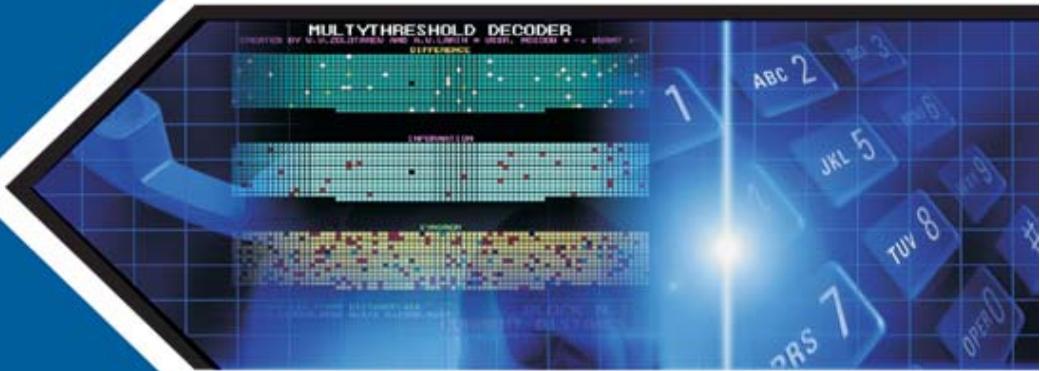


ГОРЯЧАЯ ЛИНИЯ - ТЕЛЕКОМ

В. В. Золотарёв  
Г. В. Овечкин

# ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

Методы и алгоритмы



СПРАВОЧНИК

УДК 621.391.15

ББК 32.811.4

3-80

**Золотарёв В. В., Овечкин Г. В.**

3-80      Помехоустойчивое кодирование. Методы и алгоритмы: Справочник / Под. ред. чл.-кор. РАН Ю. Б. Зубарева. – М.: Горячая линия–Телеком, 2004. – 126 с.: ил.

**ISBN 5-93517-169-4**

Систематически изложены методы и описаны характеристики различных алгоритмов помехоустойчивого кодирования. Рассмотрены методы многопорогового декодирования для блоковых и сверточных кодов, которые имеют технологические преимущества перед другими алгоритмами коррекции ошибок и могут найти применение в различных областях связи, обеспечивая высокие характеристики декодирования.

Для специалистов в области теории и техники кодирования, разработчиков систем связи, студентов и аспирантов соответствующих специальностей.

**ББК 32.811.4**

*Адрес издательства в Интернете [www.techbook.ru](http://www.techbook.ru)*

Справочное издание

**Золотарёв** Валерий Владимирович

**Овечкин** Геннадий Владимирович

## **ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ**

Методы и алгоритмы

Справочник

Компьютерная верстка Г. В. Овечкина  
Обложка художника В. Г. Ситникова

Подписано в печать 11.02.04. Формат 60x88/16.  
Бумага офсетная. Усл. печ. л. 7,9. Тираж 1500 экз. Изд. № 169.  
Электронное издание 04.10. 2014 г.

ISBN 5-93517-169-4

© В. В. Золотарёв, Г. В. Овечкин, 2004  
© Издательство «Горячая линия – Телеком», 2004, 2014

## **Введение**

Динамичный переход нашей технологической цивилизации на цифровые системы обработки и передачи информации создает много проблем при проектировании современных систем информатики и телекоммуникаций. Одной из важнейших задач, которые при этом необходимо решать во всех подобных системах, является обеспечение высокой достоверности передачи данных. Совершенно очевидно, что только при безусловно правильно переданных цифровых данных, поступивших на дальнейшую обработку, имеет смысл развивать весьма капиталоемкие технологии информатизации мирового сообщества.

К наиболее эффективным методам обеспечения высокого качества цифровой передачи в условиях высокого уровня шума канала относятся уже весьма мощные в настоящее время алгоритмы декодирования корректирующих кодов, в разработке которых теория помехоустойчивого кодирования, несомненно, имеет очень значительные успехи. За 50 лет своего интенсивного развития в технику связи были успешно внедрены пороговые декодеры, алгоритм Витерби, коды Рида-Соломона, каскадные схемы кодирования, а также разработки последнего времени: алгоритмы для турбо кодов и многопороговые декодеры (МПД). Однако требования к алгоритмам исправления ошибок в каналах с шумами, в частности, спутниковых каналах, непрерывно растут, и главная проблема – декодирование с эффективностью, близкой к оптимальной по энергетике канала, но при максимально простой реализации, еще до недавнего времени была весьма далека от своего успешного решения.

