективности цифровой передачи PCM стал важным методом аналоговой связи. Кроме того, в сочетании с высокоскоростными микропроцессорами PCM позволяет заменить аналоговые операции цифровой обработкой сигналов.

В цифровых системах обработки информации существует два основных способа коррекции ошибок в информационных сообщениях. Использование того или иного метода зависит от того что используется в системе обнаружение или же исправление ошибок.



Первый способ основан на запросе повторной передачи (Automatic Repeat reQuest, ARQ). Этот способ предусматривает возможность обнаружения ошибок. После чего система осуществляет запрос, целью которого является попытка повторной передачи данных, и так до тех пор, пока информационное сообщение не будет доставлено без ошибок.

Второй способ предусматривает прямое исправление ошибок (Forward Error Correction, FEC). Он основан на помехоустойчивом коде, декодирование которого исправляет все возникшие в ходе передачи ошибки. В случае невозможности исправить ошибки при N количестве попыток осуществляется запрос на повторную передачу информационного сообщения.

Для примера, на рис. 3.1 показаны две кривые, одна из которых показывает помехоустойчивость для двухпозиционной фазовой модуляции, а вторая для двухпозиционной фазовой модуляции с использованием помехоустойчивого кода (код Хэмминга), который предусматривает исправление ошибок.

Из кривых на рис. 3.1 можно увидеть, что чем меньше отношение сигнал/шум по мощности, тем меньше канальное кодирование влияет на помехоустойчивость сигнала. Более того, при определенных значениях сигнал/шум по мощности видно, что помехоустойчивость сигнала без кодирования выше.

Связано это с тем, что в кодированном сообщении добавлена избыточность, то есть содержится большая битовая последователь-

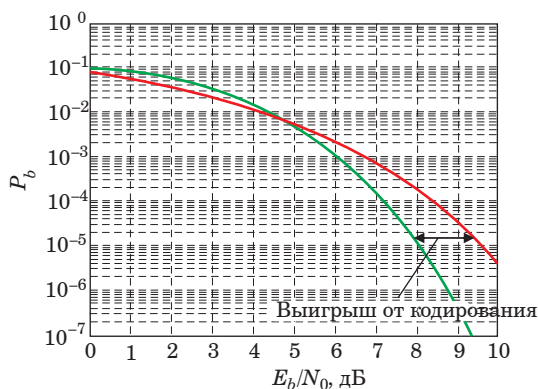


Рис. 3.1. Помехоустойчивость для сигнала с двухпозиционной фазовой модуляцией с использованием и без использования помехоустойчивого кода Хэмминга



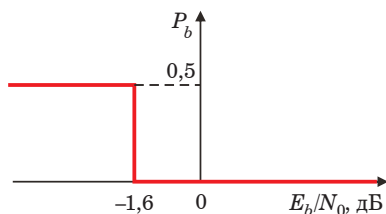


Рис. 3.2. Критические значения функции помехоустойчивости для случая, когда корректирующая способность канального кодирования ничем не ограничена

ность чем в некодированном сообщении. Это в первую очередь в меньшую сторону влияет на энергию сигнала, приходящуюся на один информационный бит. В свою очередь увеличение информации в сообщении влияет на частоту возникновения битовой ошибки в сообщении, прошедшем демодулятор.

Однако при прохождении сообщения через устройство канального декодирования, учитывая, что сообщение было закодировано кодом Хэминга, исправляющего ошибки, частота появления битовых ошибок уменьшается. Становится понятно, что при различном отношении сигнал/шум по мощности эффект от этого различен.

Таким образом, чем больше отношение сигнал/шум, тем больше уровень ошибок, связанный с уменьшением энергии сигнала на один информационный бит. Исходя из этого видно, что на рис. 2 существует область, где функции пересекаются, достигая высокого уровня частоты появления битовых ошибок.

В случае если рост корректирующей способности канального кодирования не будет ограничен, функция помехоустойчивости будет направлена к критическим значениям, которые определяются пределом Шенона. Когда отношение сигнал/шум по мощности превышает предел минус 1,6 дБ, частота появления битовых ошибок стремится к нулевому значению; при других значениях частота появления битовых ошибок равняется 0,5 или близкому к этому значению (рис. 3.2).

3.2. Классификация методов помехоустойчивого кодирования

Одним из основных классификационных признаков методов помехоустойчивого кодирования является их деление на *блоковые* и *сверточные*.



Блочные методы кодирования. При использовании этого метода кодирования в общем случае информационное сообщение разделяется на части (блоки) с одинаковым количеством бит, после чего все блоки подвергаются кодированию самостоятельно. То есть каждый блок можно рассматривать и декодировать отдельно от остальных.

Сверточные методы кодирования. В этом методе кодирования, в отличие от блочных, результирующее кодированное сообщение является зависимым от текущего кодируемого символа и от предыдущих. То есть способ кодирования от символа к символу изменяется, поскольку в случае с методами сверточного кодирования каждый кодируемый символ влияет на состояние кодера.

Не менее важным является деление методов помехоустойчивого кодирования на *систематические* и *несистематические*.

Систематические методы кодирования. Особенностью систематических методов кодирования является то, что в результирующем кодированном информационном сообщении четко читаются символы, содержащие исходную информацию и добавленные для внесения избыточности.

Несистематические методы кодирования. Такие методы кодирования являются обратными системным, то есть четкого разделения на информационные символы и неинформационные не прослеживается.

При выборе систематического или несистематического метода кодирования нельзя опираться на большую эффективность того или иного метода, поскольку она теоретически не подвержена. При этом стоит отметить, что в случае систематического метода кодирования его использование упрощает практическое воплощение, поскольку при этом методе кодирования (в случае отсутствия ошибок) на принимающей стороне для получения информации в декодере необходимо лишь устранить избыточные символы.

В случае когда в результате линейного комбинирования различных кодовых слов получается кодовое слово, соответствующее перечню возможных, такой метод кодирования (код) называется *линейным*. При ином результате метод кодирования (код) является *нелинейным*.

Также методы кодирования могут быть *двоичными* и *недвоичными*. Тот или иной классификационный признак, который присваивается методу кодирования, определяется алфавитом, используемым для кодирования исходного цифрового сообщения.



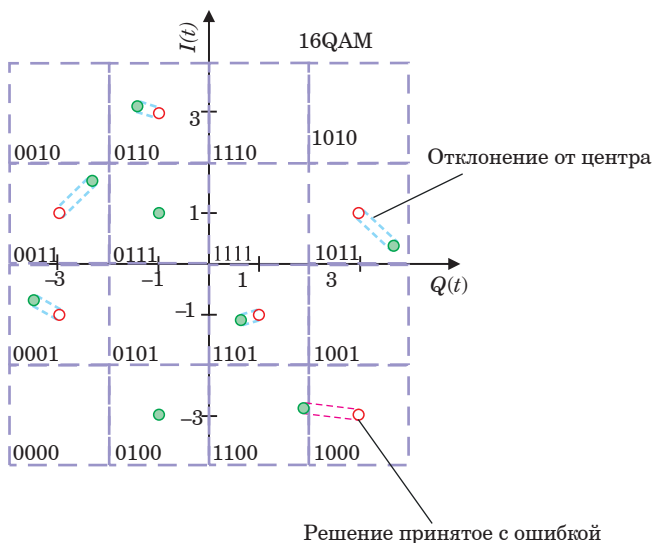


Рис. 3.3. Принятое сигнально созвездие, содержащее ошибки при демодуляции

Для общего понимания функционирования систем стоит отметить еще одну особенность систем, принимающих и декодирующих исходное информационное сообщение. Слабым местом для потенциального возникновения битовых ошибок является решение, принимаемое демодулятором. Оно может приниматься одним из двух вариантов: 1) демодулятор твердо принимает решение о том, какой бит закодирован в сигнале: «1» или «0». После этого решение передается декодеру. В этом случае такое решение не подвергается сомнению, и его достоверность не подкрепляется дополнительными факторами, таким как, например, расположение принятой точки на сигнальном созвездии относительно своего центра; 2) демодулятор принимает решение на основании дополнительной информации, такой как, например, расположение принятой точки на сигнальном созвездии относительно своего центра (рис. 3.3).

3.3. Блочные методы кодирования

Рассмотрим основные понятия при использовании блочных методов кодирования (рис. 3.4):

– при рассмотрении одного блока количество символов, содержащихся в нем, имеет название *длина сообщения k*;



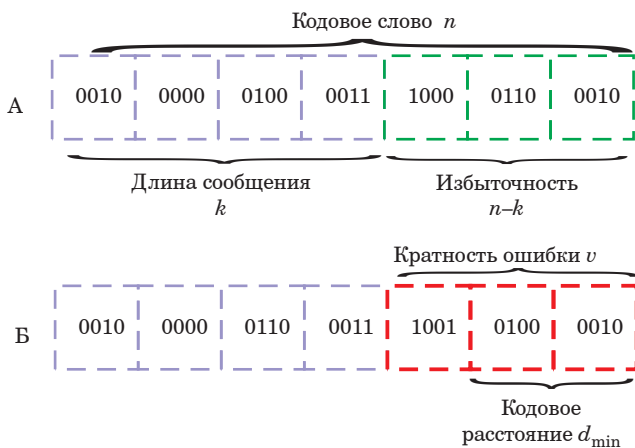


Рис. 3.4. Пояснение к основным понятиям
 блочных методов кодирования

– при добавлении к символам в блоке избыточности (реализации помехоустойчивого кодирования) блок получает название *кодовое слово n* (где избыточность равняется n минус k).

Эти термины в первую очередь необходимы при знакомстве с тем или иным кодом. Принято обозначение «*название кода (n, k)* », то есть обычно в придачу к названию указывают количество символов *кодового слова* и *длину сообщения*, что также может дать понимание того, сколько вводится *избыточных символов*.

Еще одним из основных понятий является *скорость кодирования*, которая определяется отношением

$$r = \frac{k}{n}. \quad (3.1)$$

В теории помехоустойчивого кодирования важнейшим параметром является *кодовое расстояние d_{\min}* . Это величина, которая определяется степенью различия между двумя кодовыми словами, может иметь различную размерность. При этом наиболее распространенным определением кодового расстояния является минимальное количество попарных символов, несовпадающих в двух и более кодовых словах (рис. 3.5).

При этом степень важности этой характеристики подкрепляется тем, что:



– есть случаи, когда кодовое расстояние d_{\min} также указывается к названию кода: «название кода (n, k, d_{\min})»;

– степень помехоустойчивости (или степень влияния на эффективность исправления ошибок) напрямую определяется кодовым расстоянием.

Рассмотрим случай, при котором принятое сообщение содержит ошибку равную кодовому расстоянию или больше. Тогда возможно возникновение ситуации, при которой принятое кодовое слово может восприняться истинным, при этом наличие ошибки не будет выявлено. Однако в случае, если количество ошибок в кодовом слове будет меньше хотя бы на один символ, чем кодовое расстояние, декодер примет решение о том, что кодовое слово не соответствует допустимому, тогда ошибка будет найдена. Таким образом, d_{\min} минус 1 является критическим количеством символов в кодовом слове, при котором ошибка выявляется с вероятностью близкой к 1. Такое количество ошибочных символов называется *кратностью ошибки* v .

Существует распространенное правило, при котором в случае, если декодер принимает решение о том, что кодовое слово не соответствует допустимому, из существующего перечня выбирается такое кодовое слово, которое имеет минимальное кодовое расстояние по отношению к нему. В этом случае возможно три варианта развития событий:

1 – в кодовом слове количество ошибочных символов в два раза меньше, чем кодовое расстояние, тогда исходное кодовое слово будет ближайшим, ошибка в кодовом слове *будет устранена*;

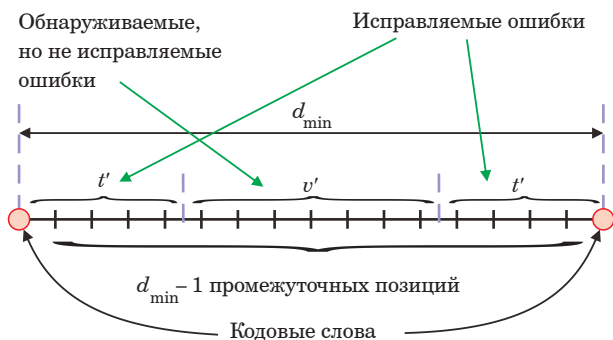


Рис. 3.5. Пояснение к отношению между количеством обнаруживаемых и исправляемых ошибочных символов



2 – в кодовом слове количество ошибочных символов больше, чем $d_{\min}/2$, в этом случае исходное кодовое слово не будет ближайшим, ошибка в кодовом слове *не будет устранена*;

3 – если в кодовом слове количество ошибочных символов равняется $d_{\min}/2$ и кодовое расстояние является четным, то результат может быть неоднозначным, поскольку подходящих кодовых слов может быть два, то есть *ошибка может быть и устранена, и не устранена*.

Учитывая эти факторы, количество символов, однозначно исправляемых при декодировании, определяется выражением

$$t = \frac{d_{\min} - 1}{2}, \quad (3.2)$$

где t – количество символов, однозначно исправляемых при декодировании.

Естественно, при декодировании кодового слова возможен вариант, при котором реализуется и обнаружение, и исправление ошибочных символов. То есть происходит исправление $t' \leq t$ ошибочных символов и в добавление к этому еще обнаружение $v' \leq v$ ошибочных символов. Для воплощения такой возможности необходимо чтобы:

– для исправления ошибочных символов декодируемое сообщение находилось на кодовом расстоянии $d_{\min} < t'$;

– «поднимался флаг» о обнаружении ошибки, если $d_{\min} > t'$;

– для обнаружения и исправления ошибочных символов реализовывалось следующее неравенство

$$2t' + v' \leq d_{\min} - 1. \quad (3.3)$$

Пояснение к (3.3) представлено на рис. 3.6.

Таким образом, поскольку количество обнаруженных ошибочных символов будет больше, чем исправленных, то в системе ARQ частота появления ошибок будет меньше, чем в системе с прямой коррекцией ошибок FEC.

3.3.1. Основные сведения о кодировании и декодировании цифровых сообщений

Рассмотрим основные принципы кодирования на базовом примере линейных методов кодирования, основанных на блоковых методах. При этом кодовые слова формируются путем использования специализированных матриц, которые называются *порождающими*. То есть исходная символьная последовательность умножается на порождающую матрицу. Для наглядности необходимо рассмотреть конкрет-



ный пример, в качестве которого выберем код Хэмминга (7,4), то есть с четырьмя полезными символами и тремя добавленными в качестве избыточности. Для понимания того, из чего образуются символы избыточности, рассмотрим следующее выражение

$$\begin{aligned} y_5 &= x_1 \oplus x_2 \oplus x_3, \\ y_6 &= x_2 \oplus x_3 \oplus x_4, \\ y_7 &= x_1 \oplus x_2 \oplus x_4, \end{aligned} \quad (3.4)$$

где x_n – полезные символы (биты) кодового слова; y_n – избыточные символы (биты) кодового слова.

Все возможные кодовые слова для рассматриваемого кода рассмотрены в табл. 1.

Теперь, имея исходные данные для более понятного пояснения, представим выражение для кодирования исходных символов цифрового сообщения

$$\mathbf{y} = \mathbf{x} \cdot \mathbf{G}, \quad (3.5)$$

где $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]$; $\mathbf{y} = [y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7]$; $\mathbf{G} = \mathbf{k} \cdot \mathbf{n}$ – порождающая матрица, которая в случае текущего примера имеет следующий вид

$$\mathbf{G} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array} \right].$$

Также в теории помехоустойчивого кодирования неотъемлемой частью является матрица, необходимая для проверки выражений, формирующих избыточные символы

$$\begin{aligned} x_1 \oplus x_2 \oplus x_3 \oplus x_5 &= 0, \\ x_2 \oplus x_3 \oplus x_4 \oplus x_6 &= 0, \\ x_1 \oplus x_2 \oplus x_4 \oplus x_7 &= 0. \end{aligned} \quad (3.6)$$

Приводя (3.7) к матричному виду, записываем

$$\mathbf{y} \cdot \mathbf{H}^T = 0, \quad (3.7)$$

где $\mathbf{H} = (\mathbf{n} - \mathbf{k}) \cdot \mathbf{n}$ – проверочная матрица, которая в случае текущего примера имеет вид

$$\mathbf{H} = \left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right].$$



Принимая во внимание все факторы, становится понятно, что (3.7) используют для обнаружения ошибочных символов в принятом блоке.

Раскрывая вопрос проверки кодового слова на предмет наличия ошибок, примем, что произведение порождающей матрицы на проверочную должно быть равно нулю

$$\mathbf{G} \cdot \mathbf{H}^T = 0. \quad (3.8)$$

Если раскрыть (3.7) в соответствии с (3.5), получим, что для каждого кодового слова должно выполняться условие (3.8)

$$\mathbf{x} \cdot \mathbf{G} \cdot \mathbf{H}^T = 0. \quad (3.9)$$

Для обнаружения и исправления ошибочных символов в теории цифровых систем обработки информации применимо такое понятие, как *синдром* \mathbf{s} . Синдромом называется ненулевой результат произведения в (3.7). С помощью синдрома появляется возможность выявить символы, которые являются ошибочными, а также их значения до возникновения искажений

$$\mathbf{y}' \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T = \mathbf{s}, \quad (3.10)$$

где $\mathbf{y}' = \mathbf{y} + \mathbf{e}$ – кодовое слово, претерпевшее искажения; \mathbf{e} – вектор, содержащий ошибочные символы; \mathbf{s} – синдром.

Таким образом, используя полученные значения синдрома и с учетом (3.1), для декодирования кодового слова и исправления ошибок можно использовать табл. 2.

В реальных условиях декодирование с использованием, например, табл. 2, возможно только при малых значениях n и k . Для других случаев используют такие способы кодирования, при которых

Таблица 2

Декодирования для кода Хэмминга (7,4)

Синдром			Вектор ошибочных символов						
s_1	s_2	s_3	e_1	e_2	e_3	e_4	e_5	e_6	e_7
1	0	1	1	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0
0	1	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	1



декодирование осуществляется с использованием математических преобразований.

3.3.2. Весовой спектр и вероятность необнаружения ошибки

Для изучения вопроса вероятности необнаружения ошибочных символов необходимо ввести еще один важный параметр – *весовой спектр*. Количество единичных бит всех символов кодового слова называется *вес*. Множество целых чисел, где каждое из них в отдельности отображает количество существующих кодовых слов с весом d , совпадающим с количеством попарных символов, несопадающих в двух и более кодовых словах, называется *весовым спектром* $\{A_d\}$. В таком случае примем, что для $d=1, \dots, d_{\min} - 1$ весовой спектр будет равен нулю, а для тех случаев, когда в коде содержится кодовое слово, не имеющее в себе единичных бит, $A_0=1$. При анализе табл. 1 была составлена табл. 3, в которой приведены значения весового спектра для различных d в рамках рассматриваемого в качестве примера кода Хэмминга (7,4).

Как уже было сказано ранее, примем, что весовой спектр также может быть использован для вычисления вероятности необнаружения ошибочных символов. Ошибочный символ не будет обнаружен в случае, когда он приводит к тому, что исходное кодовое слово превращается в совершенно другое. Учитывая это, для тех вариантов, когда кодовое расстояние между исходными и ошибочными кодовыми словами соотносится со значением d , в принятом кодовом слове существует d битовых ошибок. Однако тогда n минус d бит не были подвергнуты искажениям в канале передачи данных. Выведем вероятность такого исхода

$$p^d (1 - p)^{n-d}, \quad (3.11)$$

где p – вероятность, при которой будет осуществлен ошибочный прием одного бита переданного кодового слова.

Для вычисления p учитывают используемый вид модуляции (поскольку модуляция влияет на помехоустойчивость сигнала в канале передачи данных), принимая во внимание тот факт, что энер-

Таблица 3

Весовой спектр для кода Хэмминга (7,4)

d	0	1	2	3	4	5	6	7
A_d	1	0	0	7	7	0	0	1



гия бит в канале передачи данных понижается с коэффициентом, определяемым скоростью кодирования r .

В связи с тем, что события, в которых исходное кодовое слово на приемной стороне воспринимается иным кодовым словом являются параллельными, их вероятности p суммируются. Также при линейности рассматриваемого для примера метода кодирования отпадает необходимость усреднения результатов вычисления по различным кодовым словам в связи с тем, что результаты совпадут. Учитывая это, используем только одно кодовое слово, а для удобства возьмем то, которое не содержит единичных бит. Таким образом, вероятность необнаружения будет определяться выражением

$$P_{\text{необн}} = \sum_{d=d_{\min}}^n A_d p^d (1-p)^{n-d}. \quad (3.12)$$

Продолжая пояснение на примере кода Хэмминга (7,4) с учетом (3.12) и табл. 3, выведем формулу для $P_{\text{необн}}$

$$P_{\text{необн}} = 7p^3(1-p)^{7-3} + 7p^4(1-p)^{7-4} + p^7(1-p)^{7-7} = 7p^3(1-p)^3 + p^7. \quad (3.13)$$

Результат вычисления в (3.13) продемонстрирован в виде функции зеленого цвета на рис. 3.6. При вычислении для примера была

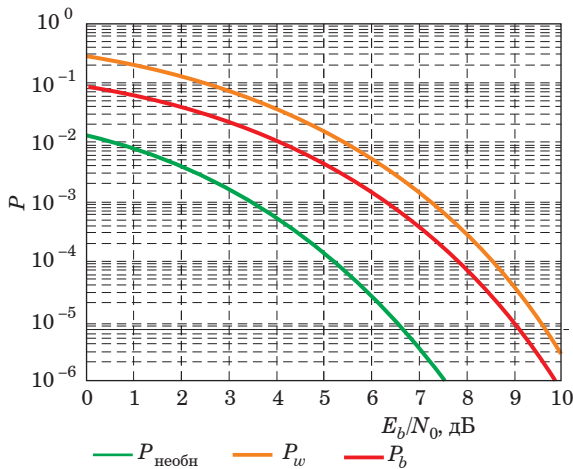


Рис. 3.6. Вероятности событий, при которых ошибка не обнаруживается, осуществлено неправильное декодирование и произошла битовая ошибка для кода Хэмминга (7,4)



выбрана модуляция сигнала методом ФМ-2, в связи с чем p определяется как

$$p = Q \left(\sqrt{r \frac{2E_b}{N_0}} \right). \quad (3.14)$$

где $r=4/7$ – скорость кодирования.

3.3.3. Вероятность ошибки декодирования

В настоящем подпункте будет затронута тема оценки вероятности возникновения события, при котором возникает ошибка в процессе декодирования P_w .

Как было выяснено ранее, в линейном блоковом коде присутствует количество символов t , которое будет исправлено с вероятностью, близкой к единице. Тогда вероятность возникновения ошибки при декодировании появляется в том случае, если в блоке присутствует больше, чем t ошибочных символов. Вероятность события, при котором в канале передачи данных окажутся искажены d бит, приведена в (3.11). Набор вариаций ошибочных бит равняется количеству сочетаний n по d (C_n^d). Исходя из этого, появляется возможность определить верхнюю границу для P_w

$$P_w \leq \sum_{d=t+1}^n C_n^d p^d (1-p)^{n-d} = 1 - \sum_{d=0}^t C_n^d p^d (1-p)^{n-d}. \quad (3.15)$$

Стоит учесть и закрепить понимание того, что выражением (3.15) определяется лишь критическое значение вероятности, в то время как тот или иной метод кодирования в частных случаях способен вносить исправления в кодовые слова, содержащие ошибки больше t . В случае если метод кодирования способен исправлять произошедшие ошибки с кратностью вплоть до t и не более, то такой метод кодирования называют *совершенным*.

При использовании, например, метода кодирования Хэмминга (7,4) кодовое расстояние d_{\min} равняется трем, что означает исправление только одного ошибочного символа (бита), то есть $t=1$. Также этот метод кодирования относится к совершенным, поэтому для него (3.15) можно записать как

$$P_w \leq \sum_{d=t+1}^n C_n^d p^d (1-p)^{n-d} = 1 - \sum_{d=0}^t C_n^d p^d (1-p)^{n-d}. \quad (3.16)$$



Функция, построенная по (3.16), представлена на рис. 3.6 оранжевой линией. Анализируя функции, можно заметить, что вероятность события, при котором ошибка не будет обнаружена, меньше чем вероятность события, при котором будет осуществлена ошибка при декодировании.

Вероятность, при которой будет допущена битовая ошибка, является зависимой от метода кодирования и определяется выражением

$$P_b \approx \frac{1}{n} \sum_{d=t+1}^n d C_n^d p^d (1-p)^{n-d} = \sum_{d=t+1}^n C_{n-1}^{d-1} p^d (1-p)^{n-d} = p \left(1 - \sum_{d=0}^{t-1} C_{n-1}^d p^d (1-p)^{n-1-d} \right). \quad (3.17)$$

Выражение (3.17) получено исходя из того, что количество ошибочных бит (символов) больше t , а при декодировании не вносятся/исправляются существующие ошибки; ошибочные символы одинаково присутствуют в полезной части кодового слова и в избыточной.

Так, для метода кодирования, которое в настоящем пособии используется в качестве демонстрационного примера, вероятность битовой ошибки вычисляется по формуле

$$P_b \approx p \left(1 - \sum_{d=0}^{t-1} C_6^d p^d (1-p)^{6-d} \right) = p \left(1 - (1-p)^6 \right). \quad (3.18)$$

На рис. 3.6 также приведен график, построенный по выражению (3.18), где функция продемонстрирована красным цветом.

3.4. Сверточные методы кодирования

Особенностью сверточного метода кодирования является структура, которая эффективно распространяется на все передаваемое цифровое сообщение, а не ограничивается блоками кодовых слов. Сверточная структура особенно хорошо подходит для космических и спутниковых систем связи, которые требуют простых устройств кодирования и достигают высокой производительности за счет сложных методов декодирования. Рассмотрение этого важного семейства кодов состоит из примеров, которые знакомят с характерными особенностями сверточного кодирования и декодирования.



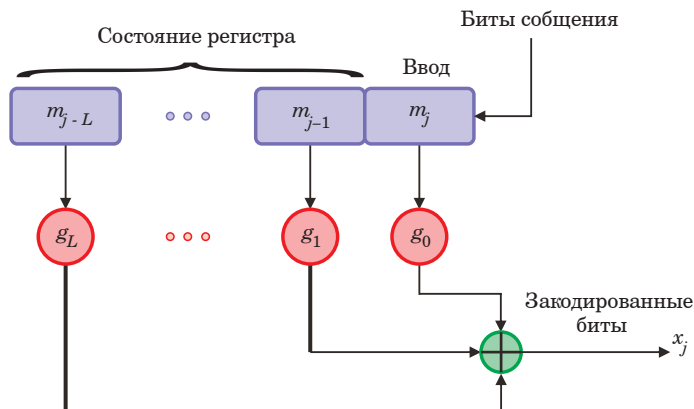


Рис. 3.7. Подключаемый регистр сдвига для сверточного метода кодирования

Основным аппаратным блоком для сверточного кодирования является регистр сдвига с отводами с $L+1$, как показано на рис. 3.7. Коэффициент усиления каждого отвода g представляет собой двоичную цифру, представляющую соединение с коротким замыканием или разомкнутую цепь. Биты сообщения в регистре объединяются путем сложения по модулю 2 для формирования закодированного бита

$$x_j = m_{j-L}g_L \oplus \dots \oplus m_{j-1}g_1 \oplus m_jg_0 = \sum_{i=0}^L m_{j-i}g_i. \quad (3.19)$$

Название *сверточного метода кодирования* происходит от того факта, что уравнение (3.19) имеет форму двоичной свертки, аналогичной сверточному интегралу

$$x(t) = \int m(t - \lambda)g(\lambda)d\lambda. \quad (3.20)$$

Обратите внимание, что в уравнении (3.19) x_j зависит от текущего входного m_j и от состояния регистра, определенного предыдущими битами цифрового сообщения L . Также стоит отметить, что конкретный бит сообщения влияет на интервал из $L+1$ последовательных закодированных бит при его перемещении через регистр.

Чтобы обеспечить избыточные биты, необходимые для обнаружения и устранения ошибок, полное устройство сверточного кодирования должно генерировать выходные биты со скоростью, превышающей скорость передачи сообщений r_b . Это достигается пу-



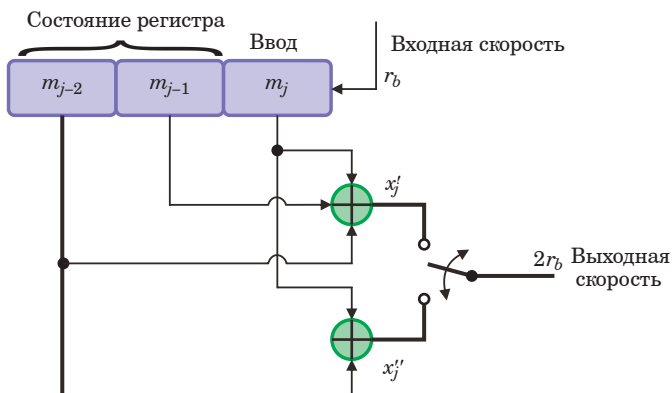


Рис. 3.8. Сверточное устройство кодирования с $n=2$, $k=1$ и $L=2$

тем подключения двух или более суммирующих устройств по модулю 2 к регистру и чередования закодированных бит с помощью коммутатора. Например, кодер на рис. 3.8 генерирует $n=2$ закодированных бита

$$x'_j = m_{j-2} \oplus m_{j-1} \oplus m_j \quad \text{и} \quad x''_j = m_{j-2} \oplus m_j. \quad (3.21)$$

Эти биты чередуются переключателем для получения выходного потока

$$X = x_1' x_1'' x_2' x_2'' x_3' x_3'' \dots$$

Таким образом, выходная скорость передачи битов равна $2r_b$, а скорость кодирования равна $R_c=1/2$ – как в (n, k) блочном коде с $R_c = k/n = 1/2$.

Однако в отличие от блочного кода, биты на входе не были сгруппированы в кодовые слова. Вместо этого каждый бит сообщения влияет на интервал из $n(L+1)$ последовательных выходных бит. Величина $n(L+1)$ называется *длиной ограничения*, измеренной в терминах закодированных выходных бит, тогда как L – это *память кодера*, измеренная в терминах бит входного сообщения. Мы говорим, что этот кодер выдает (n, k, L) сверточный код с $n=2$, $k=1$ и $L=2$.

Для изучения сверточного метода кодирования были разработаны три различных, но связанных графически представления: *кодированное дерево*, *кодированная решетка* и *диаграмма состояний*. Представим каждое из них для устройства кодирования $(2, 1, 2)$ на рис. 3.8, начиная с кодированного дерева. В соответствии с обычной рабочей про-



цедурой мы предполагаем, что регистр был очищен, чтобы в его состоянии были все нули, когда поступает первый бит сообщения m_1 . Следовательно, начальное состояние равно $m_{-1}m_0=00$, а уравнение (3.21) выдает результат $x_1'x_1''$, если $m_1=0$, или $x_1'x_1''=11$, если $m_1=1$.

Кодовое дерево, нарисованное на рис. 3.9, начинается с точки ветвления или узла с отметкой a , представляющей начальное состояние. Если $m_1=0$, мы берем верхнюю ветвь от узла a , чтобы найти выходное значение 00 и следующее состояние, которое также помечено как a , поскольку в этом случае $m_0m_1=00$. Если $m_1=1$, мы берем нижнюю ветвь от a , чтобы найти выходное значение 11 и сле-

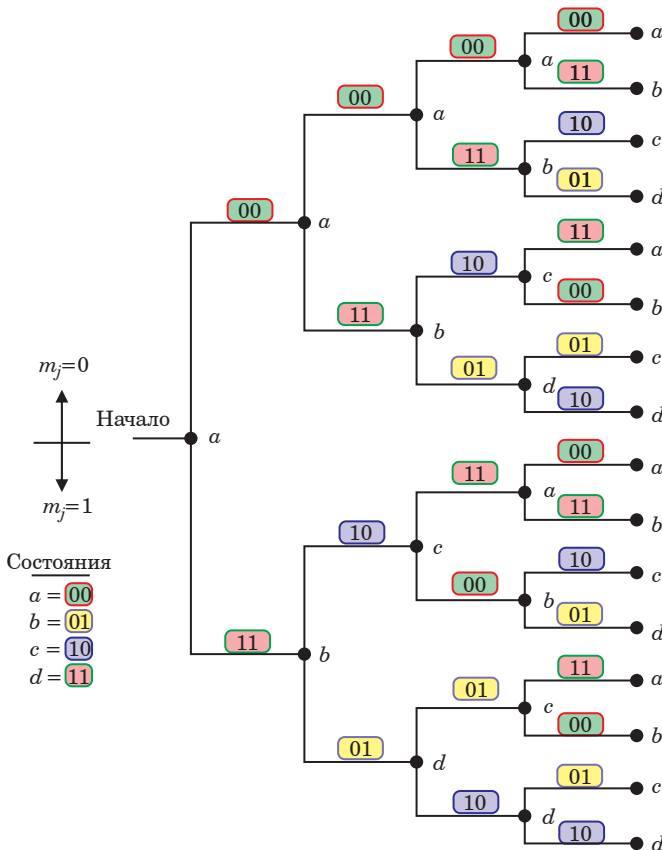


Рис. 3.9. Кодовое дерево для устройства кодирования (2, 1, 2)



дующее состояние $m_0m_1=01$, обозначенное меткой b . Таким образом, дерево кода постепенно эволюционирует для каждого нового входного бита. Узлы помечены буквами, обозначающими текущее состояние $m_{j-2}m_{j-1}$; мы идем вверх или вниз от узла, в зависимости от значения m_j ; каждая ветвь показывает результирующий закодированный вывод $x_j'x_j$, вычисленный из уравнения (3.19), и он заканчивается на другом узле, помеченном следующим состоянием. Существует 2^j возможных ветвей для i -го бита сообщения, но шаблон ветвления начинает повторяться при $j=3$, поскольку длина регистра равна $L+1=3$.

Наблюдая повторение в кодовом дереве, мы можем построить более компактную картину, называемую *кодовой решеткой*, показанной на рис. 3.10а. Здесь узлы слева обозначают четыре возможных текущих состояния, в то время как узлы справа являются результирующими следующими состояниями. Сплошная зеленая линия представляет переход состояния или ветвь для $m_j=0$, а прерывистая фиолетовая линия представляет ветвь для $m_j=1$. Каждая ветвь помечается результирующими выходными битами $x_j'x_j$. Делая еще один шаг вперед, мы объединяем левую и правую стороны решетки, чтобы получить *диаграмму состояния* на рис. 3.10б. Циклы в узлах a и d представляют собой переходы состояний a - a и d - d .

Учитывая последовательность бит цифрового сообщения и начальное состояние, вы можете использовать либо кодовую решетку,

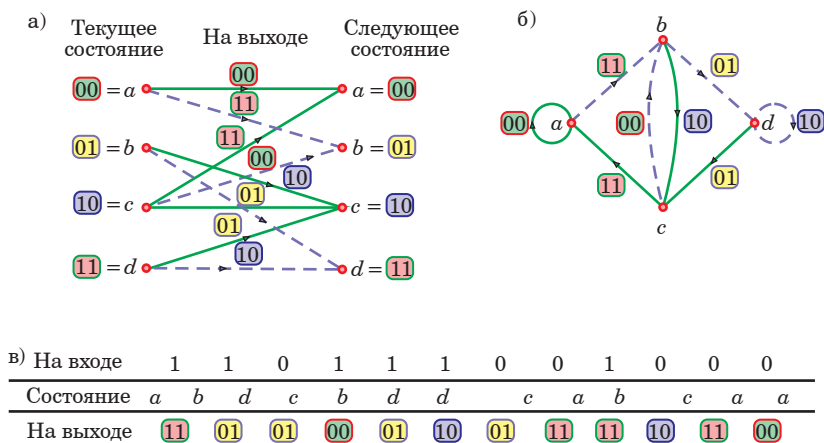


Рис. 3.10. Схема: а – кодовая решетка; б – диаграмма состояний для (2, 1, 2) кодера; в – битовая последовательность для общего понимания



либо диаграмму состояний, чтобы найти результирующую последовательность состояний и выходные биты. Процедура продемонстрирована на рис. 3.10в, начиная с начального состояния a .

Другие многочисленные сверточные коды получаются путем модификации устройства кодирования, продемонстрированного на рис. 3.8. Если мы просто изменим соединения с суммами по модулю 2, то кодовое дерево, решетка и диаграмма состояний сохранят ту же структуру, поскольку состояния и шаблон ветвления отражают только содержимое регистра. Выходные биты, конечно, будут разными, так как они конкретно зависят от подключений сервера.

Если мы расширим регистр сдвига до произвольной длины $L+1$ и подключим его к $n \geq 2$ суммам по модулю 2, то получим (n, k, L) сверточный код с $k=1$ и кодовой скоростью $R_c = 1/n \leq 1/2$. Состояние кодера определяется L предыдущими входными битами, поэтому кодовая решетка и диаграмма состояний имеют $2L$ разных состояний, а шаблон кодового дерева повторяется после $j=L+1$ ветвей. Подключение одного терминала коммутатора непосредственно к первому каскаду регистра приводит к закодированному потоку бит, которые определяет систематический сверточный код с $R_c = 1/2$

$$X = m_1 x_1'' x_1''' \dots m_2 x_2'' x_2''' \dots m_3 x_3'' x_3''' \dots \quad (3.22)$$

Опять же, обратите внимание, что в систематическом коде информационные биты являются непосредственно частью кодового слова. За исключением турбокодов и кодов, используемых при модуляции с кодированием в виде решетки, большинство сверточных кодов являются несистематическими.

Скорость кодирования выше $1/n$ требует $k \geq 2$ регистров сдвига и переключателя распределителя входного сигнала. Эта схема проиллюстрирована кодером $(3, 2, 1)$ на рис. 3.11. Биты сообщения распределяются поочередно между $k=2$ регистрами, каждый из которых имеет длину $L+1=2$. Мы рассматриваем пару битов как текущий входной сигнал, в то время как пара $m_{j-3}m_{j-2}$ представляет собой состояние кодера. Для каждой входной пары суммирующие блоки по модулю 2 генерируют $n=3$ закодированных выходных бита, задаваемых исходя из

$$\begin{aligned} x_j' &= m_{j-3} \oplus m_{j-2} \oplus m_j; \\ x_j'' &= m_{j-3} \oplus m_{j-1} \oplus m_j; \\ x_j''' &= m_{j-2} \oplus m_j. \end{aligned} \quad (3.23)$$



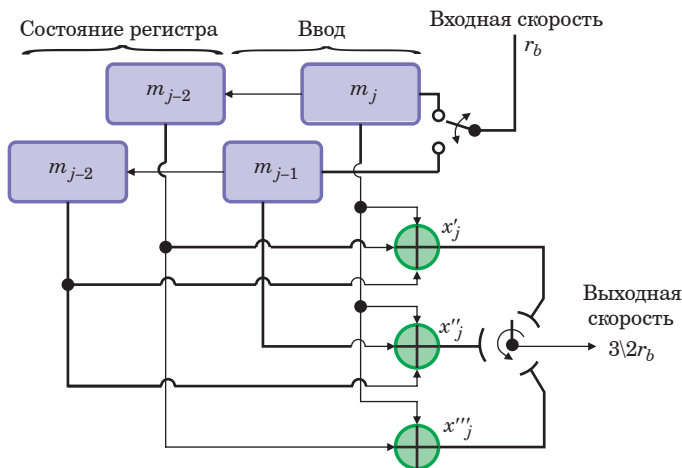


Рис. 3.11. Устройство кодирования, основанного на сверточном методе кодирования (3, 2, 1)

Таким образом, выходная скорость передачи бит составляет $3r_b/2$, что соответствует скорости кодирования $R_c = k/n = 2/3$. Длина ограничения равна $n(L+1) = 6$, поскольку конкретный входной бит влияет на диапазон из $n = 3$ выходных битов из каждой $L+1 = 2$ позиций регистра.

Графическое представление становится более громоздким для сверточных кодов с $k > 1$, поскольку мы должны иметь дело с входными битами в группах по 2^k . Следовательно, 2^k ветвей исходят и заканчиваются на каждом узле, и существует 2^{kL} различных состояний. В качестве примера на рис. 3.12 показана диаграмма состояния для устройства кодирования, основанного на сверточном методе кодирования (3, 2, 1), продемонстрированного на рис. 3.11. Ветви помечаются $k = 2$ входными битами, за которыми следуют результирующие $n = 3$ выходных бита.

Сверточные коды, используемые для систем ФЕС, обычно имеют небольшие значения n и k , в то время как длина ограничения обычно находится в диапазоне от 7 до 30. Все сверточные энкодеры требуют наличия коммутатора на выходе, как показано рис. 3.8–3.12.

Для кодов с $k > 1$ входной распределительный переключатель может быть устранен путем использования одного регистра длиной kL и сдвига битов в группах по k .

В любом случае аппаратное обеспечение сверточного кодирования проще, чем аппаратное обеспечение для блочного кодиро-



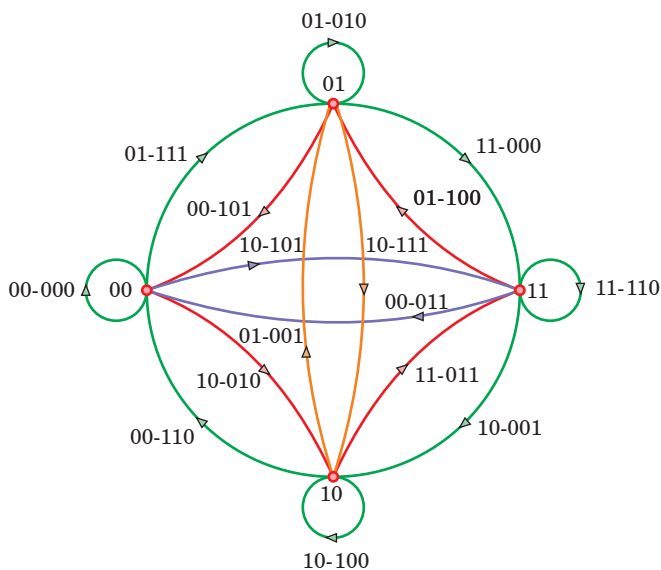


Рис. 3.12. Диаграмма состояния для устройства кодирования, основанного на сверточном методе кодирования (3, 2, 1)

вания, поскольку биты сообщения поступают в регистровый блок с постоянной скоростью r_b и входной буфер не требуется.

3.5.1. Использование коэффициента усиления при кодировании

Ранее мы выяснили, что способность блочного кода контролировать ошибки зависит от его минимального кодового расстояния, определяемого по весам кодовых слов. Сверточный код не подразделяется на кодовые слова, поэтому вместо этого мы рассматриваем вес $w(X)$ всей переданной последовательности X , сгенерированной некоторой последовательностью сообщений. Затем свободное расстояние сверточного кода определяется как

$$d_f = [w(X)]_{\min}, \quad (3.24)$$

$$X \neq 000 \dots$$

где d_f – мера мощности контроля ошибок.

Было бы чрезвычайно скучной и утомительной задачей пытаться оценить d_f , перечисляя все возможные передаваемые последова-



тельности. К счастью, есть лучший способ, основанный на обычной рабочей процедуре добавления «хвоста» из нулей в конце сообщения, чтобы очистить регистровый блок и вернуть кодер в исходное состояние. Эта процедура удаляет определенные ветви из кодовой решетки для последних L переходов.

Возьмем, к примеру, кодовую решетку на рис. 3.10а. Чтобы перейти в состояние a , предпоследнее состояние должно быть либо a , либо c , поэтому последние несколько ветвей любой передаваемой последовательности X должны следовать по одному из путей, показанных на рис. 3.14. Здесь конечное состояние обозначается e , и каждая ветвь была помечена числом 1 в закодированных битах, что равно весу, связанному с этой ветвью. Общий вес передаваемой последовательности X равен сумме весов ответвлений вдоль пути X . В соответствии с уравнением (3.24) мы ищем путь, который имеет наименьшую сумму весов ветвей, отличную от тривиального пути с нулевым значением.

Оглядываясь назад на $L+1=3$ ответвления от e , мы находим последний путь, который исходит из состояния a перед завершением в e . Теперь предположим, что все предыдущие переходы следовали по нулевому пути вдоль верхней строки, давая последовательность состояний $aa\cdots abce$. Поскольку ветвь a - a имеет вес 0, эта последовательность состояний соответствует нетривиальному пути с *минимальным весом*. Поэтому мы приходим к выводу, что $d_f=0+0+\cdots+0+2+1+2=6$. Существуют и другие пути с минимальным весом, такие как $aa\cdots abc$ и $aa\cdots abc$, но ни один нетривиальный путь не имеет меньшего веса, чем $d_f=6$.

Другой подход к вычислению свободного расстояния включает в себя *генерирующую функцию* сверточного кода. Генерирующую функцию можно рассматривать как передаточную функцию коде-

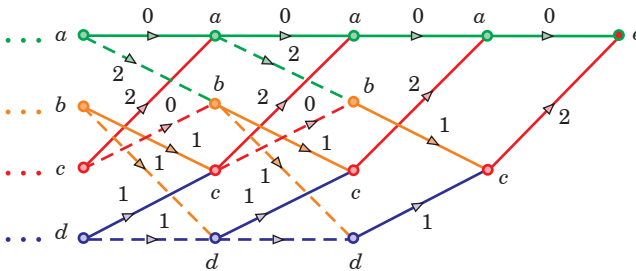


Рис. 3.13. Завершение кодовой решетки (2, 1, 2)



ра по отношению к переходам состояний. Таким образом, вместо того, чтобы связывать входные и выходные потоки битов путем свертки, генерирующая функция связывает начальное и конечное состояния путем умножения. Генерирующие функции предоставляют важную информацию о производительности кода, включая свободное расстояние и вероятность ошибки декодирования.