В предлагаемом вниманию читателей справочнике предлагаются конкретные схемы, реализующие возможности различных алгоритмов декодирования. За последнее десятилетие в технике помехоустойчивого кодирования произошли очередные революционные изменения, еще более значительные, чем в те далекие уже времена, когда были созданы первые аппаратные версии алгоритма Витерби и каскадные схемы с использованием этого алгоритма и декодеров для кодов Рида-Соломона. Те изменения в технологии микроэлектроники, которые еще 20 лет назад казались если и возможными, то в очень отдаленном буд-

дущем, сейчас обеспечивают возможность создания весьма мощных алгоритмов коррекции ошибок. К ним можно отнести турбо коды и другие близкие к ним структуры с алгоритмами их декодирования.

Другим еще более эффективным подходом к решению проблемы декодирования являются многопороговые декодеры (МПД), которые много лет успешно развиваются российскими специалистами. В справочнике эти методы представлены особенно широко как для блоковых, так и для сверточных кодов, в том числе и недвоичных.

Подчеркнем, что в настоящее время алгоритмы типа МПД могут декодировать длинные коды почти так же, как и оптимальные переборные алгоритмы. Однако сами МПД сохраняют при этом линейную сложность реализации. При одинаковой эффективности МПД выполняет примерно в 100 раз меньшее число операций, чем соответствующие турбо декодеры. При аппаратной реализации МПД может быть в некоторых случаях более быстрым, чем, например, турбо декодеры, почти на 3 десятичных порядка. Авторы полагают, что по совокупности характеристик эффективности, сложности реализации и быстродействия к уровню алгоритмов МПД в ближайшее время не смогут даже приблизиться никакие методы, известные в настоящее время.

Значительную роль в теории и технике кодирования играют каскадные методы кодирования и декодирования. Они практически всегда обеспечивают гораздо более высокий энергетический выигрыш (ЭВК) кодирования, чем те исходные базовые методы, из которых формируются затем сами каскадные коды. Этим и определяется важность каскадных методов кодирования и декодирования.

Справочник состоит из четырех глав. Первая глава содержит вводный материал, в котором описываются общие принципы работы системы передачи информации, рассматриваются наиболее часто используемые модели дискретных каналов связи и способы вычисления их основных характеристик. Также в данной главе даются базовые понятия помехоустойчивого кодирования и рассматриваются наиболее важные свойства линейных кодов. Во второй главе описываются блоковые линейные

коды, проводится анализ эффективности методов их декодирования, как с точки зрения обеспечиваемых вероятностных характеристик, так и сложности практической реализации. Третья глава посвящена методам декодирования сверточных кодов. В ней рассмотрены алгоритм декодирования Витерби, методы последовательного декодирования, а также пороговые и многопороговые алгоритмы декодирования. В последней четвертой главе представлены широко распространенные каскадные коды (каскадный код, состоящий из кода Рида-Соломона и сверточного кода, турбо коды). Кроме того, описаны совершенно новые способы каскадирования кодов, для которых применим многопороговый алгоритм декодирования, с другими простейшими кодами, которые существенно улучшают возможности методов многопорогового декодирования без существенно усложнения вычислений, что является еще одним важнейшим достоинством данных методов.

Авторы понимают, что фактически первая России попытка создания справочника по очень динамично изменяющейся тематике эффективного и простого декодирования в условиях предельно малых отношений сигнал-шум будет не лишена определенных недостатков. Многие коды в справочнике не описаны. В значительной мере это связано с тем, что их эффективность невелика по сравнению с турбо декодерами и многопороговыми алгоритмами, а также с каскадными схемами на их основе. Вполне возможно, что успехи в исследованиях этих и других более эффективных методов позволят обратить большее внимание и на эти разработки.

Тем не менее, наличие такого пособия для разработчиков программных и аппаратных средств обеспечения высокого уровня достоверности передачи данных, несомненно, поможет им хорошо ориентироваться в современной проблематике построения систем помехоустойчивого кодирования и покажет пути дальнейшего повышения эффективности декодеров.

Авторы просят направлять все возникающие у читателей вопросы и замечания по адресу: 390001, г. Рязань, ул. Гагарина, д. 59/1, РГРТА, деканат ФВТ или e-mail: [g\\_ovechkin@mail.ru](mailto:g_ovechkin@mail.ru), [zolatasd@yandex.ru](mailto:zolotasd@yandex.ru).

# 1. ОБЩИЕ СВЕДЕНИЯ

## 1.1. Элементы систем цифровой связи

Основной задачей помехоустойчивого кодирования является решение проблемы обеспечения высокой достоверности передаваемых данных за счет применения устройств кодирования/декодирования в составе системы передачи цифровой информации, структурная схема которой представлена на рис. 1.1 [15]. Данная схема широко используется в теории помехоустойчивого кодирования, поскольку она охватывает большинство ситуаций, которые встречаются на практике.