Мы разработаем генерирующую функцию для нашего кодера (2, 1, 2), используя модифицированную диаграмму состояний на рис. 3.14а. Эта диаграмма была получена из рис. 3.10б с четырьмя модификациями. Во-первых, мы устранили цикл $a-a$, который ничего не добавляет к весу последовательности X . Во-вторых, нарисовали ветвь $c-a$ как конечный переход $c-e$. В-третьих, присвоили переменную состояния W_a узлу a , а также всем другим узлам. В-четвертых, обозначили каждую ветвь двумя переменными «усиления» D и I так, что показатель степени D равен весу ветви (как на рис. 3.13), в то время как показатель степени I равен соответствующему числу ненулевых битов сообщения (что обозначается сплош-

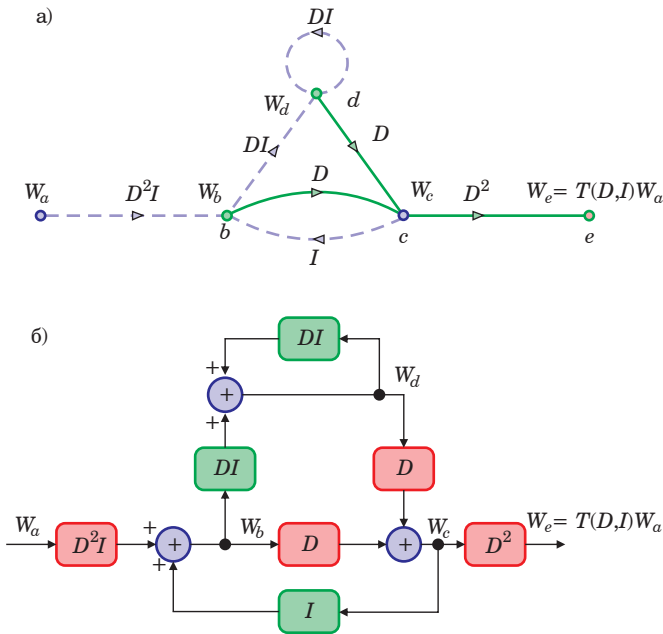


Рис. 3.14. Последовательность пути: а – модифицированная диаграмма состояний для устройства кодирования (2, 1, 2); б – соответствующая к ней структурная схема



ной или пунктирной линией ответвления). Например, поскольку ветвь $c-e$ представляет $x'_j x''_j = 11$ и $m_j = 0$, она помечена как $D^2 I^0 = D^2$. Этот экспоненциальный трюк позволяет нам выполнять *суммирование путем умножения* членов D и I , которые станут независимыми переменными производящей функции.

Наша модифицированная диаграмма состояний теперь выглядит как график потока сигналов того типа, который иногда используется для анализа систем с обратной связью. В частности, если мы рассматриваем узлы как суммирующее соединение, а члены DI – как усиления ветвей, то на рис. 3.14а представлен набор алгебраических уравнений состояния

$$\begin{aligned} W_b &= D^2 I W_a + I W_c & W_c &= D W_b + D W_d \\ W_d &= D I W_b + D I W_d & W_e &= D^2 W_c. \end{aligned} \quad (3.25)$$

Генерирующая функция кодера $T(D, I)$ теперь может быть определена *уравнением ввода-вывод*

$$T(D, I) = \frac{W_e}{W_a}. \quad (3.26)$$

Эти уравнения также эквивалентны структурной схеме, которая представлена на рис. 3.14б, дополнительно подчеркивающая взаимосвязи между переменными состояния, коэффициентами усиления ветвей и производящей функцией. Обратите внимание, что здесь введен знак минус, чтобы два пути обратной связи $c-b$ и $d-d$ соответствовали отрицательной обратной связи.

Далее выражение для $T(D, I)$ получается алгебраическим решением уравнения (3.26) или сокращением структурной схемы на рис. 3.14б с использованием соотношений передаточных функций для параллельных, каскадных соединений и соединений с обратной связью. Любой из этих методов дает конечный результат

$$T(D, I) = \frac{D^5 I}{1 - 2DI}, \quad (3.27)$$

$(1-2DI)^{-1}$ было раскрыто, чтобы получить ряд в уравнении (3.28):

$$\begin{aligned} T(D, I) &= D^5 I + 2D^6 I^2 + 4D^7 I^3 + \dots \\ &= \sum_{d=5}^{\infty} 2^{d-5} D^d I^{d-4}. \end{aligned} \quad (3.28)$$



Имея в виду, что $T(D, I)$ представляет все возможные передаваемые последовательности, которые заканчиваются переходом $c-e$, уравнение (3.28) имеет следующую интерпретацию: для любого $d \geq 5$ существует ровно 2^{d-5} допустимых путей с весом $w(X)$, которые заканчиваются переходом a в переход $c-e$, и эти пути генерируются сообщениями, содержащими $d-4$ ненулевых бита. Наименьшее значение $w(X)$ – это свободное расстояние, поэтому мы снова приходим к выводу, что $d_f=6$.

В качестве обобщения уравнений (3.27) и (3.28) генерирующая функция для произвольного сверточного кода принимает вид

$$T(D, I) = \sum_{d=d_f}^{\infty} \sum_{i=1}^{\infty} A(d, i) D^d I^i, \quad (3.29)$$

где $A(d, i)$ – количество различных путей ввода-вывода через измененную диаграмму состояний, которые имеют вес d и генерируются сообщениями, содержащими i ненулевых битов.

Теперь рассмотрим принятую последовательность $Y=X+E$, где E представляет ошибки передачи. Затем путь Y отклоняется от пути X и может быть или не быть допустимым путем для рассматриваемого кода. Когда Y не соответствует допустимому пути, декодер максимального правдоподобия должен искать допустимый путь, который имеет наименьшее расстояние Хэмминга от Y . Прежде чем описать, как может быть реализован такой декодер, мы установим взаимосвязь между генерирующими функциями, свободным расстоянием и вероятностью ошибки при декодировании сверточных кодов с максимальным правдоподобием.

Если ошибки передачи происходят с равной и независимой вероятностью α на бит, то вероятность ошибки декодированного сообщения в битах ограничена сверху на

$$P_{be} \leq \frac{1}{k} \left. \frac{\partial T(D, I)}{\partial I} \right|_{D=2\sqrt{\alpha(1-\alpha)}, I=1}. \quad (3.30)$$

Когда α достаточно мал, разложение в ряд $T(D, I)$ дает аппроксимацию

$$P_{be} \approx \frac{M(d_f)}{k} 2^{d_f} \alpha^{d_f/2}, \quad (3.31)$$

где $M(d_f) = \sum_{i=1}^{\infty} i A(d_f, i)$; $\sqrt{\alpha} \ll 1$.



Величина $M(d_f)$ просто равна общему числу ненулевых битов сообщения по всем путям ввода-вывода с минимальным весом в измененной диаграмме состояний.

Уравнение (3.31) подтверждает наше предыдущее утверждение о том, что способность сверточного кода контролировать ошибки зависит от его свободного расстояния. Для сравнения производительности с некодированной передачей мы сделаем обычное предположение о том, что $\text{SNR}=2R_c\gamma_b \geq 10$, поэтому уравнение (3.31) дает вероятность ошибки передачи

$$\alpha \approx (4\pi R_c \gamma_b)^{-1/2} e^{-R_c \gamma_b}. \quad (3.32)$$

Затем вероятность декодированной ошибки становится

$$P_{be} \approx \frac{M(d_f)2^{d_f}}{k(4\pi R_c \gamma_b)^{d_f/4}} e^{-(R_c d_f/2)\gamma_b}. \quad (3.33)$$

Принимая во внимание, что некодированная передача приведет к

$$P_{ube} \approx \frac{1}{(4\pi\gamma_b)^{1/2}} e^{-\gamma_b}. \quad (3.34)$$

Поскольку экспоненциальные члены доминируют в этих выражениях, мы видим, что сверточное кодирование повышает надежность, когда $R_c d_f/2 > 1$. Соответственно, величина $R_c d_f/2$ известна как *коэффициент усиления при кодировании*, обычно выражаемый в дБ.

Таблица 4

**Максимальное расстояние и коэффициент кодирования
выбранных сверточных кодов**

n	k	R_c	L	d_f	$R_c d_f/2$	$R_c d_f/2$ (дБ)
4	1	1/4	3	13	1,63	2,12
3	1	1/3	3	10	1,68	2,25
2	1	1/2	3	6	1,5	1,176
			6	10	2,5	3,98
			9	12	3	4,77
3	2	2/3	3	7	2,33	3,67
3	3	3/4	3	8	3	4,77



К сожалению, явных формул проектирования для d_f не существует, поэтому хорошие сверточные коды должны быть обнаружены с помощью компьютерного поиска и моделирования. В табл. 4 перечислены максимальное свободное расстояние и коэффициент усиления при кодировании сверточных кодов для выбранных значений n , k и L . Обратите внимание, что свободное расстояние и коэффициент усиления при кодировании увеличиваются с увеличением объема памяти L , когда скорость кодирования R_c остается фиксированной. Все перечисленные коды являются несистематическими; систематический сверточный код имеет меньший d_f , чем оптимальный несистематический код с той же скоростью и памятью.

Также мы должны отметить, что некоторые сверточные коды демонстрируют *катастрофическое распространение ошибок*. Это происходит, когда конечное число ошибок канала вызывает бесконечное число ошибок декодирования, даже если последующие символы верны. Кодеры, которые демонстрируют такое поведение, покажут на своей диаграмме состояний то, в котором заданный ненулевой входной сигнал вызывает переход обратно в это состояние, что само по себе приводит к нулевому выходу. Катастрофические коды также могут быть идентифицированы, если их порождающие полиномы имеют общий коэффициент степени, по крайней мере один.

3.5.2. Методы декодирования сверточных кодов

Существует три общих метода декодирования сверточных кодов. С одной стороны, алгоритм Витерби выполняет декодирование с максимальным правдоподобием и достигает оптимальной производительности, но требует большого количества аппаратных средств для вычислений и хранения. С другой стороны, декодирование с обратной связью жертвует производительностью в обмен на упрощенное аппаратное обеспечение. Между этими крайностями последовательное декодирование приближается к оптимальной производительности в той степени, которая зависит от сложности декодера. Мы опишем, как эти методы работают с кодом $(2, 1, L)$. Расширение на другие коды концептуально просто, но становится запутанным для отображения при $k > 1$.

Напомним, что декодер максимального правдоподобия должен исследовать всю принятую последовательность Y и найти допустимый путь, который имеет наименьшее расстояние Хэмминга от Y . Однако существует 2^N возможных путей для произвольной после-



довательности сообщений из N битов (или N_n/k битов в Y), поэтому исчерпывающее сравнение Y со всеми допустимыми путями было бы абсурдной задачей в обычном случае $N \gg 1$. Алгоритм Витерби применяет принципы максимального правдоподобия, чтобы ограничить сравнение 2^{kL} выживших путей, независимо от N , тем самым переводя декодирование с максимальным правдоподобием в область осуществимости.

Декодер Витерби присваивает каждой ветви каждого сохранившегося пути метрику равную его расстоянию Хэмминга от соответствующей ветви Y . (Здесь мы предполагаем, что нули и единицы имеют одинаковую вероятность ошибки передачи; если нет, метрика ветвления должна быть переопределена для учета различных вероятностей.) Суммирование метрик ветвления дает метрику пути, и Y , наконец, декодируется как оставшийся путь с наименьшей метрикой. Чтобы проиллюстрировать вычисления метрики и объяснить, как выбираются выжившие пути, мы рассмотрим пример декодирования Витерби.

Предположим, что на передатчике используется наш кодер (2, 1, 2), и принятая последовательность начинается с $Y=11\ 01\ 11$. На рис. 3.15 показаны первые три ветви допустимых путей, исходящих из начального узла a_0 в кодовой решетке. Число в круглых скобках под каждой ветвью – это метрика ветвления, полученная

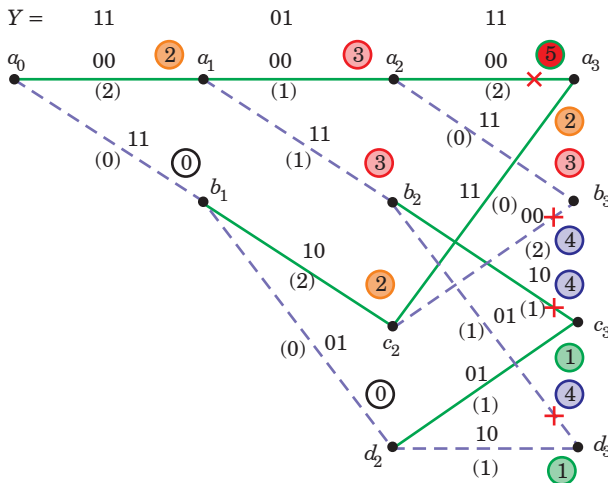


Рис. 3.15. Принятая устройством кодирования последовательности $Y=11\ 01\ 11$



путем подсчета разностей между закодированными битами и соответствующими битами в Y . Число, обведенное кружком в правом конце каждой ветви, представляет собой метрику текущего пути, полученную путем суммирования метрик ветвей из a_0 . Например, метрика пути $a_0b_1c_2b_3$ равна $0+2+2=5$.

Теперь обратите внимание, что другой путь $a_0a_1a_2b_3$ также достигает узла b_3 и имеет меньшую метрику $2+1+0=3$. Независимо от того, что произойдет впоследствии, этот путь будет иметь меньшее расстояние Хэмминга от Y , чем другой путь, поступающий в b_3 , поэтому с большей вероятностью будет представлять фактическую переданную последовательность. Следовательно, мы отбрасываем путь с большей метрикой, отмеченный знаком \times , и объявляем путь с меньшей метрикой выжившим в этом узле. Аналогично мы отбрасываем пути с большей метрикой, прибывающие в узлы a_3 , c_3 и d_3 , оставляя в общей сложности $2^{KL}=4$ выживших пути. Тот факт, что ни одна из сохранившихся метрик пути не равна нулю, указывает на наличие обнаруживаемых ошибок в Y .

На рис. 3.16 показано продолжение рис. 3.15 для полного сообщения из $N=12$ битов, включая конечные 0. Все отброшенные ветви и все метки, кроме метрик текущего пути, были опущены для большей ясности. Буква T под узлом указывает на то, что два пути прибытия имели равные показатели выполнения, и в этом случае мы просто подбрасываем монетку, чтобы выбрать выжившего (почему?). Путь максимального правдоподобия следует за жирной линией от a_0 до a_{12} , а конечное значение метрики пути означает, по

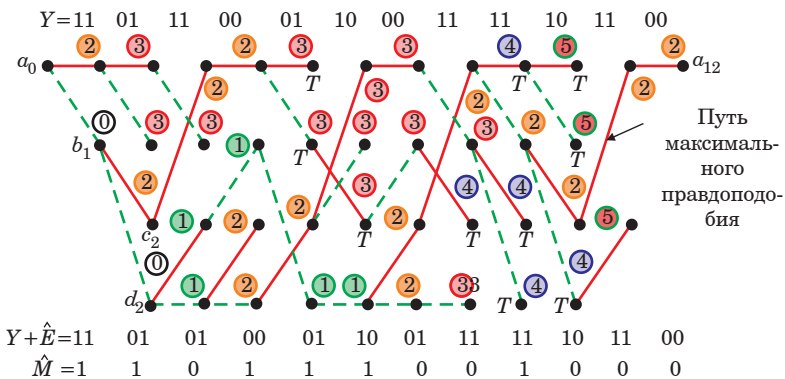


Рис. 3.16. Алгоритм Витерби для декодирования с максимальным правдоподобием



меньшей мере, две ошибки передачи в Y . Декодер принимает соответствующую переданную последовательность $Y+E$ и последовательность сообщений M , записанную под решеткой.

Декодер Витерби должен вычислить две метрики для каждого узла и сохранить 2^{KL} выживших путей, каждый из которых состоит из N ветвей. Следовательно, сложность декодирования возрастает экспоненциально с увеличением L и линейно с увеличением N . Экспоненциальный коэффициент ограничивает практическое применение алгоритма Витерби для кодов с малыми значениями L .

Когда $N \gg 1$, требования к хранилищу могут быть уменьшены с помощью процесса усечения, основанного на следующем эффекте расхождения метрик: если два выживших пути исходят из одного и того же узла в какой-то момент, то текущая метрика менее вероятного пути имеет тенденцию увеличиваться быстрее, чем метрика другого выжившего в течение примерно $5L$ ответвляется от общего узла. Этот эффект несколько раз появляется на рис. 3.16; рассмотрим, например, два пути, исходящие из узла $b1$. Следовательно, декодирование не нужно откладывать до конца передаваемой последовательности. Вместо этого первые k битов сообщения могут быть декодированы, и первый набор ответвлений может быть удален из памяти после обработки первых $5L_n$ принятых битов. Последовательные группы из k битов сообщения затем декодируются для каждого дополнительного n битов, полученных впоследствии.

Последовательное декодирование, которое было изобретено до алгоритма Витерби, также основано на эффекте расхождения метрик. Упрощенная версия последовательного алгоритма проиллюстрирована на рис. 3.17а с использованием той же решетки полученной последовательности и метрик, что и на рис. 3.17. Начиная с a_0 , последовательный декодер следует по единственному пути, выбирая ветвь с наименьшей метрикой ветвления в каждом последующем узле. Если две или более ветвей от одного узла имеют одинаковую метрику, например, в узле d_2 , декодер выбирает одну из них случайным образом и продолжает работу. Всякий раз, когда текущий путь оказывается маловероятным, текущая метрика быстро увеличивается, и декодер в конечном итоге решает вернуться к узлу с более низкой метрикой и попробовать другой путь. В нашем примере есть три таких заброшенных пути. Тем не менее сравнение с рис. 3.17 показывает, что последовательное декодирование требует меньше вычислений, чем декодирование Витерби.



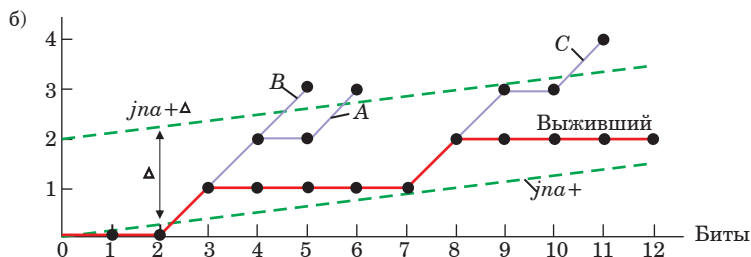
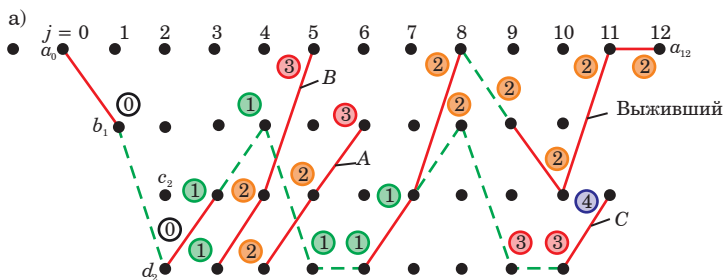


Рис. 3.17. Пояснения к понятию последовательного декодирования: а – последовательный алгоритм; б – текущие показатели алгоритма

Решение вернуться назад и повторить попытку основывается на ожидаемом значении текущей метрики на данном узле. В частности, если α – вероятность ошибки передачи на бит, то ожидаемая метрика выполнения в j -м узле правильного пути равна $j\alpha$, ожидаемому количеству битовых ошибок в Y в этой точке. Последовательный декодер отказывается от пути, когда его метрика превышает некоторый заданный порог Δ выше $j\alpha$. Если ни один путь не выдерживает порогового теста, значение увеличивается, и декодер снова выполняет обратный отсчет. На рис. 3.17б показаны текущие показатели в зависимости от j , а также $j\alpha$ и пороговая линия $j\alpha + \Delta$ для $\alpha = 1/16$ и $\Delta = 2$.

Последовательное декодирование приближается к производительности декодирования с максимальным правдоподобием, когда пороговое значение достаточно свободно, чтобы позволить исследовать все вероятные пути. Однако частое отслеживание назад требует больше вычислений, чем при декодировании по Витерби, и приводит к задержке декодирования. Более жесткий порог уменьшает вычисления и задержку декодирования, но может фактически исключить наиболее вероятный путь, тем самым увеличивая вероятность ошибки на выходе по сравнению с вероятностью декодиро-



вания с максимальным правдоподобием при том же коэффициенте усиления при кодировании. В качестве компенсации последовательное декодирование позволяет практически применять сверточные коды с большим L и большим коэффициентом усиления при кодировании, поскольку сложность декодера по существу не зависит от L .

Мы описали последовательное декодирование и декодирование Витерби в терминах алгоритмов, а не блок-схем аппаратного обеспечения. Действительно, эти методы обычно реализуются в виде программного обеспечения, которое выполняет вычисления метрик и сохраняет данные о пути. Когда обстоятельства исключают алгоритмическое декодирование и более высокая вероятность ошибки допустима, декодирование с обратной связью может быть подходящим методом. *Декодер обратной связи* действует в целом как «декодер скользящего блока», который декодирует биты сообщения один за другим на основе блока из L или более последовательных ветвей дерева. Мы сосредоточимся на специальном классе декодирования с обратной связью, который использует мажоритарную логику для достижения простейшей аппаратной реализации сверточного декодера.

Рассмотрим последовательность сообщений $M=m_1m_2 \dots$ и систематическую $(2, 1, L)$ кодированную последовательность

$$X = x'_1 x''_1 x'_2 x''_2 \dots, \quad (3.35)$$

где $x'_j = m_j$; $x''_j = \sum_{i=0}^{L-1} m_{j-g_i}$ по модулю 2.

Мы будем рассматривать всю последовательность X как кодовое слово неопределенной длины. Затем, заимствуя матричное представление, используемое для блочных кодов, определим порождающую матрицу G и матрицу проверки H такую, что

$$X = MG \quad XH = 0 \ 0 \ \dots \quad (3.36)$$

Чтобы представить уравнение (3.35), G должна быть полубесконечной матрицей с диагональной структурой, заданной

$$G = \begin{bmatrix} 1 & g_0 & 0 & g_1 & \dots & 0 & g_L \\ & 1 & g_0 & 0 & g_1 & \dots & 0 & g_L \\ & & \cdot & \cdot & \cdot & & \cdot & \\ & & & \cdot & \cdot & \cdot & & \cdot \\ & & & & \cdot & \cdot & & \cdot \\ & & & & & \cdot & & \cdot \end{bmatrix}. \quad (3.37)$$



4. ОСНОВЫ ТЕОРИИ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Физический радиосигнал, являющийся непрерывной функцией от времени, называется аналоговым сигналом $s(t)$. Сигнал, формирующийся при помощи цифровой обработки, называется дискретным $s(n)$. Дискретный сигнал (рис. 4.1), в отличие от аналогового, не может существовать на всем времени наблюдения. В связи с этим он представляется в определенные (дискретные) моменты времени, которые называются отсчетами. Отсчеты выбираются через равные промежутки времени T_D , которые в свою очередь называются периодом дискретизации. Величина, обратная периоду дискретизации, называется частотой дискретизации $f_D=1/T_D$. Процесс преобразования аналогового сигнала в дискретный называется дискретизацией.

Вычислительные устройства, находящиеся в радиотехнических системах, оперируют дискретными сигналами, отсчеты которых представляются в виде двоичных чисел, а такой процесс называется квантованием по уровню. Устройства, осуществляющие преобразование аналогового радиосигнала в цифровой, называются аналого-цифровым преобразователем (АЦП). В связи с тем, что разрядность таких устройств ограничена, отсчеты могут принимать только ограниченное количество значений. Таким образом, ограниченное число отсчетов приводит к округлению обрабатываемого сигнала. Ошибки, возникающие вследствие такого округления, называются ошибками квантования. Радиосигнал, представленный в виде дискретных отсчетов и квантованный по уровню, называется цифровым. В настоящем учебном пособии проектируемое радиоприемное устройство не включает в себя аналого-цифро-

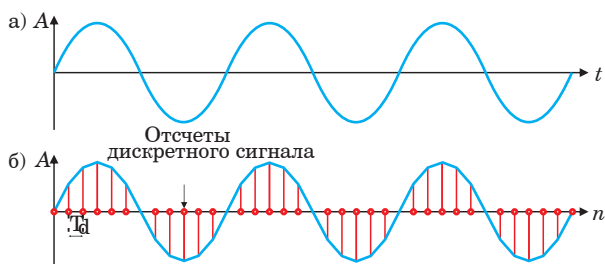


Рис. 4.1. Радиосигнал: а – аналоговый; б – дискретный



вой преобразователь, в связи с чем цифровые сигналы не рассматриваются.

Очевидно, что дискретизация сигнала, в связи с тем, что неизвестно значение сигнала между дискретными отсчетами, может привести к потере информации. Для обеспечения минимизации потерянной информации необходимо познакомиться с теоремой Котельникова.

4.1. Теорема Котельникова

Теорема Котельникова гласит о том, что любой радиосигнал $s(t)$, спектр которого не содержит синусоид с частотой выше некоторого значения f_B , может быть без потерь представлен с помощью дискретных отчетов с интервалом T_D , удовлетворяющему неравенству

$$T_D \leq \frac{1}{2f_B}, \quad (4.1)$$

где f_B – частота синусоиды с максимальной частотой в составе радиосигнала; T_D – период дискретизации.

Выбор такого периода обуславливается тем, что в случае несоблюдения условия теоремы Котельникова дискретный сигнал может не соответствовать исходному аналоговому сигналу, что в свою очередь приводит к потере информации. Соблюдая выражение

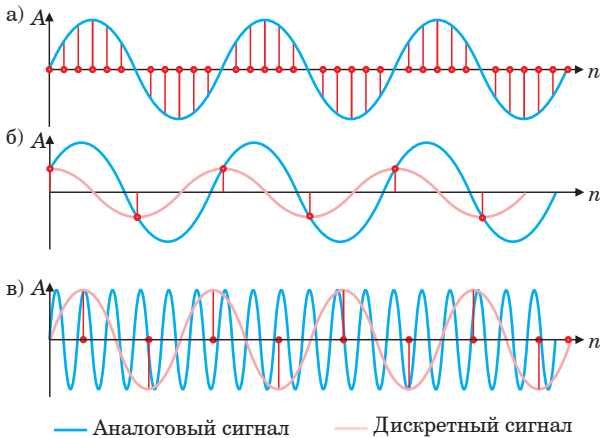



Рис. 4.2. Дискретный сигнал: а – несущая частота меньше чем в два раза частоты дискретизации; б – несущая частота равна частоте дискретизации; в – несущая частота больше частоты дискретизации



(4.1) (в случае, если период дискретизации соответствует Теореме Котельникова), аналоговый сигнал правильно восстанавливается в процессе дискретизации (рис. 4.2а). Однако в случае, если частота f_B равна частоте дискретизации (рис. 4.2б) или больше частоты дискретизации (рис. 4.2в), то амплитуда, фаза и частота радиосигнала могут быть искажены и сообщение принято с ошибками.

*Пример формирования дискретного радиосигнала
в пакете прикладных программ Matlab*

Для демонстрации необходимости соблюдения условия теоремы Котельникова сформируем тестовые сигналы для различных периодов дискретизации в пакете прикладных программ Matlab. Для создания проекта «Script» в Matlab необходимо в верхнем левом углу нажать на виртуальную кнопку  или сочетание клавиш «Ctrl+N». В первую очередь в Matlab задаются переменные. Переменные, необходимые для формирования простого синусоидального тестового сигнала, были заданы исходя из выражения

$$s(t_n) = A \sin(\omega_0 t_n), \quad (4.2)$$

где A – амплитуда радиосигнала; $\omega_0 = 2\pi f$ – угловая частота; $t_n = 0:T_D:T$ – вектор дискретного времени; $T_D = \frac{1}{f_D}$ – период дискретизации; f_D – частота дискретизации; T – интервал наблюдения.

Программный код для формирования тестовых радиосигналов с различным периодом дискретизации:

```
clc
clear all
close all
a=1; %Амплитуда радиосигнала
f=1; %Частота радиосигнала
fd=[ f 3*f 10*f]; %Вектор, содержащий различные значения частоты дискретизации
Td=1./fd; %Вектор, содержащий различные значения периода дискретизации
t1=0:Td(1):2; % Дискретный временной интервал для периода дискретизации 1/f
t2=0:Td(2):2; % Дискретный временной интервал для периода дискретизации 1/3*f
t3=0:Td(3):2; % Дискретный временной интервал для периода дискретизации 1/10*f
s1=a*sin(2*pi*f*t1);% Дискретный сигнал с периодом дискретизации 1/f
s2=a*sin(2*pi*f*t2);% Дискретный сигнал с периодом дискретизации 1/3*f
```



```

s3=a*sin(2*pi*f*t3);% Дискретный сигнал с периодом дискретизации
1/10*f
subplot(3,1,1) % Формирование окна для трех графиков и указание на
место расположения
plot(t1,s1) % Построение графика для дискретно сигнала с периодом
дискретизации 1/f
xlabel('Дискретные отсчеты времени')
ylabel('Амплитуда сигнала')
legend('Период дискретизации 1/f')
subplot(3,1,2) % Формирование окна для трех графиков и указание на
место расположения
plot(t2,s2) % Построение графика для дискретно сигнала с периодом
дискретизации 1/3*f
xlabel('Дискретные отсчеты времени')
ylabel('Амплитуда сигнала')
legend('Период дискретизации 1/3*f')
subplot(3,1,3) % Формирование окна для трех графиков и указание на
место расположения
plot(t3,s3) % Построение графика для дискретно сигнала с периодом
дискретизации 1/10*f
xlabel('Дискретные отсчеты времени')
ylabel('Амплитуда сигнала')
legend('Период дискретизации 1/10*f')

```

Таким образом, как видно из программного кода в первую очередь были заданы параметры, входящие в выражение (4.2). Стоит отметить, что в представленном программном коде частота дискретизации f_d задана в виде вектора строки. Это сделано для того, чтобы задать различные периоды дискретизации T_d и не «плодить» большое количество переменных, дабы не запутаться в программном коде. В соответствии с тем, что f_d является вектором строкой, перед алгебраическими операциями над ним ставится «.» – это говорит программе о том, что операцию нужно производить над каждым элементом вектора. Далее в строчках, где задаются значения для t_1 , t_2 и t_3 , выбирается только одно значения вектора строки, содержащего периоды дискретизации T_d , с помощью индекса, указанного в скобках после указания переменной T_d .

Определив все переменные и математические выражения для них, необходимо перейти к анализу полученного результата путем построения графиков. В данном примере перед началом построения графиков для удобства анализа необходимо было вывести все графики в одно графическое окно. Реализовать данное требование возможно с помощью программной функции, разбивающей графическое окно на подокна – `subplot(m, n, p)`, где m – количество подокон по вертикали; n – количество подокон по горизонтали; p –



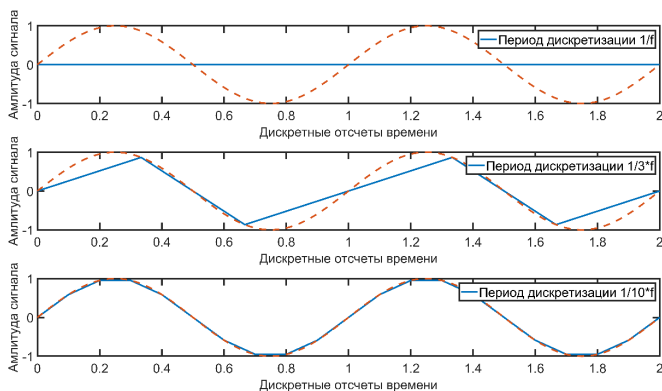


Рис. 4.3. Дискретный радиосигнал с различным периодом дискретизации

номер подокна. Выбрав необходимое подокно функцией `subplot`, можно приступить к построению графика. В Matlab, для построения графика зависимости переменной y от переменной x , используется функция `plot(x,y)`.

После формирования программного кода его нужно запустить, нажав виртуальную кнопку «Run» ► или нажав клавишу «F5» на клавиатуре.

На рис. 4.3 представлен результирующий график, рассчитанный с использованием программного кода для формирования тестовых радиосигналов. На верхнем графике рисунка видно, что, не соблюдая условие теоремы Котельникова, дискретный сигнал равен нулю на всем интервале наблюдения. Связано это с тем, что частота дискретизации равна частоте сигнала, а начальная фаза – нулю, то есть отсчеты сигнала на временной оси расположены там, где амплитуда исходного сигнала равно нулю. На среднем графике рисунка период дискретизации был задан с использованием теоремы Котельникова, и спектр такого сигнала можно восстановить без потери информации, однако видно, что есть несовпадения по амплитуде. При формировании нижнего графика рисунка период дискретизации был выбран в пять раз меньше требуемого в условии теоремы Котельникова. Таким образом, как видно из графика, исходный и дискретный сигналы полностью совпали.

4.2. Спектр сигнала

Как говорилось, ранее радиосигнал может быть представлен во временной и частотной областях (рис. 4.4). Представление сигнала



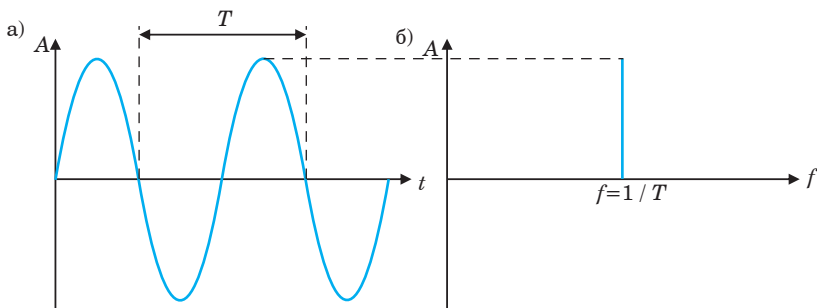


Рис. 4.4. Представление радиосигнала в частотной области:
 а – радиосигнал во временной области; б – радиосигнал в частотной области

в частотной области называется его спектром (рис. 4.4б). Спектр сигнала показывает амплитуду и частоту гармонических колебаний, из которых он состоит. Для полного понимания необходимо рассмотреть понятие «ряд Фурье».

4.3. Ряд Фурье

Теорема Фурье гласит о том, что любой периодический сигнал можно представить суммой синусных и косинусных гармонических сигналов с различными амплитудами и частотами с общим делителем (рис. 4.5). Такое разложение сигнала на слагаемые называется рядом Фурье.

Существует три основные формы записи ряда Фурье: синусно-косинусная, вещественная и комплексная форма. В синусно-косинусной форме записи ряд Фурье представляется в соответствии с выражением

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(k\omega_0 t) + b_n \sin(k\omega_0 t)), \quad (4.3)$$

где $\omega_0 = \frac{2\pi}{T}$ – круговая частота, соответствующая периоду сигнала; n – номер гармоники, входящей в состав радиосигнала; $k\omega_0$ – частота n -й гармоники; $T_D = \frac{1}{f_D}$ – коэффициент ряда Фурье; f_D – коэффициент ряда Фурье; T – постоянная составляющая.

Вещественная форма ряда Фурье является более удобной формой записи, поскольку в ней с помощью тригонометрических пре-



образований избавились от суммы синусной и косинусной составляющей. Таким образом, сумма была заменена косинусом с амплитудой и начальной фазой n -й гармоники

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} A_n \cos(k\omega_0 t + \varphi_n), \quad (4.4)$$

где $A_n = \sqrt{a_n^2 + b_n^2}$ – n -я амплитуда гармоники;

$$\varphi_n = \begin{cases} -\text{artctg} \frac{b_n}{a_n}, & a_n \geq 0 \\ -\text{arctg} \frac{b_n}{a_n} \pm \pi, & a_n < 0 \end{cases} \quad \text{– } n\text{-я начальная фаза гармоники.}$$

Поскольку наиболее часто для удобства расчетов радиосигналы представляют в комплексном виде, комплексная форма ряда Фурье чаще всего используется в радиотехнических задачах. Она образуется с применением формулы Эйлера к вещественной форме и в итоге определяется формулой

$$s(t) = \sum_{n=-\infty}^{\infty} \dot{C}_n e^{-jn\omega_0 t}, \quad (4.5)$$

где $\dot{C}_n = \frac{1}{2} A_n e^{j\varphi_n}$ – комплексный коэффициент ряда Фурье.

Поскольку коэффициент \dot{C}_n является комплексным, как уже говорилось ранее, амплитуда является модулем функции, а фаза аргументом. Тогда коэффициенты ряда Фурье для комплексной формы будут определяться выражениями

$$\begin{aligned} A_n &= 2|\dot{C}_n|, \\ b_n &= \arg(\dot{C}_n). \end{aligned} \quad (4.6)$$

В общем случае коэффициент \dot{C}_n определяется выражением

$$\dot{C}_n = \frac{1}{T} \int_0^T s(t) e^{-jn\omega_0 t} dt. \quad (4.7)$$

Таким образом, понимая понятие ряда Фурье и определив все формулы преобразования сигнала в ряд Фурье, становится понятно происхождение амплитуд гармоник радиосигнала в частотной области. На рис. 4.6 представлен спектр периодического прямоу-



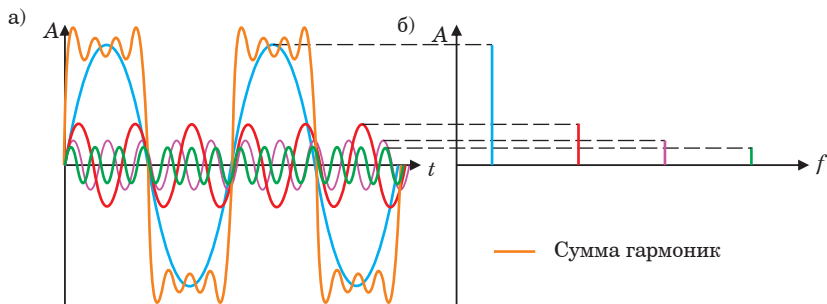


Рис. 4.6. Спектр прямоугольного радиосигнала:
 а – прямоугольный радиосигнал, состоящие из суммы четырех гармоник;
 б – спектр прямоугольного радиосигнала

гольного радиосигнала. Отсюда видно, что спектр состоит из кратных частот. Такой спектр называется линейчатый.

4.4. Преобразование Фурье

Преобразование Фурье является инструментом спектрального анализа сигналов. Его используют для преобразования сигнала из временной области в частотную. В отличие от ряда Фурье, его используют как для периодических сигналов, так и для непериодических. Однако в случае непериодических сигналов спектр уже не линейчатый, а сплошной. Преобразование Фурье тесно связано с выражением (4.7), с теми лишь отличиями, что кратность частот $n\omega_0$ заменяется непрерывной ω , функция перестает быть ограничена коэффициентом $1/T$, а результатом преобразования теперь является не коэффициент ряда, а спектральная плотность $\dot{S}(\omega)$. Так, итоговое выражение для преобразования Фурье будет в виде

$$\dot{S}(\omega) = F\{s(t)\} = \int_{-\infty}^{\infty} s(t)e^{-j\omega t} dt, \quad (4.8)$$

где $F\{s(t)\}$ – преобразование Фурье от сигнала $s(t)$.

Также можно и преобразовать сигнал из частотной области во временную, такое преобразование называется обратным преобразованием Фурье и определяется выражением

$$s(t) = F^{-1}\{\dot{S}(\omega)\} = \int_{-\infty}^{\infty} \dot{S}(\omega)e^{j\omega t} d\omega, \quad (4.9)$$



где $F^{-1}\{s(t)\}$ – обратное преобразование Фурье от спектральной плоскости $\dot{S}(\omega)$.

Дискретное преобразование Фурье

Дискретное преобразование Фурье (DFT) предназначено для преобразования дискретного сигнала из временной области $s(t_n)$ в частотную $\dot{S}(\omega_n)$. В связи с тем, что в цифровой обработке сигналов, анализу в основном подвергаются дискретные сигналы, то такое преобразование является основным инструментом. Дискретное преобразование Фурье определяется выражением

$$\dot{S}(k) = F\{s(t_n)\} = \sum_{k=0}^{N-1} s(t_n) e^{-j\frac{2\pi t_n k}{N}} = \sum_{k=0}^{N-1} s(t_n) e^{-j\frac{2\pi t_n k}{N}}, \quad (4.10)$$

где $F\{s(t_n)\}$ – дискретное преобразование Фурье от сигнала $s(t_n)$; $\dot{S}(k)$ – k комплексных амплитуд гармоник k -й частоты в частотной области; t_n – дискретные отсчеты времени; N – количество дискретных отсчетов; k – индекс частоты.

Обратное дискретное преобразование Фурье отличается от прямого наличием множителя $1/N$ и соответствует выражению

$$s(t_n) = F^{-1}\{\dot{S}(k)\} = \frac{1}{N} \sum_{k=0}^{N-1} \dot{S}(k) e^{j\frac{2\pi t_n k}{N}} = \sum_{k=0}^{N-1} s(t_n) e^{-j\frac{2\pi t_n k}{N}}, \quad (4.11)$$

где $F^{-1}\{\dot{S}(k)\}$ – обратное дискретное преобразование Фурье от спектральной плоскости $\dot{S}(\omega)$.

Преобразование дискретного сигнала из временной области в частотную требовательно к вычислительной мощности и затрачивает определенное время. Связано это с тем, что для вычисления одной частотной составляющей по формуле (4.10) необходимо выполнить N комплексных умножений и сложений. Естественно, увеличивая количество отсчетов N растет число операций, что может привести к невозможности устройства обработать сигнал или к большой задержке, что может быть критично, когда спектр нужно наблюдать в «реальном» времени или близкому к нему. Для упрощения процесса вычислений дискретного преобразования Фурье используют его частный случай – быстрое преобразование Фурье (FFT). Быстрое преобразование Фурье является алгоритмом ускоренного вычисления дискретного преобразования Фурье. В связи тем, что настоящее пособие носит больше практический, чем теоре-



тический характер, теория и процесс алгоритмов быстрого преобразования Фурье рассмотрена не будет. Однако рассмотрим практическое применение FFT.

Пример преобразования дискретного радиосигнала из временной области в частотную в пакете прикладных программ Matlab

Для примера сформируем графические подокна в среде Matlab, соответствующие графикам сигнала во временной и частотной области.

Программный код, формирующий четыре синусоиды с различными амплитудами и кратными частотами, их сумму и спектр суммы, прямоугольный радиоимпульс и его спектр:

```
clc
clear all
close all
%Исходная частота
f=1;
%Частота дискретизации
Fc=1e3;
%Период дискретизации
Td=1/(Fc);
%Количество дискретных отсчетов для FFT
N=4096;
%Дискретные отсчеты времени
t=0:Td:4;
%Формирование сигнала, состоящего из суммы четырех гармоник с различными амплитудами и частотами кратными f
s= 10*sin(2*pi*1*t)...
   +3*sin(2*pi*3*f*t)...
   +2*sin(2*pi*5*f*t)...
   +1*sin(2*pi*7*f*t);
%Выполнение быстрого преобразования Фурье сигнала s
S=abs(fft(s));
%Нормировка спектра по амплитуде
S=2*S./N;
%Формирование вектор дискретных отсчетов по частоте
fx=(0:N-1)*Fc/N;
%Ограничение дискретных отсчетов по частоте для построения графика
fx=fx(1:length(t));
%длина прямоугольного импульса
tao = 1;
%Максимальное время наблюдения
etl = 2;
%частота повторения импульса
Fs = 1e3;
%амплитуда прямоугольного импульса
```



```

A = 5;
%Дискретные отсчеты времени
t1 = (0:1/Fc:et1);
%Длительность прямоугольного импульса;
"+ tao/2" - это сдвиг самого импульса, т.к. rectpuls считается от
середины (центрирован)
d = (0:Ffs:et1) + tao/2;
%Формирование прямоугольного импульса
x = A.*pulstran(t1, d, @rectpuls, tao);
%Выполнение быстрого преобразования Фурье
S1=abs(fft(x));
%Нормировка спектра по амплитуде
S1=2*S1./N;
%Формирование вектор дискретных отсчетов по частоте
fx1=(0:N-1)*Fc/N;
%Ограничение дискретных отсчетов по частоте для построения графика
fx1=fx1(1:length(t1));
figure
subplot(3,2,1)
%Построение графика для 4 гармоник
plot(t,10*sin(2*pi*1*t),...
      t,3*sin(2*pi*3*t),...
      t,2*sin(2*pi*5*t),...
      t,sin(2*pi*7*t))

ylabel('Амплитуда')
xlabel('Время')
legend('Амплитуда = 10, Частота = 1 Гц', 'Амплитуда = 3,
Частота = 3 Гц', 'Амплитуда = 2, Частота = 7 Гц')
subplot(3,2,3)
%Построение графика для суммы 4 гармоник
plot(t,s)
ylabel('Амплитуда')
xlabel('Время')
legend('Сумма гармоник')
subplot(3,2,5)
%Построение спектра для сигнала, состоящего из суммы 4 гармоник
plot(fx,S)
ylabel('Амплитуда')
title('Масштабированный спектр суммы гармоник')
xlabel('Частота')
subplot(3,2,6)
%Построение спектра для сигнала, состоящего из суммы 4 гармоник
plot(fx,S)
ylabel('Амплитуда')
title('Спектр суммы гармоник')
xlabel('Частота')
subplot(3,2,2)
% Построение графика для прямоугольного импульса

```



```

plot(t1, x)
ylabel('Амплитуда')
xlabel('Время')
legend('Прямоугольный радиоимпульс')
subplot(3,2,4)
%Построение спектра для прямоугольного импульса
plot(fx1,S1)
ylabel('Амплитуда')
xlabel('Частота')
legend('Масштабированный спектр прямоугольного радиоимпульса')

```

Как видно, в начале программного кода задается частота f , к которой формируются кратные частоты (для примера выбрана частота $f = 1$ Гц). По аналогии с программным кодом, формирующим четыре синусоиды, необходимо было в первую очередь определить все переменные в начале кода. Таким образом, в соответствии с выражением (4.10) для выражения дискретного преобразования Фурье был сформирован радиосигнал во временной области $s(t_n)$, определено количество дискретных отсчетов N . Как видно из программного кода, количество дискретных отсчетов N для преобразования Фурье и для вектора времени t_n имеют разное значение. Такая особенность вызвана тем, что к программному коду было выдвинуто требование (вызванное необходимостью удобства) – задавать время наблюдения конкретным значением вместо того, чтобы определять его следующим выражением: $[0:N-1] \cdot T_d$. Однако стоит понимать, что при загрузке сигнала в векторный генератор произвольных форм он ждет его определенной длительностью и с определенным количеством отсчетов N (sample time). Быстрое преобразование Фурье реализуется с помощью функции `fft`. Стоит понимать, что в соответствии с выражением (4.10) результатом дискретного преобразования Фурье является комплексная величина. В связи с этим, для получения амплитудного спектра необходимо взять модуль (функция `abs`) от результата, получившегося после `fft`. Также для сравнения был построен прямоугольный радиоимпульс с помощью функции `pulsetran` (рис. 4.7г). Функция `pulsetran` генерирует последовательность импульсов с произвольной длительностью, периодом следования и выбранным типом сигнала, например, могут быть сгенерированы прямоугольные импульсы, треугольные и др. На рис. 4.7д представлен спектр прямоугольного импульса. Как было выяснено ранее, чем больше кратных гармоник сложено, тем прямоугольный импульс более гладкий. Так, из спектра прямоугольного импульса видно, что количество гармоник в разы больше чем в спектре на рис. 4.7в.



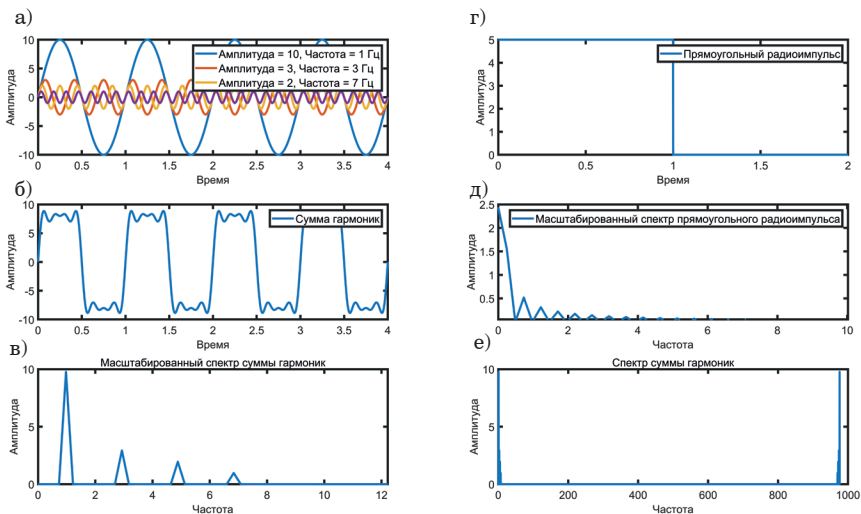


Рис. 4.7. Формирование спектра в пакете прикладных программ Matlab:
 а – гармоники с различными амплитудами и кратными частотами;
 б – сумма гармоник рис. 4.9а; в – масштабированный спектр суммы гармоник;
 г – прямоугольный радиоимпульс; д – спектр прямоугольного радиоимпульса;
 е – спектр суммы гармоник

В соответствии с рис. 4.7а, б, в, на которых представлен результат работы программного кода, видно, что результат представленного кода совпадает с рис. 4.6. Стоит отметить, что в результате выполнения представленной программы, получается амплитудный спектр, представленный на рис. 4.9е. Это связано с тем, что спектр дискретного сигнала периодичен с периодом равным частоте дискретизации. Восстановить спектр можно с помощью функции `fftshift`, как это видно из рис. 4.8. Поскольку нас интересует только положительная область функции `fftshift`, масштабирование производилось ручным способом.

Также стоит обратить внимание на разницу между спектром сигнала, рассчитанным с помощью математического выражения для дискретного преобразования Фурье и с помощью функции `fft`. На рис. 4.9б показан спектр радиосигнала с рисунка 4.9а, рассчитанный с использованием формулы (4.10), а на рис. 4.9 в спектр сигнала, рассчитанный с помощью функции `fft`. Как видно, спектр, рассчитанный с использованием математического выражения, не изменяется. Однако спектр, рассчитанный с помощью функции `fft`, зависит от исходных параметров. Так, на рис. 4.9а представлен



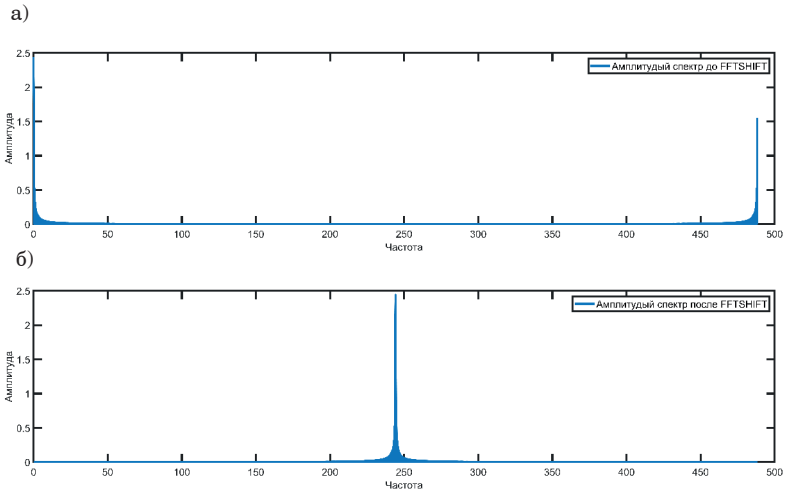


Рис. 4.8. Амплитудный спектр сигнала:
 а – до применения функции FFTSHIFT;
 б – после применения функции FFTSHIFT

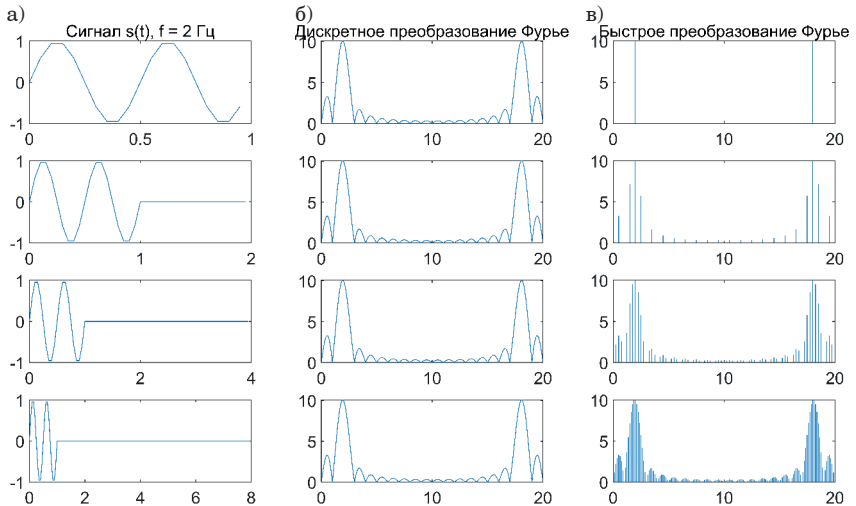


Рис. 4.9. Разница между дискретным и быстрым преобразованием Фурье:
 а – сигнал во временной области с различным количеством дискретных отсчетов N ; б – результат дискретного преобразования Фурье;
 в – результат быстрого преобразования Фурье



сигнал, у которого не изменяется длительность и частота, но изменяется интервал дискретных отсчетов времени, то есть их количество N . Быстрое преобразование Фурье в Matlab предполагает, что принятая последовательность является периодом сигнала T_c . Расстояние между составляющими в спектре Δf определяется исходя из выражения

$$\Delta f = \frac{1}{T_c}. \quad (4.12)$$

На рис. 4.9 в видно, что с увеличением дискретных отсчетов, на которых сигнал не существует, уменьшается расстояние между составляющими Δf и увеличивается их количество с увеличением N .

На рис. 4.10–4.13 представлены сигналы во временной и частотной областях, а также программные коды для их формирования, с амплитудной, частотной, линейной частотной и фазовой модуляцией. Как видно из рисунков, непериодические сигналы имеют непрерывный амплитудный спектр.

Программный код, формирующий сигнал с амплитудной модуляцией:

```

clc
clear all
close all
fn = 1e1; % несущая частота, Гц
fc = 1; % частота сигнала, Гц
Fs = 1e3; % частота дискретизации, Гц
N=4096;% Количество дискретных отсчетов для FFT
t = 0:1/(Fs):4;% Дискретные отсчеты времени
m = 0.5; % индекс модуляции
s_am = sin(2*pi*fn*t) .* (1 + m * sin(2*pi*fc*t));% сигнал с АМ
figure
S=abs(fft(s_am));% Выполнение быстрого преобразования Фурье сигнала s_am
S=2*S./N;% Нормировка спектра по амплитуде
fx=(0:N-1)*Fs/N;% Формирование вектор дискретных отсчетов по частоте
fx=fx(1:length(t));% Ограничение дискретных отсчетов по частоте для построения графика
subplot(2,2,2)
plot(fx,S)
title('Спектр АМ сигнала')
xlabel('Частота')
ylabel('Амплитуда')
subplot(2,2,4)
plot(t,s_lcm)

```



```
title('АМ сигнал')
xlabel('Время')
```

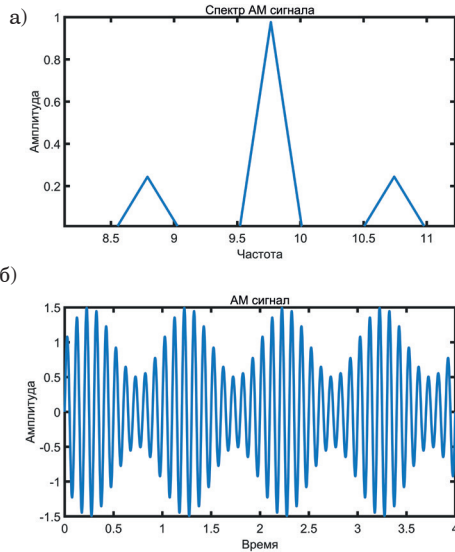


Рис. 4.10. Сигнал с амплитудной модуляцией: а – спектр сигнала с амплитудной модуляцией; б – сигнал с амплитудной модуляцией во временной области

Программный код, формирующий сигнал с частотной модуляцией:

```
clc
clear all
close all
fn = 1e1; % несущая частота, Гц
fc = 1; % частота сигнала, Гц
Fs = 1e3; % частота дискретизации, Гц
N=4096;% Количество дискретных отсчетов для FFT
t = 0:1/(Fs):4;% Дискретные отсчеты времени
s_cm = sin(2*pi*fc*(cos(4*t)).*t); % сигнал ЧМ
figure
S=abs(fft(s_cm));% Выполнение быстрого преобразования Фурье сигнала s_cm
S=2*S./N;% Нормировка спектра по амплитуде
fx=(0:N-1)*Fs/N;% Формирование вектор дискретных отсчетов по частоте
fx=fx(1:length(t));% Ограничение дискретных отсчетов по частоте для построения графика
subplot(2,2,2)
plot(fx,S)
title('Спектр ЧМ сигнала')
xlabel('Частота')
```



```

ylabel('Амплитуда')
subplot(2,2,4)
plot(t,s_cm)
title('ЧМ сигнал')
xlabel('Время')
ylabel('Амплитуда')

```

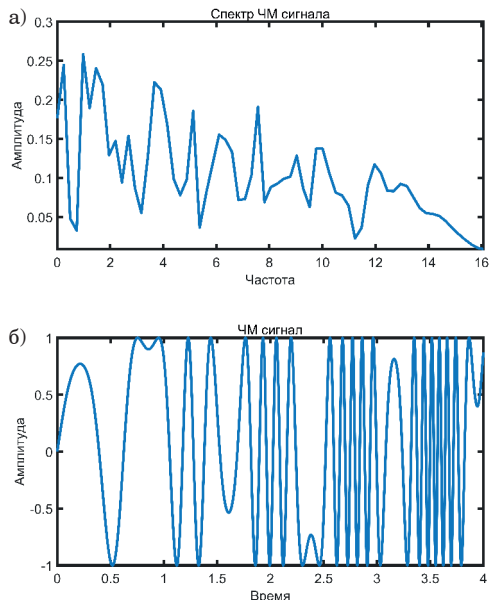


Рис. 4.11. Сигнал с частотной модуляцией: а – спектр сигнала с частотной модуляцией; б – сигнал с частотной модуляцией во временной области

Программный код, формирующий сигнал с линейной частотной модуляцией:

```

clc
clear all
close all
fn = 1e1; % несущая частота, Гц
fc = 1; % частота сигнала, Гц
Fs = 1e3; % частота дискретизации, Гц
N=4096;% Количество дискретных отсчетов для FFT
t = 0:1/(Fs):4;% Дискретные отсчеты времени
s_lcm = sin(2*pi*t.^2); % сигнал с ЛЧМ
figure
S=abs(fft(s_lcm));% Выполнение быстрого преобразования Фурье сигнала s_lcm
S=2*S./N;% Нормировка спектра по амплитуде

```



```

fx=(0:N-1)*Fs/N;% Формирование вектор дискретных отсчетов по частоте
fx=fx(1:length(t));% Ограничение дискретных отсчетов по частоте
для построения графика
subplot(2,2,2)
plot(fx,S)
title('Спектр ЛЧМ сигнала')
xlabel('Частота')
ylabel('Амплитуда')
subplot(2,2,4)
plot(t,s_lcm)
title('ЛЧМ сигнал')
xlabel('Время')
ylabel('Амплитуда')

```

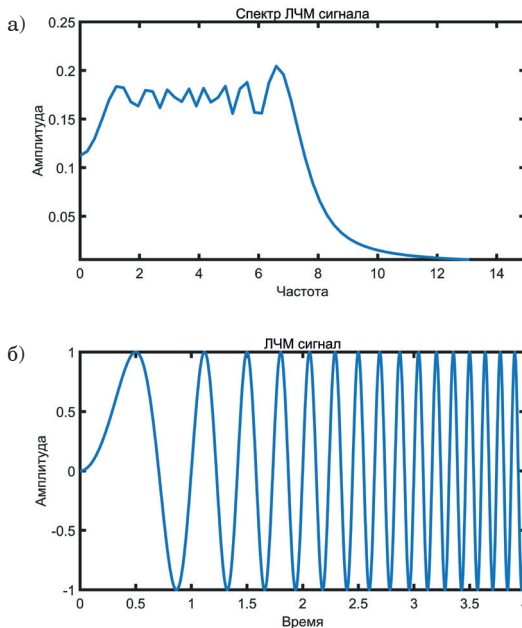


Рис. 4.12. Сигнал с линейной частотной модуляцией:
а – спектр сигнала с линейной частотной модуляцией;
б – сигнал с линейной частотной модуляцией во временной области

Программный код, формирующий сигнал с фазовой модуляцией:

```

clc
clear all
close all
fn = 1e1; % несущая частота, Гц

```



```

fc = 1; % частота сигнала, Гц
Fs = 1e3; % частота дискретизации, Гц
N=4096;% Количество дискретных отсчетов для FFT
t = 0:1/(Fs):4;% Дискретные отсчеты времени
s_fm = sin(2*pi*fn*t + 5*cos(2*pi*t)); % сигнал ФМ
figure
S=abs(fft(s_fm));% Выполнение быстрого преобразования Фурье сигнала s_fm
S=2*S./N;% Нормировка спектра по амплитуде
fx=(0:N-1)*Fs/N;% Формирование вектор дискретных отсчетов по частоте
fx=fx(1:length(t));% Ограничение дискретных отсчетов по частоте для построения графика
subplot(2,2,2)
plot(fx,S)
title('Спектр ФМ сигнала')
xlabel('Частота')
ylabel('Амплитуда')
subplot(2,2,4)
plot(t,s_fm)
title('ФМ сигнал')
xlabel('Время')
ylabel('Амплитуда')

```

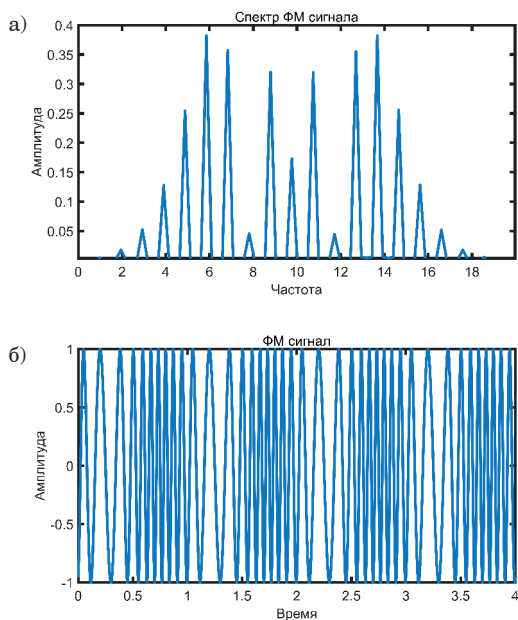


Рис. 4.13. Сигнал с фазовой модуляцией: а – спектр сигнала с фазовой модуляцией; б – сигнал с фазовой модуляцией во временной области



5. МОДУЛЯЦИЯ

Целью модуляции является передача информации с помощью аналогового сигнала, параметры которого соответствуют каналу передачи и аналоговой части цифровой системы приема радиосигналов. На практике существует некоторое количество причин, являющихся преимуществом для использования тех или иных методов модуляции сигнала.

Одной из причин использования модуляции является передача информации на большие расстояния. Эффективность передачи информации на расстояние N с использованием конкретного способа передачи зависит от несущей частоты радиосигнала. Так, зачастую информацию в радиосигнале закладывает в его огибающую (рис. 5.1а), которая может иметь частотное заполнение (несущую) соответствующую частоте выбранной для передачи информации (рис. 5.1б). Пример реализации приведен на рис. 5.2.

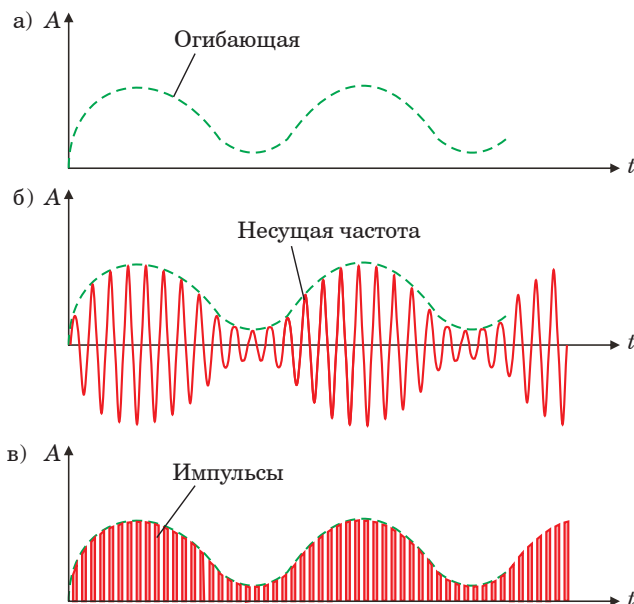


Рис. 5.1. Пояснение к понятию модуляция: а – модулирующий сигнал (огибающая); б – синусоидальный сигнал (несущая) с амплитудной модуляцией; в – импульсы, модулированные по амплитуде



Программный код, формирующий сигнал с амплитудной модуляцией:

```
clc
clear all
close all
fn = 1e1; % несущая частота, Гц
fc = 1; % частота сигнала, Гц
Fs = 1e3; % частота дискретизации, Гц
N=4096;% Количество дискретных отсчетов для FFT
t = 0:1/(Fs):4;% Дискретные отсчеты времени
m = 0.5; % индекс модуляции
s_am = sin(2*pi*fn*t) .* (1 + m * sin(2*pi*fc*t));% сигнал с AM
figure
S=abs(fft(s_am));% Выполнение быстрого преобразования Фурье сигнала s_am
S=2*S./N;% Нормировка спектра по амплитуде
fx=(0:N-1)*Fs/N;% Формирование вектор дискретных отсчетов по частоте
fx=fx(1:length(t));% Ограничение дискретных отсчетов по частоте для построения графика
subplot(2,2,2)
plot(fx,S)
title('Спектр AM сигнала')
xlabel('Частота')
ylabel('Амплитуда')
subplot(2,2,4)
plot(t,s_lcm)
title('AM сигнал')
xlabel('Время')
```

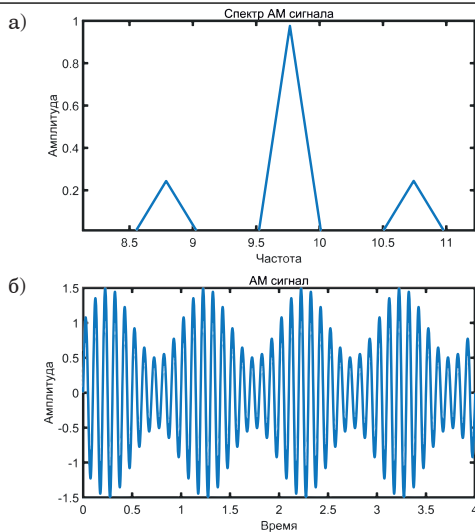


Рис. 5.2. Сигнал с амплитудной модуляцией: а – спектр сигнала с амплитудной модуляцией; б – сигнал с амплитудной модуляцией во временной области



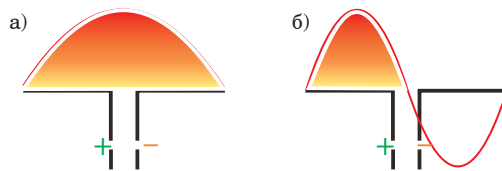


Рис. 5.3. Пояснение к выбору размера антенны: а – размер антенны l равен половине длины волны λ ; б – размер антенны l равен длине волны λ

Выбор частоты также обуславливает особенности оборудования для приема и передачи информации с использованием радиосигнала. Так, для эффективной передачи информации с использованием радиосигнала вдоль линии прямой видимости требуются антенны, физические размеры которых составляют половину длины волны $0,5\lambda$, что обуславливается необходимостью формирования в центре антенны максимума диаграммы направленности (рис. 5.3).

Передача звукового сообщения с частотой, например, 300 Гц в случае с немодулированным радиосигналом потребовалось бы использовать антенну с размером в 500 км. Тогда как при модулированной передаче на частоте, например, в 100 МГц (на этой частоте как раз работает современное ФМ радио), размер антенны получается всего 1,5 м.

Для общего понимания вопроса распространения радиоволн на рис. 5.4 приведено разделение частотных диапазонов по классам, что соответствует регламенту международного союза электросвязи и Российскому ГОСТ 24375-80.

Следующим преимуществом использования модуляции для передачи информации с помощью радиосигналов является обеспечение преодоления аппаратных и регламентных ограничений. Проектирование цифровой системы обмена информацией зачастую подвержено ограничениям связанными со стоимостью и доступностью аппаратных средств, а также выделенным диапазоном. Однако использование модуляции позволяет разработчику переместить передаваемый радиосигнал в тот частотный диапазон, который позволяет обойти аппаратные ограничения.

Стоит понимать, что для сигналов, предназначенных для использования в цифровых системах с большой полосой пропуска-



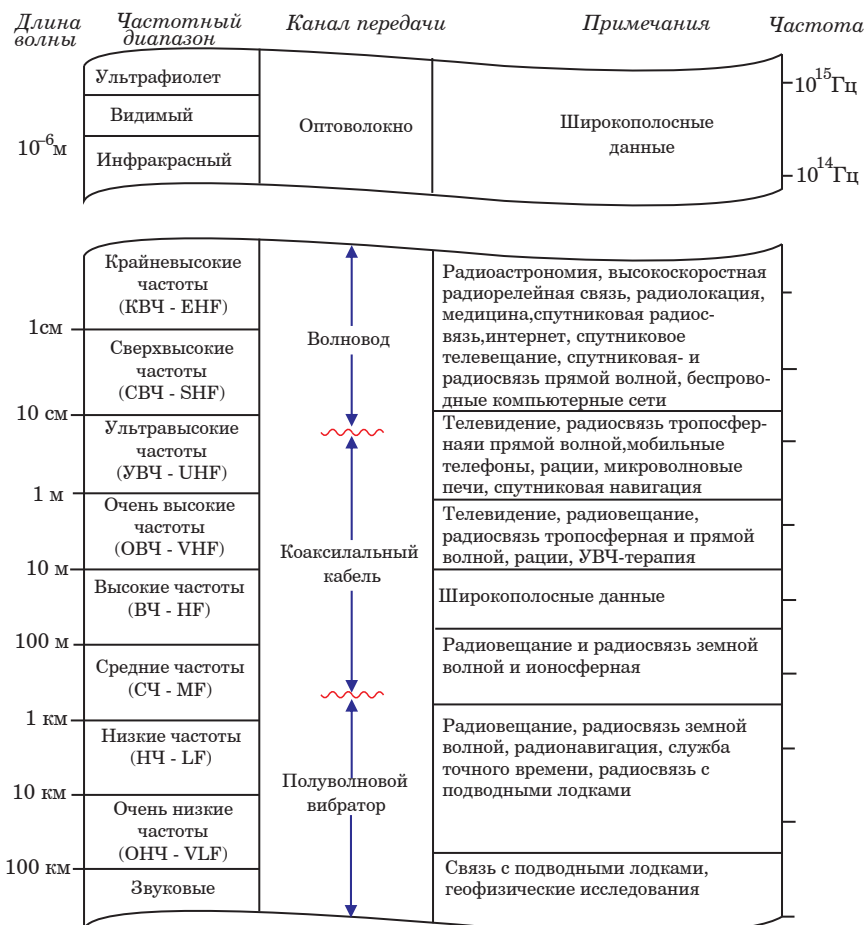


Рис. 5.4. Частотные диапазоны

ния, необходимо использовать модуляцию с высокочастотной несущей частотой. Поскольку скорость передачи информации пропорциональна полосе пропускания, согласно закону Хартли – Шеннона, мы приходим к выводу, что высокая скорость передачи информации требует высокой несущей частоты. Например, микроволновая система с частотой 5 ГГц может вместить в 10 000 раз больше информации за заданный интервал времени, чем радиоканал с частотой 500 кГц. Увеличивая же еще частоту несущей, один



оптический лазерный луч обладает пропускной способностью, эквивалентной 10 млн телевизионных каналов.

Также выбор несущей частоты является характерным преимуществом, например, для настройки радио или телевизора на определенную станцию. Для этого устройству необходимо выбрать один из множества сигналов в пространстве, принимаемых в это время. В связи с тем, что каждой станции назначена разная несущая частота, желаемый сигнал может быть отделен от других с помощью фильтрации. Если бы не модуляция, то только одна станция смогла бы вещать в определенной области или же две (или более) вещательных станций создавали бы смесь помех, из которых невозможно было бы выделить полезный сигнал.

Стоит отметить, что выбор того или иного частотного диапазона в том числе определяется средой распространения радиосигнала (рис. 5.5). Так, например, в радиолокационных системах длину волны выбирают исходя из окон прозрачности.

Также одним из преимуществ использования того или иного метода модуляции является уменьшение уровня шума и помех. Очевидным вариантом борьбы с шумами и помехами является увеличение мощности сигнала до достижения требуемого отношения сигнал/шум по мощности. Однако увеличение мощности сигнала усложняет разработку и производство радиотехнических устройств, поскольку при большой мощности увеличивается риск повреждения оборудования при использовании, в связи с чем также растет его стоимость. В связи с этим для уменьшения уровня шумов и помех

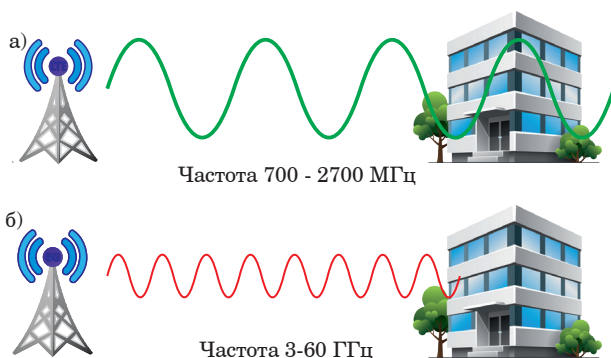


Рис. 5.5. Прохождение радиоволн сквозь препятствия на различных частотах: а – для сети 4G; б – для сети 5G



используют помехозащищенные методы модуляции сигнала, например, такие, как фазовая модуляция.

Это свойство называется широкополосным шумоподавлением, поскольку оно требует, чтобы полоса пропускания передатчика была намного больше полосы пропускания модулирующего сигнала (рис. 5.6). Широкополосная модуляция, таким образом, позволяет разработчику не увеличивать мощность передаваемого сигнала за счет увеличения ширины полосы пропускания, что подразумевается в законе Хартли – Шеннона. Однако стоит понимать, что для обеспечения широкополосной модуляции может потребоваться более высокая несущая частота.

Еще одной из причин использования модуляции является применение ее с целью мультиплексирования. Мультиплексирование – это механизм объединения K сигналов для одновременной передачи в рамках одного канала. Мультиплексирование в основном основывается на двух методах:

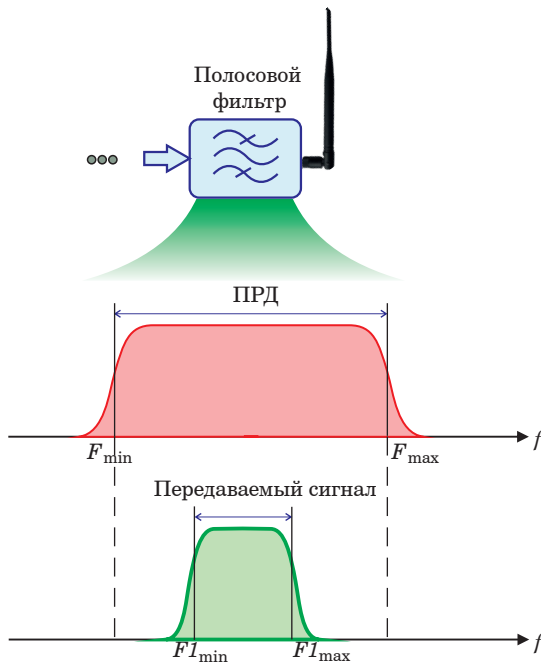


Рис. 5.6. Передача сигнала с меньшей по сравнению с передатчиком шириной частотной спектра



– частотного разделения, в рамках которого на передаче модулятор распределяет сигналы на различных несущих, в то время как на принимаемой стороне они разделяются с использованием системы фильтрации;

– временного разделения, в котором предполагается использование импульсной модуляции для поочередной передачи выборок различных сигналов в непрерывающихся временных интервалах. К примеру, на рис. 5.1в временные промежутки между импульсами могут быть заполнены другими сигналами, также несущими информацию.

Еще одной разновидностью мультиплексирования является множественный доступ. Классические методы мультиплексирования предполагают фиксированное назначение общих ресурсов канала передачи данных (частотный спектр, время) на локальном уровне, в то время как множественный доступ предполагает удаленное совместное использование этих ресурсов. Например, множественный доступ с кодовым разделением (CDMA) присваивает уникальный код каждому пользователю цифровой системы сотовой связи, за счет чего отдельно переданные информационные сообщения разделяются корреляцией между кодами передающей и принимаемой стороной. Поскольку в этом случае разные пользователи используют один и тот же частотный диапазон, обеспечивается дополнительное повышение эффективности цифровой системы связи. Подробнее про множественный доступ будет рассказано в разделе 5.

Цифровые системы обработки информации подразумевают использование одного из двух основных видов модуляции:

- низкочастотная;
- полосовая.

Низкочастотная модуляция предполагает формирование сигнала, спектр которого находится на уровне низких частот, ширина частотного спектра которого равна символьной скорости R_{sym} . То есть в соответствии с рис. 5.1а формируется только огибающее (модулирующее) колебание, в амплитуде которого закодировано информационное сообщение. При этом использование высокочастотного заполнения (несущей) не подразумевается.

В случае же полосовой модуляции (рис. 5.1б, в) используется модулирующий сигнал с целью кодирования информационного сообщения путем изменения параметров несущего колебания. Так, в результате таких преобразований сигнал имеет полосовой спектр, на-



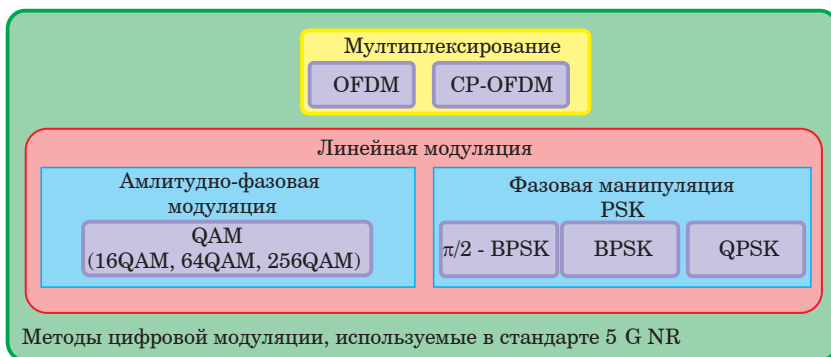


Рис. 5.7. Методы модуляции, используемые в стандарте 5G NR

ходящийся на частоте несущего колебания. Стоит отметить, что, как и в случае низкочастотной модуляции, именно символьная скорость определяет ширину частотного спектра радиосигнала.

Как уже было выяснено, цифровые виды модуляции делятся на низкочастотные и полосовые, а также мультиплексирование и расширение спектра (на рис. 5.7 приведен пример перечня методов модуляций, используемых в стандарте 5G NR). Однако еще одним немаловажным классификационным признаком методов модуляции является линейность и нелинейность процессов формирования модулированного сигнала, которые мы рассмотрим в следующих подпунктах.

5.1. Линейная цифровая модуляция

Под линейными цифровыми методами модуляции понимаются те виды модуляции, в которых информационная составляющая, заложенная в низкочастотном сигнале или огибающей несущего колебания, представляется в виде линейной суммы модулированных сигналов, каждому из которых соответствует один бит или символ, разнесенных по времени

$$s(t) = \sum_{k=-\infty}^{\infty} a_k g(t = kT), \quad (5.1)$$

где T – межсимвольный интервал; $g(t)$ – форма сигнала, соответствующего одному биту или символу; a_k – коэффициенты, значения которых соответствуют перечню \mathbf{C} возможных в рамках конкретного вида модуляции.



Перечень коэффициентов C называется сигнальным созвездием. Коэффициенты a_k являются определяющими в определении передаваемой цифровой информации, поскольку каждый из них соответствует конкретному биту или символу, то есть набору бит, расположенных в определенном порядке. Количество коэффициентов C_M в созвездии определяется количеством возможных комбинаций в соответствии с выражением (1.1).

5.2. Используемые на практике виды сигнальных созвездий

Основным видом модуляции, используемым в цифровых системах с низкочастотной модуляцией, является *амплитудно-импульсная модуляция*.

Сигнальное созвездие в амплитудно-импульсной модуляции имеет одномерное пространство. При этом коэффициенты a_k в этом случае бинарны и имеют значения ± 1 . Отсюда также становится понятно, что коэффициенты a_k и форма сигнала, соответствующего одному биту или символу, являются вещественными. На рис. 5.8 приведены примеры амплитудно-импульсной модуляции.

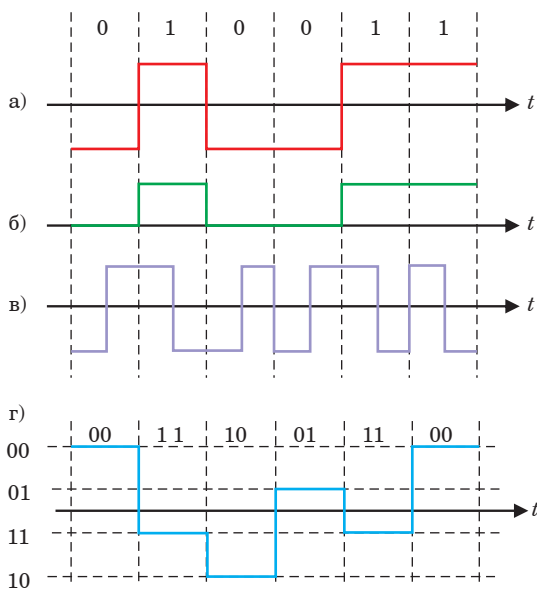


Рис. 5.8. Примеры амплитудно-импульсной модуляции:
 а – NRZ; б – On/Off; в – Manchester; г – PAM 4



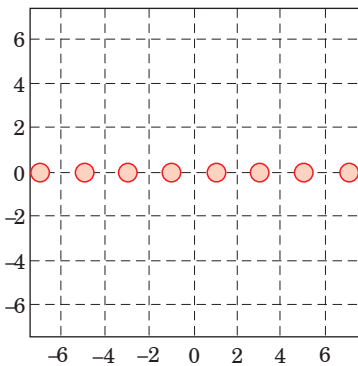


Рис. 5.9. Сигнальное созвездие для восьмипозиционной амплитудной модуляции

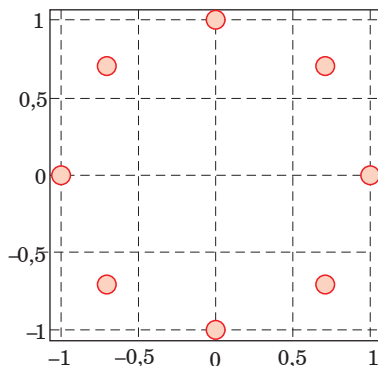


Рис. 5.10. Сигнальное созвездие для восьмипозиционной фазовой модуляции

Пример сигнального созвездия для низкочастотной модуляции представлен на рис. 5.9.

В случае фазовой модуляции синфазная составляющая является константой и изменяется только одна составляющая (рис. 5.10).

Одним из простейших видов цифровой модуляции, применяемых как для низкочастотной, так и для полосовой, является BPSK. Она двухпозиционна, где $M=2$, то есть «0» или «1». Сигнальное созвездие BPSK модуляции приведено на рис. 5.11, а сам сигнал во временной области на рис. 5.12.

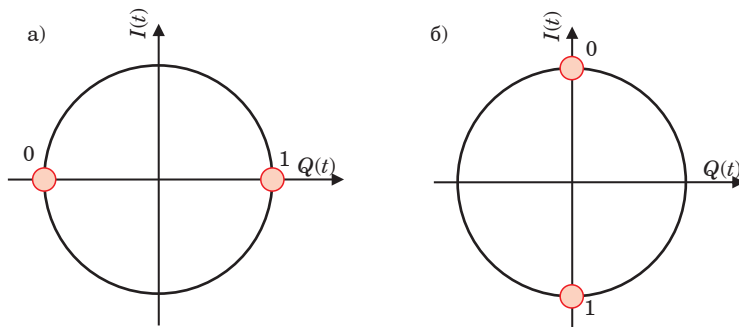


Рис. 5.11. Сигнальное созвездие фазовой манипуляции: а – BPSK модуляция; б – $\pi/2$ – BPSK модуляция



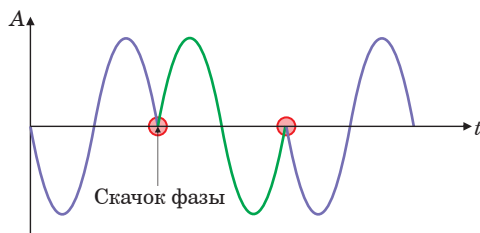


Рис. 5.12. Радиосигнал с BPSK модуляцией во временной области

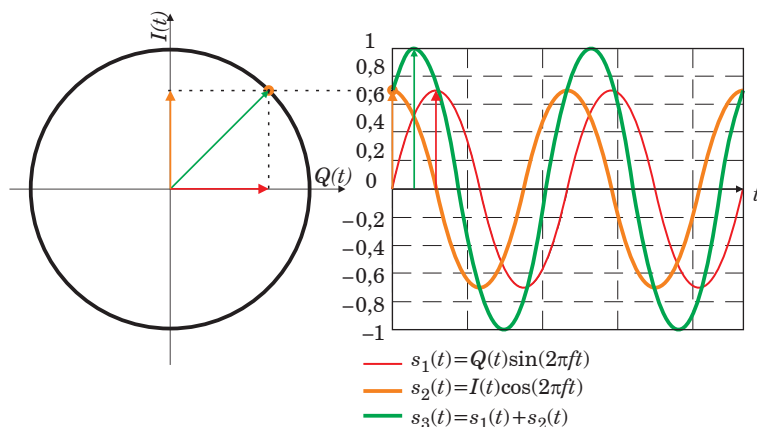


Рис. 5.13. Пояснение к понятию комплексного коэффициента a_k

В случае *полосового вида модуляции* созвездие имеет комплексный характер. То есть каждый сигнал, соответствующий одному биту или символу, коэффициенту a_k , формируется с помощью двух квадратурных составляющих (рис. 5.13) в соответствии с формулой

$$s(t) = Q(t) \sin(2\pi ft) + I(t) \cos(2\pi ft), \quad (5.2)$$

где $Q(t)$, $I(t)$ – квадратурные коэффициенты.

Среди используемых в современных стандартах цифровой связи методов модуляции одним из простых является квадратурная фазовая модуляция (QPSK). В отличие от BPSK, в этом методе модуляции радиосигнала с использованием одного коэффициента созвездия (рис. 5.14) передается 2 бита информации, то есть $M=2$. Для формирования радиосигнала с использованием QPSK модуляции (рис. 5.15) используется выражение



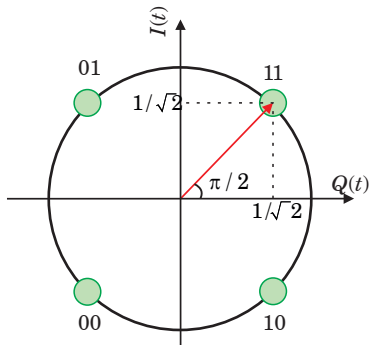


Рис. 5.14. Сигнально созвездие QPSK модуляции

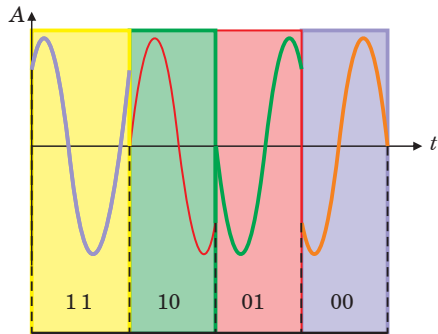


Рис. 5.15. Радиосигнал с использованием QPSK модуляции

$$b_{I,Q}(t) = \frac{1}{k}(I(t) \cos(\omega_0 t) + Q(t) \sin(\omega_0 t)), \quad (5.3)$$

где $b_{I,Q}(t)$ – сигнал, соответствующий одному биту или символу коэффициента a_k сигнального созвездия; $k = \sqrt{4}$ – нормирующий множитель; $I(t)$ и $Q(t)$ – могут принимать значения равные $\pm 1 / \sqrt{2}$.

Для передачи больших объемов данных в современных цифровых системах связи используется квадратурно-амплитудная модуляция. Для обеспечения необходимого уровня помехоустойчивости используются 16-позиционная (16QAM), 64-позиционная (64QAM) и 256-позиционная (256QAM) модуляции. На рис. 5.16 представлено созвездие 16QAM модуляции. Как видно из рисунка, за один коэффициент можно передать символ, состоящий из четырех бит. В системах цифровой связи двойного назначения 16-позиционная модуляция является максимально возможно для обеспечения помехоустойчивости на необходимом уровне. По аналогии с 16 QAM на рис. 5.16 представлено созвездие для 64QAM модуляции, где за один символ передается шесть бит.

Формирование сигнала, модулированного по методу 16QAM, возможно реализовать с использование следующего выражения

$$b_{I,Q}(t) = \frac{1}{k}(I(t) \cos(\omega_0 t) + Q(t) \sin(\omega_0 t)), \quad (5.4)$$

где $b_{I,Q}(t)$ – сигнал, соответствующий одному биту или символу коэффициента a_k сигнального созвездия; $k = \sqrt{10}$ – нормирующий множитель; $I(t)$ и $Q(t)$ – могут принимать значения равные $\pm 1, \pm 5$.



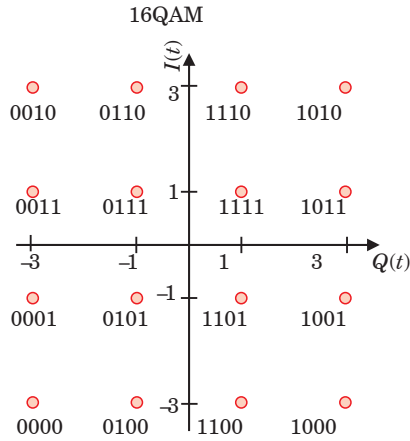


Рис. 5.16. Сигнальное созвездие для 16QAM модуляции

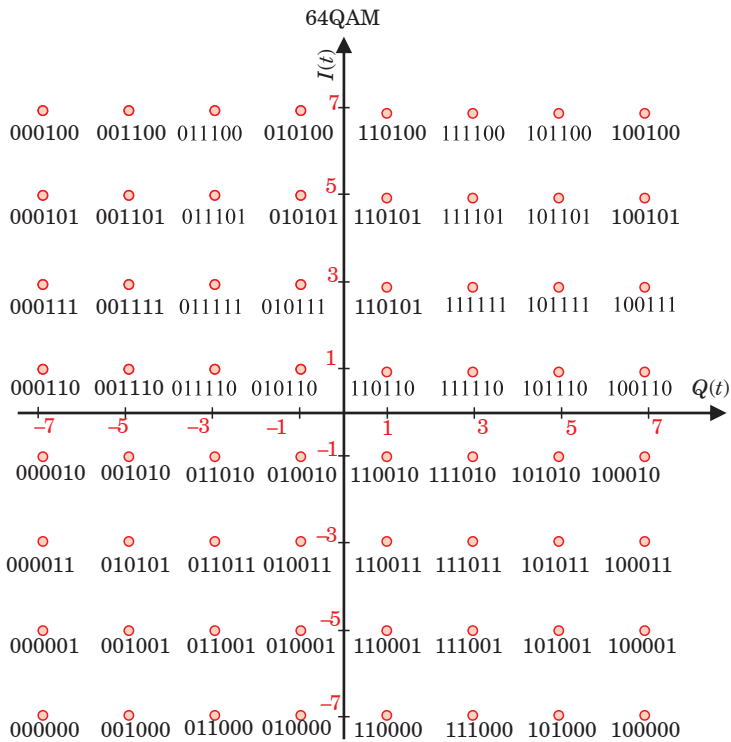


Рис. 5.17. Сигнальное созвездие 64QAM модуляции



Пример формирования радиосигнала с 16-позиционной квадратурно-амплитудной модуляцией в среде Matlab представлен на рис. 5.18. Стоит отметить, что при написании программного кода используются дискретные сигналы и для их реализации необходимо ознакомиться с основами цифровой обработки сигналов. Основы цифровой обработки сигналов приведены в [2].

Результатом работы программного кода является радиосигнал во временной области. Пример программного кода, формирующий радиосигнал с модуляцией 16QAM:

```

clc
clear all
close all
M = 16; % Порядок модуляции
x = (0:M-1)'; % Входная последовательность
IQ=qammod(x,M,'bin'); % коэффициенты квадратурной модуляции
scatterplot(IQ) % Построение комплексной плоскости
title('16QAM модуляция')
axis([-4 4 -4 4])
N = 16; % число символов
aa = randi(numel(IQ), 1, N); % случайные индексы массива IQ
bb = randi(numel(IQ), 1, N); % случайные индексы массива IQ
for com=1:M % Цикл по числу коэффициентов
for j=find(aa==com) % Цикл по количеству случайных чисел
aa(j)= real(IQ(com)); % Замена индекса коэффициента на сам коэф-
фициент
bb(j) = imag(IQ(com)); % Замена индекса коэффициента на сам коэф-
фициент
end
end
Fd = 2.43e3; % символьная скорость
Fc = 1.6e3; % несущая частота
FsFd =128; % число отсчетов на один символ
Fs = Fd * FsFd; % частота дискретизации дублируем каждый отсчет
FsFd раз
a1 = repmat(aa, FsFd, 1);
a1 = a1(:);
b1 = repmat(bb, FsFd, 1);
b1 = b1(:);
t = (0:N*FsFd-1)/Fs; % дискретное время
t = t'; % превращаем строку в столбец
k=1/sqrt(10); % нормирующий множитель
s_qam16 = k*(a1 .*...
cos(2*pi*Fc*t) ...
+b1 .* sin(2*pi*Fc*t)); %Сигнал с QAM модуляцией
figure
plot(t, s_qam16)

```



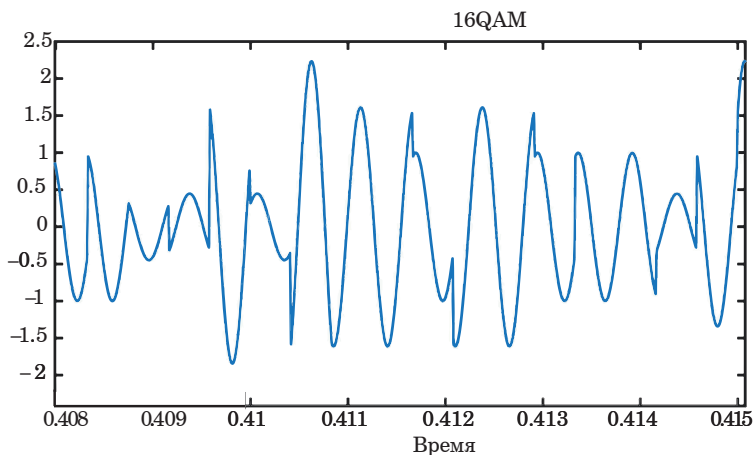


Рис. 5.19. Радиосигнал с модуляцией 16QAM во временной области

Как видно из рис. 5.19, результирующий радиосигнал имеет последовательность отрезков с сигналами, имеющими различную начальную фазу и амплитуду.

Для примера и дальнейшего анализа на рис. 5.20 и 5.21 представлены программные коды в среде Matlab.

Программный код реализации сигнала с амплитудной манипуляцией:

```
clear all
close all
sy = [1 3 2 4 1]; % передаваемые символы
Fd = 1; % символьная скорость
Fc = 10; % несущая частота
FsFd = 100; % отношение Fs/Fd
Fs = Fd * FsFd; % частота дискретизации
N0=4096; % Количество дискретных отсчетов для FFT
t = (0:length(sy)*FsFd-1)/Fs; % дискретное время
% формируем АМН-сигнал
s_ask = sy(floor(Fd*t)+1) .* cos(2*pi*Fc*t);
s_ask=awgn(s_ask,10);
W=abs(fft(s_ask));
W=2*W./N0;
fx=(0:N0-1)*Fs/N0;
fx=fx(1:length(t));
subplot(2,2,2)
plot(t, s_ask)
title('Амплитудная манипуляция')
xlabel('Время')
```



```

ylabel('Амплитуда')
subplot(2,2,4)
plot(fx, W)
title('Спектр')
xlabel('Частота')
xlabel('Символы')
ylabel('Амплитуда')
ylim([-1.1 1.1])

```

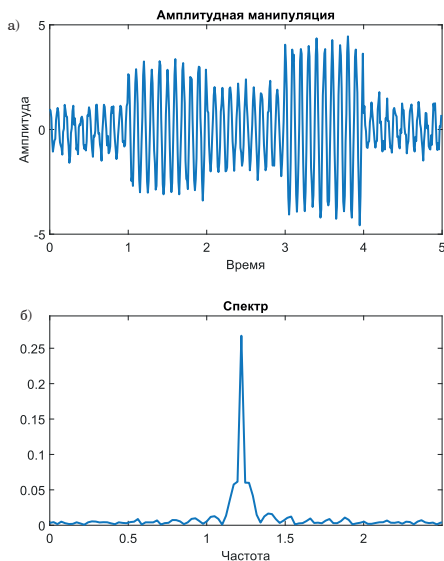


Рис. 5.20. Пример программного кода для сигнала с амплитудной манипуляцией:
 а – результат выполнения программы на временной оси;
 б – результат выполнения программы на частотной оси

Программный код сигнала с частотной манипуляцией:

```

clear all
close all
bits = [0 1 0 0 1]; % цифровое сообщение
N = length(bits); % длина сообщения
N0=4096;% Количество дискретных отсчетов для FFT
Fd = 300; % символьная скорость
FsFd = 200; % отношение Fs/Fd
Fs = Fd * FsFd; % частота дискретизации
f = [980 1180]; % частоты манипуляции
t = (0:N*FsFd-1)/Fs; % дискретное время
carriers = cos(2*pi*t'*f); % столбцы – колебания для "0" и "1"
mask = repmat(bits, FsFd, 1); % маска переключения колебаний

```



```

mask = mask(:); % "растягиваем" маску в один
столбец
s_fsk = carriers(:,1) .*...
(1-mask) + carriers(:,2) .* mask; % ЧМН сигнал
s_fsk=awgn(s_fsk,20)
td = (0:N*FsFd-1)/FsFd; % время для графика - в символах
W=abs(fft(s_fsk));
W=2*W./N0;
fx=(0:N0-1)*Fs/N0;
fx=fx(1:length(t));
subplot(2,2,2)
plot(td, (s_fsk)')
title('Частотно манипулированный сигнал')
xlabel('Символы')
ylabel('Амплитуда')
ylim([-1.1 1.1])
subplot(2,2,4)
plot(fx,W)
xlabel('Частота')

```

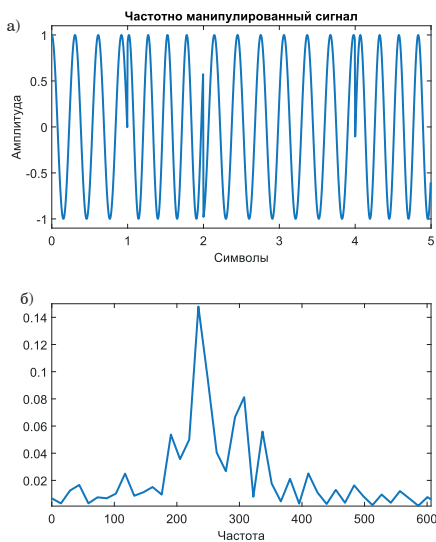


Рис. 5.21. Пример программного кода для сигнала с частотной манипуляцией:
а – результат выполнения программы на временной оси;
б – результат выполнения программы на частотной оси



6. МЕТОДЫ ОБЩЕГО ИСПОЛЬЗОВАНИЯ ЧАСТОТНО-ВРЕМЕННОГО РЕСУРСА ЦИФРОВОЙ СИСТЕМЫ ОБМЕНА ИНФОРМАЦИЕЙ

Организация множественного доступа основывается на двух основных архитектурах цифровых систем обработки информации: системы с использованием базовых станций и системы без их использования.

В системах с использованием базовых станций передача информации осуществляется по одному из двух путей:

- линия вверх, в которой осуществляется передача данных от пользователя или устройства к базовой станции;
- линия «вниз», в которой осуществляется передача данных от базовой станции к пользователю или устройству.

Для организации каналов «вверх» и «вниз» существует два основных метода (рис. 6.1):

- частотное разделение каналов (FDD), при котором каждому из каналов выделяется свой частотный диапазон;

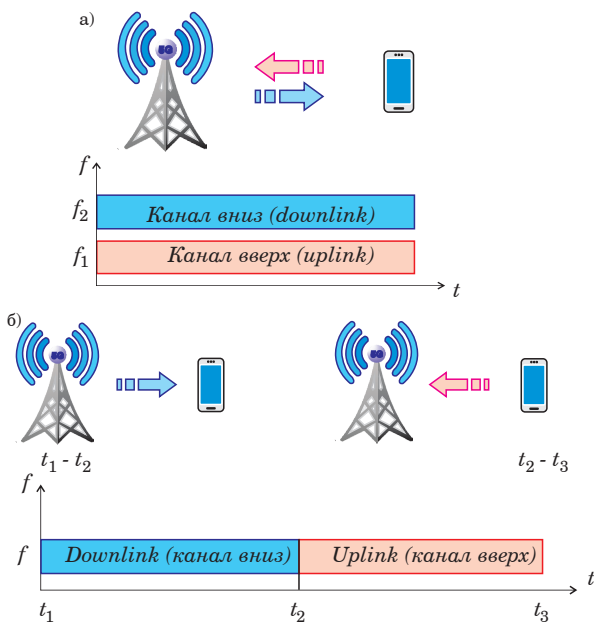


Рис. 6.1. Режимы обмена информацией в цифровых системах передачи информации: а – частотное разделение каналов (FDD); б – временное разделение каналов (TDD)



– временное разделение каналов (TDD), при котором каждому каналу отводится определенный промежуток времени.

Особенностями линии «вниз» является:

– обеспечение высокого уровня синхронизации по времени за счет того, что сигнал от базовой станции формируется суммой сигналов для всех абонентских устройств одновременно;

– искажение полезного сигнала и помех, являющихся сигналами для других абонентских устройств, вносится одинаково.

Для линии «вверх» свойственно:

– отсутствие временной синхронизации сигналов, направляемых от всех абонентских устройств в области одной базовой станции, поскольку все сигналы являются независимыми и направляются с различных направлений и расстояний (обеспечить временную синхронизацию возможно, однако это потребует усложнение системы, поскольку для этого необходимо было бы организовать синхронизирующий сигнал по линии «вниз»);

– искажение всех сигналов различны в связи с тем, что каждый из них отправляется к базовой станции по своему радиоканалу.

Системы без использования базовых станций осуществляют взаимодействие в условиях отсутствия контроля инфраструктуры. То есть взаимодействие осуществляется напрямую между пользователями или устройствами, некоторые из которых в отдельных случаях могут являться ретрансляторами передаваемых сообщений и данных.

Первая архитектура множественного доступа (например, сети беспроводной связи, системы ближней навигации, радиорелейная связь) основана на том, что пользователи не осуществляют обмен информацией между собой. Для этого используются базовые станции. Они координируют работу сети, разгружают и нагружают частотные диапазоны, выступают в роли ретрансляторов.

Организовать использование цифровой системы обмена информацией двумя или более абонентскими устройствами можно за счет использования одного из двух методов:

– для постоянного обмена информационными сообщениями при заданной символьной или битовой скорости использовать один из методов разделения каналов (множественного доступа), при этом всем абонентским устройствам обеспечивается свободный участок частотно-временного ресурса (радиоинтерфейса);

– для случая, когда осуществляется регулярный обмен информацией, но необходимо обеспечить минимальный уровень задерж-



ки передачи сигнала, используют методы случайного доступа, при этом абонентские устройства используют общие каналы передачи данных в условиях отсутствия четкой организации последовательности действий (в этом случае наличие общего доступа обеспечивается с высокой долей вероятности, причем отсутствие сбоев в работе не гарантируется).

Множественный доступ – это методы мультиплексирования, которые предоставляют абонентским устройствам общий доступ к частотно-временному ресурсу в проводной или беспроводной среде с одной полосой пропускания. Каналы обмена цифровой информацией, будь то диапазоны частотного спектра или кабельные соединения, стоят дорого. К примеру, телекоммуникационные операторы услуг связи должны привлекать нескольких платных пользователей с ограниченными ресурсами, чтобы получать прибыль. Однако методы множественного доступа позволяют большому количеству абонентских устройств совместно использовать эти ограниченные каналы, обеспечивая экономию за счет масштаба, необходимую для успешного коммуникационного бизнеса. Существует пять основных методов множественного доступа, или мультиплексирования:

- с частотным разделением (FDMA);
- с временным разделением (TDMA);
- с кодовым разделением (CDMA);
- с ортогональным частотным разделением (OFDMA);
- с пространственным разделением (SDMA).

6.1. Множественный доступ с разделением каналов по частоте

FDMA – это метод разделения одного канала (или диапазона частотного спектра) на несколько отдельных поддиапазонов, каждый из которых предназначен для использования одним абонентским устройством (рис. 6.2). Каждая отдельная полоса или частотный поддиапазон достаточно широки, чтобы вместить спектры сигналов, передаваемых цифровых информационных сообщений. Данные, подлежащие передаче, модулируются на каждой поднесущей, и все они линейно смешиваются вместе.

Хорошим примером цифровой системы передачи данных с использованием метода множественного доступа FDMA является система кабельного телевидения. В качестве канала передачи цифровых данных используется один коаксиальный кабель, который



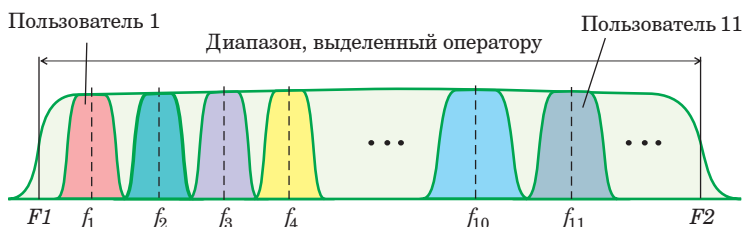


Рис. 6.2. Организация каналов передачи данных методом множественного доступа FDMA

применим для трансляции сотен видео-, аудиоканалов и программ в дома. Коаксиальный кабель имеет полезную полосу пропускания примерно от 4 МГц до 1 ГГц. Эта полоса пропускания разделена на каналы шириной 6 МГц. Первоначально одна телевизионная станция или канал использовали один диапазон 6 МГц. Но благодаря цифровым технологиям сегодня несколько телевизионных каналов могут применять одну полосу благодаря методам сжатия и мультиплексирования, используемым в каждом канале.

Этот метод также используется в волоконно-оптических линиях связи. Один волоконно-оптический кабель имеет огромную полосу пропускания, которая может быть разделена для обеспечения FDMA. Каждому из различных источников данных или информации назначается своя частота света для передачи. Свет обычно обозначается не частотой, а длиной его волны (λ). В результате волоконно-оптический FDMA называется методом множественного доступа с разделением по длине волны (WDMA) или просто мультиплексированием с разделением по длине волны (WDM).

Одной из старых систем FDMA является оригинальная аналоговая телефонная система, которая использовала систему методов частотного мультиплексирования для передачи нескольких телефонных звонков по одной линии. Аналоговые речевые сигналы частотой от 300 до 3400 Гц применялись для модуляции поднесущих в 12 каналах от 60 до 108 кГц. Модулятор/микшеры создавали сигналы с одной боковой полосой (SSB) как в верхней, так и в нижней боковых полосах. Затем эти поднесущие были дополнительно мультиплексированы по частоте на поднесущих в диапазоне от 312 до 552 кГц с использованием тех же методов модуляции. На приемной части системы сигналы сортировались и восстанавливались с помощью фильтров и демодуляторов.



В аэрокосмических телеметрических системах использовали систему FDMA для передачи данных нескольких датчиков по одному радиоканалу. Ранние спутниковые системы совместно применяли отдельные транспондеры с полосой пропускания 36 МГц в диапазоне от 4 до 6 ГГц для передачи нескольких сигналов голоса, видео или данных через FDMA. Сегодня все эти системы используют метод множественного доступа TDMA.

Принцип работы цифровой системы, основанной на FDMA, проиллюстрирован на рис. 6.3а. На рис. изображены несколько входящих сообщений, каждое из которых модулируют поднесущие f_{c1} , f_{c2} и так далее после прохождения через фильтр нижних частот для ограничения полосы пропускания сообщений. Мы показыва-

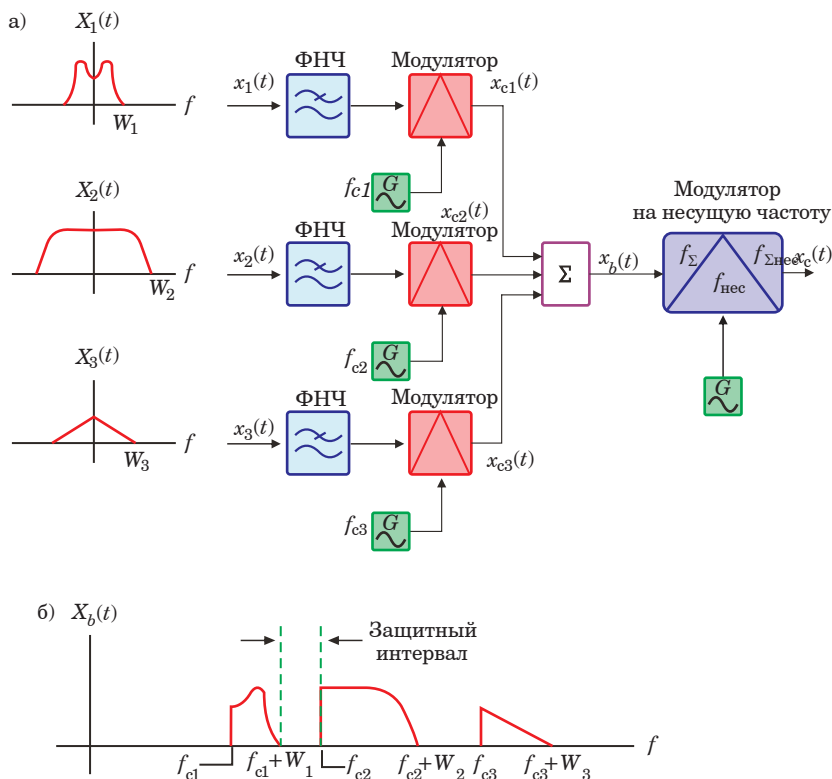


Рис. 6.3. Пример работы цифровой системы передачи информации, основанной на использовании метода FDMA: а – структурная схема передатчика; б – спектр системы с FDMA



ем модуляцию поднесущей как однополосная модуляция, как это часто бывает, но можно использовать любой из методов модуляции СВ или их смесь. Затем модулированные сигналы суммируются для получения сигнала основной полосы частот со спектром $x_b(f)$, как показано на рис. 6.2 – 1b (Обозначение «основная полоса частот» указывает на то, что окончательная модуляция несущей еще не произошла.) Временная функция основной полосы частот $x_b(f)$ остается на усмотрение читателя.

Предполагая, что частоты поднесущих выбраны правильно, операция мультиплексирования назначила слот в частотной области для каждого из отдельных сообщений в модулированной форме, отсюда и название мультиплексирование с частотным разделением. Затем сигнал основной полосы частот может передаваться непосредственно или использоваться для модуляции передаваемой несущей частоты f_c . Здесь нас не особенно интересует природа конечной модуляции несущей, поскольку спектр основной полосы говорит сам за себя.

Восстановление сообщения или демодуляция FDM выполняется в три этапа. Сначала демодулятор несущей воспроизводит сигнал основной полосы частот $x_b(f)$. Затем модулированные поднесущие разделяются параллельно блоком полосовых фильтров, после чего сообщения обнаруживаются индивидуально.

Основной практической проблемой FDM являются перекрестные помехи, нежелательное соединение одного сообщения с другим. Понятные перекрестные помехи (перекрестная модуляция) возникают главным образом из-за нелинейностей в системе, которые заставляют один сигнал сообщения отображаться как модуляция на другой поднесущей. Следовательно, стандартная практика требует отрицательной обратной связи для минимизации нелинейности усилителя в системах FDM. (Исторически сложилось так, что проблема перекрестных помех FDM была основным стимулом для разработки усилителей с отрицательной обратной связью.)

Неразборчивые перекрестные помехи могут быть вызваны нелинейными эффектами или несовершенным разделением спектра с помощью блока фильтров. Чтобы уменьшить последнее, спектры модулированных сообщений разнесены по частоте защитными полосами, в которые могут быть вставлены области перехода фильтра. Например, защитная полоса, обозначенная на рис. 6.2 – 1b, имеет ширину $fc2$ ($fc1 W1$). Таким образом, чистая полоса пропускания основной полосы представляет собой сумму полос пропускания



модулированных сообщений плюс защитные полосы. Но схема на рис. 6.2 – 2 – не единственный пример FDM. Коммерческие диапазоны вещания АМ или FM являются повседневными примерами FDMA, когда несколько вещательных компаний могут передавать одновременно в одном и том же диапазоне, но на немного разных частотах.

6.2. Множественный доступ с разделением каналов по времени

TDMA – это цифровой метод множественного доступа, который делит один канал или временную часть частотно-временного ресурса на интервалы (рис. 6.4). Каждый временной интервал используется для передачи одного байта или другого цифрового сегмента каждого сигнала в последовательном формате последовательных данных. Этот метод хорошо работает при малой скорости передачи сигналов с голосовыми данными, но также он полезен при передаче сжатого видео и других данных, требующих обеспечения высокой скорости передачи информации.

Хорошим примером является широко используемая система передачи данных T1, которая уже много лет используется в телекоммуникационной отрасли (рис. 6.5). Линии T1 передают до 24 отдельных голосовых телефонных вызовов по одной линии. Каждый речевой сигнал обычно охватывает диапазон от 300 до 3000 Гц и оцифровывается со скоростью 8 кГц, что лишь немного превышает минимальную частоту Найквиста, в два раза превышающую самую высокочастотную составляющую, необходимую для сохранения всего аналогового контента.

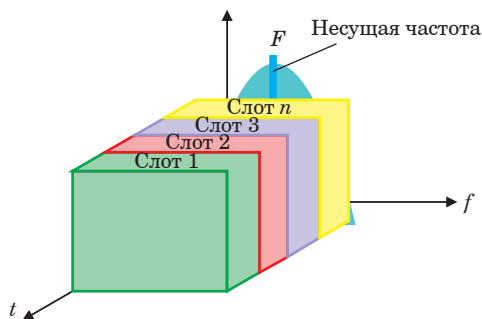


Рис. 6.4. Организация частотно-временного ресурса канала передачи данных методом множественного доступа TDMA



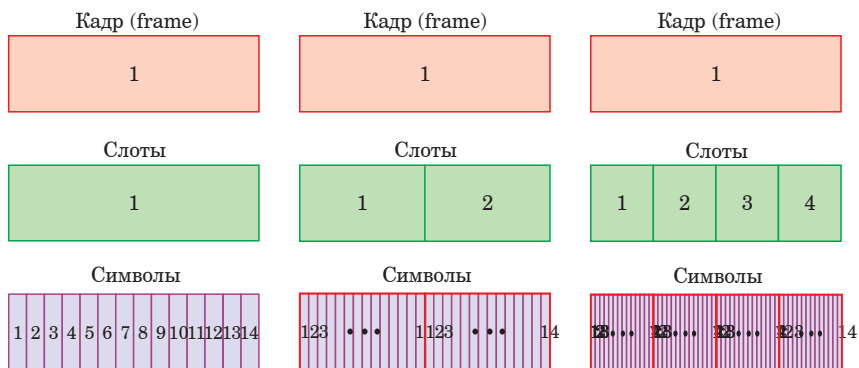


Рис. 6.5. Пример структуры кадра для различных технологий

Оцифрованный голос отображается в виде отдельных последовательных байтов, которые происходят со скоростью 64 кГц, и 24 из этих байтов чередуются, создавая один кадр данных T1. Кадр выполняется с частотой 1,536 МГц (24 на 64 кГц), в общей сложности 192 бита. Для синхронизации добавляется один синхронизирующий бит, обеспечивающий общую скорость передачи данных 1,544 Мбит/с. На принимающей стороне отдельные голосовые байты восстанавливаются со скоростью 64 кГц и передаются через цифроаналоговый преобразователь (ЦАП), который воспроизводит аналоговый голос.

Ранее используемая базовая система сотовой связи GSM (глобальная система мобильной связи) была основана на TDMA. Радиочастотный спектр делился на полосы частот 200 кГц, а затем использовались методы разделения по времени для передачи восьми голосовых сообщений в один канал. Восемь временных интервалов могли быть голосовыми сигналами или данными, такими как текстовые сообщения или электронные письма. Кадр передавался со скоростью 270 кбит/сек. с использованием гауссовой минимальной манипуляции сдвигом (GMSK), которая является формой модуляции с частотной манипуляцией (FSK).

6.3. Множественный доступ с кодовым разделением

Еще одним методом организации множественного доступа к общему каналу передачи данных (рис. 6.6) является CDMA (Code-Division Multiple Access), организованный путем кодового разде-



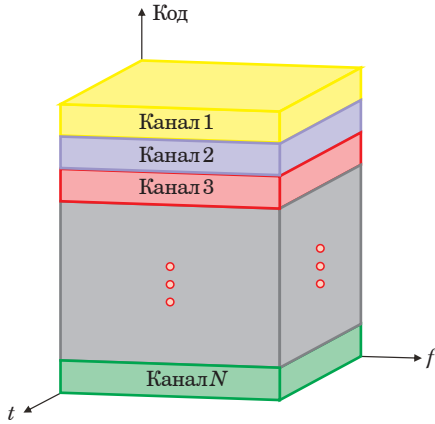


Рис. 6.6. Кодовое разделение каналов

ления абонентских устройств и рассчитанный исключительно для цифровых систем. CDMA также известен как метод расширения спектра, поскольку предполагается, что он принимает оцифрованную версию аналогового сигнала и распространяет его по более широкой полосе пропускания при более низком уровне мощности (рис. 6.7). Он также называется методом прямой последовательности для расширения спектра DSSS (Direct Sequence Spread

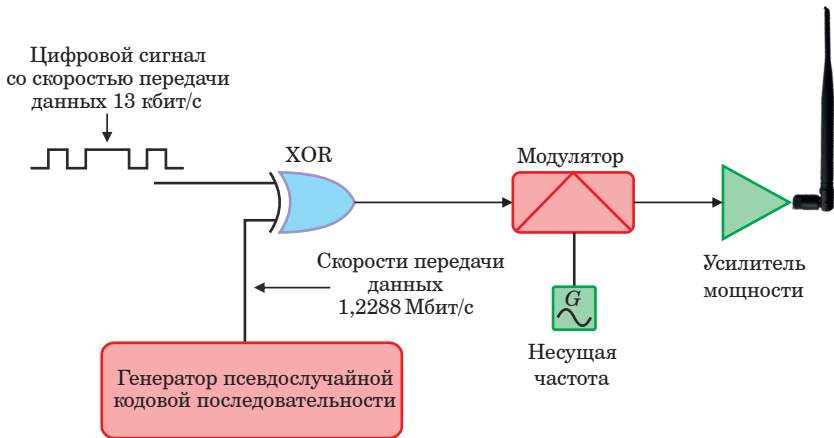


Рис. 6.7. Схема передатчика в цифровой системе, использующей метод множественного доступа CDMA



Spectrum). Оцифрованный и сжатый речевой сигнал в виде последовательных данных распространяется путем обработки его в схеме XOR вместе с сигналом скола на гораздо более высокой частоте. Например, в стандарте CDMA IS-95 сигнал со скоростью передачи данных 1,2288 Мбит/с передает оцифрованный сжатый голос со скоростью 13 Кбит/с.

Сигнал кодирования подается с генератора псевдослучайного кода, который присваивает уникальный код каждому пользователю канала. Этот код распространяет голосовой сигнал по полосе пропускания 1,25 МГц. В итоге результирующий сигнал имеет низкий уровень мощности и больше похож на шум. Множество таких сигналов может занимать один и тот же канал одновременно. Например, использование 64 уникальных кодов шифрования позволяет пользователям (до 64) одновременно использовать один и тот же канал с частотой 1,25 МГц. В приемнике коррелирующая схема находит и идентифицирует код конкретного вызывающего абонента и восстанавливает его.

Также, к примеру, технология сотовой связи третьего поколения (3G), называемая широкополосной CDMA (WCDMA), использует аналогичный метод со сжатым голосом и кодами шифрования 3,84 Мбит / с в канале 5 МГц, чтобы позволить нескольким пользователям использовать один и тот же диапазон.

6.4. Множественный доступ с ортогональным частотным разделением каналов

OFDMA (Orthogonal Frequency-Division Multiplexing Acces) – это метод множественного доступа, используемый в системах сотовой связи со стандартом беспроводной передачи данных для абонентских терминалов LTE и 5G NR для размещения нескольких пользователей в заданной полосе пропускания. Мультиплексирование с ортогональным частотным разделением (OFDM) – это метод модуляции, который делит канал на несколько узких ортогональных полос, расположенных друг от друга таким образом, чтобы они не мешали друг другу. Каждая полоса разделена на сотни или даже тысячи поднесущих шириной 15, 30 или 60 кГц.

Данные, подлежащие передаче, разделяются на множество битовых потоков с более низкой скоростью и модулируются на поднесущие. Временные интервалы в каждом потоке данных подканала используются для организации передаваемых данных (рис. 6.8). Этот метод очень эффективен в спектральном отношении, поэтому



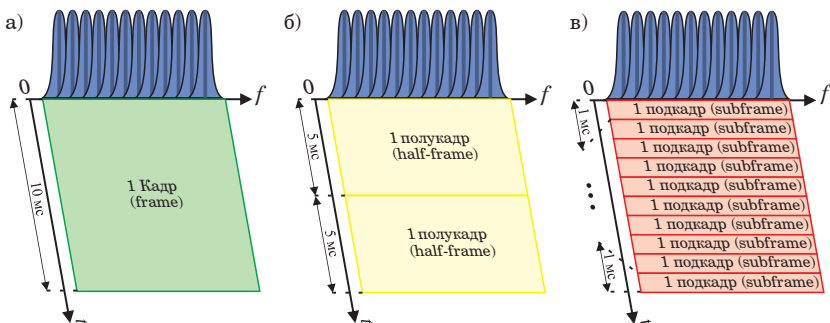


Рис. 6.8. Структура кадра радиоинтерфейса NR:
 а – кадр; б – полукадр; в – подкадр

он обеспечивает очень высокие скорости передачи данных. На него также в меньшей степени влияют эффекты многолучевого распространения.

На рис. 6.9 для наглядности приведена ресурсная сетка радиоинтерфейса 5G NR, организованная с учетом использования метода множественного доступа OFDM.

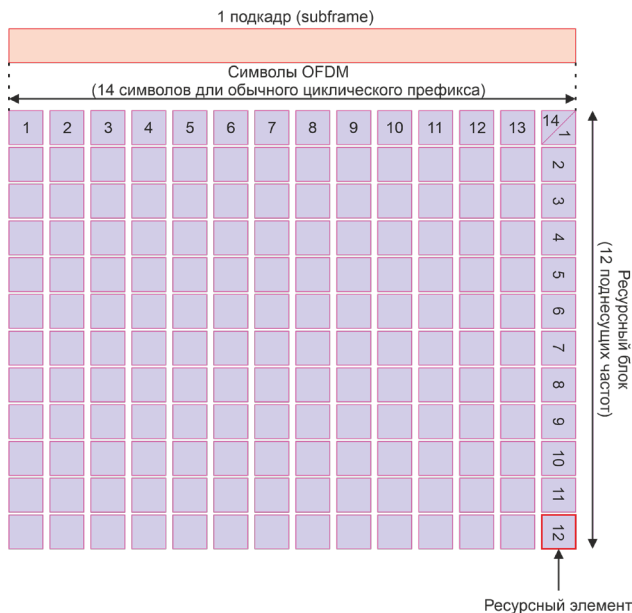


Рис. 6.9. Ресурсная сетка частотно-временного интерфейса NR



6.5. Множественный доступ с пространственным разделением каналов

Еще одним методом множественного доступа к общему каналу передачи данных является SDMA (Spatial Division Access). Этот метод основан на использовании принципов физического разделения, которые позволяют различным абонентским устройствам одновременно использовать беспроводные каналы передачи данных. Например, один канал может использоваться одновременно, если пользователи находятся на достаточном расстоянии друг от друга, чтобы избежать помех. Этот метод, известный как повторное использование частоты, широко применяется в системах сотовой радиосвязи. Базовые станции расположены на расстоянии друг от друга, чтобы свести к минимуму помехи.

В дополнение к расстоянию направленные SMART антенны используются для предотвращения помех. Большинство базовых станций используют три антенны для создания секторов под углом 120° , которые позволяют совместно использовать частоты (рис. 6.10а). Новые технологии, такие как умные SMART антенны или адаптивные антенные решетки, используют динамическое формирование луча для сжатия сигналов в узкие лучи, которые могут быть сфокусированы на конкретных абонентских устройствах, исключая всех остальных (рис. 6.10б).

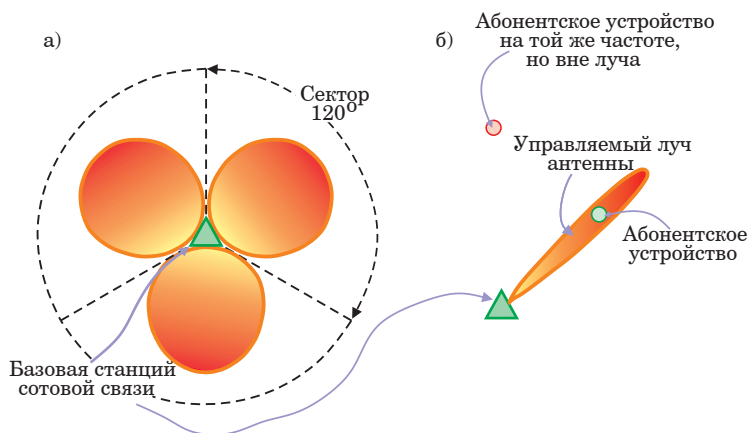


Рис. 6.10. Пояснение к методу множественного доступа SDMA:
а – сектора сканирования базовой станции;
б – пояснение к понятию SMART антенны



Одна из уникальных разновидностей SDMA – множественный доступ с разделением поляризации (PDMA) – разделяет сигналы с помощью различных поляризаций антенн. Затем два разных сигнала могут использовать одну и ту же частоту: один передает вертикально поляризованный сигнал, а другой горизонтально поляризованный сигнал.

Сигналы не будут мешать друг другу, даже если они находятся на одной частоте, потому что они ортогональны, и антенны не будут реагировать на противоположно поляризованный сигнал. Отдельные вертикальные и горизонтальные приемные антенны используются для восстановления двух ортогональных сигналов. Этот метод широко используется в спутниковых системах.

Поляризация также применяется для мультиплексирования в волоконно-оптических системах. Новые системы со скоростью 100 Гбит/с используют двухполяризационную квадратурную фазовую манипуляцию (DP-QPSK) для достижения высоких скоростей на одном волокне. Высокоскоростные данные разделяются на два более медленных потока данных, один из которых использует вертикальную поляризацию света, а другой – горизонтальную. Поляризационные фильтры разделяют два сигнала на передатчике и приемнике и объединяют их обратно в высокоскоростной поток.

6.6. Случайный доступ

В цифровых системах обработки информации под термином «случайный доступ» понимают такую структуру общего доступа к каналу передачи данных, которая предполагает отсутствие предоставления элементов радиointерфейса для конкретных абонентских устройств. То есть сообщения передаются в произвольные временные интервалы, а его получение находится под контролем со стороны приема по каналу обратной связи.

Из этого следует, что передаваемые сообщения могут перекрывать друг друга во времени, в связи с чем могут возникать ошибки, из-за которых некоторые сообщения не будут доставлены. Тогда в случае, если передающая сторона не дожидается подтверждения о передаче сообщения, оно отправляется повторно.

В настоящем учебном пособии будет рассмотрен один из самых простых методов – ALOHA.

Для описания максимальной нагрузки на сеть, использующую ALOHA, воспользуемся законом Пуассона. Для этого учтем, что суммарно сила приема цифровых сообщений от всех абонентских



устройство определяется λ сообщений за единицу времени, а их длительность – τ . Таким образом, вероятность события, при котором за временной промежуток Δt будет принято k сообщений, определяется формулой

$$P(k, \Delta t) = \frac{(\lambda \Delta t)^k}{k!} e^{-\lambda \Delta t}. \quad (6.1)$$

Примем, что цифровые сообщения не искажаются в радиоканале, но в случае, если два сообщения поступят в одно и то же время, то сообщение будет утеряно. Такая вероятность не произойдет в случае, если во временном интервале 2τ не будет передано более одного сообщения. То есть вероятность такого события можно выразить формулой

$$P_{\text{досм}} = P(0, 2\tau) = e^{-2\lambda\tau}. \quad (6.2)$$

Таким образом, частота передачи сообщения без его потери будет определяться $\lambda P_{\text{досм}}$. Для определения временного промежутка, в котором будут присутствовать только сообщения, доставленные без потери, используется выражение

$$\rho = \lambda\tau P_{\text{досм}} = \lambda\tau e^{-2\lambda\tau}. \quad (6.3)$$

На рис. 6.11 продемонстрированы результаты расчета в соответствии с выражением 6.3.

Также на рис. 6.11 приведена функция эффективности при использовании так называемой slotted ALOHA. Этот метод появился

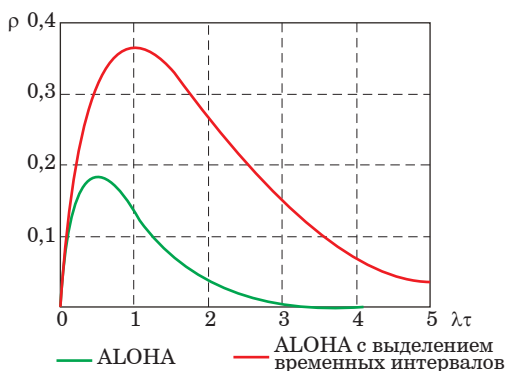


Рис. 6.11. Эффективность канала передачи данных при использовании метода ALOHA



при модернизации АЛОНА и подразумевает выделение временных интервалов. То есть во времени выделяются слоты с временным промежутком равным τ

$$\rho = \lambda \tau e^{-\lambda \tau}. \quad (6.4)$$

Зависимость, показанная на рис. 6.11 красным цветом, имеет максимум при $\lambda \tau = 1$, что в свою очередь позволяет достичь вдвое большей эффективности использования канала, чем при «чистой» АЛОНА.



СПИСОК ИСТОЧНИКОВ

1. *Тихвинский В. О., Терентьев С. В., Коваль В. А.* Сети мобильной связи 5G, технологии, архитектура и услуги. М.: Медиа Паблишер, 2020.
2. *Голдсмит А.* Беспроводные коммуникации / Пер. с англ. М.: Техносфера, 2011.
3. *Zhang X.* LTE Optimization Engineering HandbookX // China Mobile Group Design Institute Co., Ltd, 2018. 845 p.
4. *Сергиенко А. Б.* Цифровая связь: учеб. пособие. СПб.: СПбГЭТУ ЛЭТИ, 2012. 165 с.
5. Широкополосные беспроводные сети передачи информации / В. М. Вишневецкий, А. И. Ляхов, С. Л. Портной, И. В. Шахнович. М.: Техносфера, 2005. 592 с.
6. *Оппенгейм А. В., Шафер Р. В.* Цифровая обработка сигналов: учеб. пособие. М.: Техносфера, 2010. 1049 с.
7. *Сергиенко А. Б.* Цифровая обработка сигналов: учеб. пособие. СПб.: БХВ-Петербург, 2013. 768 с.
8. *Скляр Б.* Цифровая связь. Теоретические основы и практическое применение / Пер. с англ. М.: Вильямс, 2003.
9. *Прокис Д.* Цифровая связь / Пер. с англ.; под ред. Д. Д. Кловского. М.: Радио и связь, 2000.
10. *Ипатов В. П.* Широкополосные системы и кодовое разделение сигналов. Принципы и приложения / Пер. с англ. М.: Техносфера, 2007.



СОДЕРЖАНИЕ

Список условных обозначений и сокращений.....	3
Введение	5
1. Общие понятия и принципы цифровых систем обработки информации	6
1.1. Основные термины цифровых систем обработки информации	6
1.2. Основные функциональные элементы цифровых систем обработки информации.....	13
2. Канал передачи данных.....	18
2.1. Канал связи с аддитивным гауссовым шумом	18
3. Канальное кодирование	22
3.1. Общие принципы и терминология	22
3.2. Классификация методов помехоустойчивого кодирования.....	25
3.3. Блочные методы кодирования.....	27
3.4. Сверточные методы кодирования.....	36
4. Основы теории цифровой обработки сигналов	56
4.1. Теорема Котельникова.....	57
4.2. Спектр сигнала.....	60
4.3. Ряд Фурье	61
4.4. Преобразование Фурье.....	63
5. Модуляция	75
5.1. Линейная цифровая модуляция.....	82
5.2. Используемые на практике виды сигнальных созвездий.....	83
6. Методы общего использования частотно-временного ресурса цифровой системы обмена информацией	92
6.1. Множественный доступ с разделением каналов по частоте.....	94
6.2. Множественный доступ с разделением каналов по времени	98
6.3. Множественный доступ с кодовым разделением.....	99
6.4. Множественный доступ с ортогональным частотным разделением каналов	101



6.5. Множественный доступ с пространственным разделением каналов	103
6.6. Случайный доступ	104
Список источников	107



Учебное издание

**Ермаков Александр Константинович,
Поваренкин Николай Владимирович**

**ЦИФРОВЫЕ СИСТЕМЫ
ОБРАБОТКИ ИНФОРМАЦИИ.
МЕТОДЫ ЦИФРОВОЙ ОБРАБОТКИ**
Учебное пособие

ISBN: 978-5-8088-1836-1



9 785808 818361

Редактор *Е. В. Торопова*
Компьютерная верстка *А. Н. Колешко*

Подписано к печати 26.05.2023. Формат 60 × 84 1/16.
Усл. печ. л. 6,3. Уч.-изд. л. 6,6.
Тираж 50 экз. Заказ № 174.

Редакционно-издательский центр ГУАП
190000, г. Санкт-Петербург, ул. Большая Морская, д. 67, лит. А

FreeBusta is knowledge
without borders!