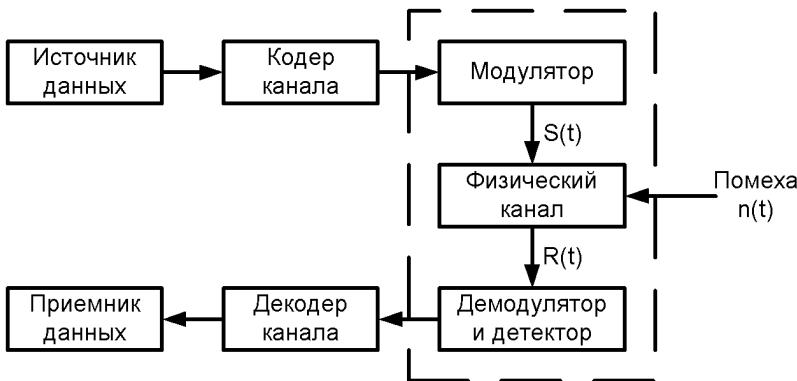


Рис. 1.1. Структурная схема системы передачи цифровой информации

Рассмотрим основные принципы работы представленной схемы. Сначала источник данных порождает данные в виде двоичных символов. Обычно предполагают, что «нули» и «единицы» появляются независимо друг от друга и с одинаковыми вероятностями. Затем кодер канала вносит в принятую информационную последовательность некоторую избыточность (данный процесс называется *кодированием*), которую декодер сможет использовать для исправления возникающих при передаче данных по каналу связи ошибок.

Закодированные данные с выхода кодера поступают на модулятор, который с помощью какого-либо метода модуляции

реализует их отображение в аналоговый сигнал  $S(t)$ . Модулятор может просто отобразить каждый двоичный символ в один из  $M=2$  возможных сигналов  $s_0(t)$  и  $s_1(t)$  (в этом случае говорят о *двоичной модуляции*), а может передавать  $q$ -битовые блоки ( $q>1$ ) при помощи  $M=2^q$  возможных сигналов (*M-позиционная модуляция*).

В физическом канале сигнал  $S(t)$  подвергается воздействию шума  $n(t)$ . Для количественной оценки степени влияния шума  $n(t)$  на сигнал  $S(t)$  обычно используют *отношение сигнал-шум*  $E_s/N_0$ , определяемое как отношение мощности сигнала  $P_C$  к мощности шума  $P_{\text{шум}}$ . Часто данное отношение выражается в децибелах, т.е.  $E_s/N_0=10\cdot\lg(P_C/P_{\text{шум}})$ . Заметим, что на всех приводимых в справочнике графиках отношение сигнал-шум выражено в децибелах, а во всех формулах используется безразмерная величина  $E_s/N_0=P_C/P_{\text{шум}}$ . Далее демодулятор преобразует принятый из канала сигнал  $R(t)$  в последовательность чисел, представляющих оценку переданных данных. После этого детектор квантует выход демодулятора на  $Q$ -уровней. В случае если  $Q=M$ , то говорят, что детектор выносит *жесткое решение* относительно переданных символов, если же  $Q>M$ , то детектор выносит *мягкие решения*.

Затем квантованный выход детектора поступает на декодер канала, который, используя внесенную кодером избыточность, определяет переданное источником сообщение (данний процесс называется *декодированием*).

## 1.2. Модели каналов связи

Наиболее важной частью структурной схемы системы передачи цифровой информации для кодера и декодера является *составной* или *дискретный канал*, заключенный на рис. 1.1 в штриховую рамку. Рассмотрим наиболее часто встречающиеся математические модели данного канала.

Самой простой является модель *двоичного симметричного канала* (ДСК), представляемая графом на рис. 1.2 и соответствующая случаю использования двоичной модуляции в канале с аддитивным шумом (т.е. каналу, в котором выходной сигнал  $R(t)$  равен сумме входного сигнала  $S(t)$  и шума  $n(t)$ ) и жесткого

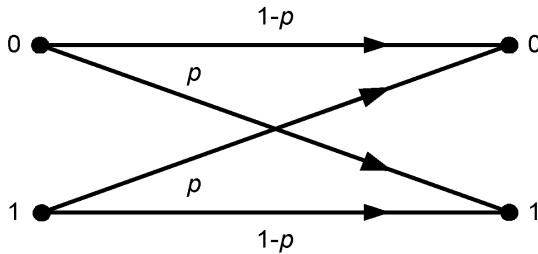


Рис. 1.2. Двоичный симметричный канал

решения демодулятора. Входом и выходом данного канала являются наборы  $X=\{0, 1\}$  и  $Y=\{0, 1\}$  из двух возможных двоичных символов. ДСК также характеризуется набором переходных вероятностей  $P(Y|X)$ , определяющих вероятность приема из канала символа  $Y$  при передаче символа  $X$ . Для ДСК переходные вероятности задаются выражениями:

$$\begin{aligned} P(Y=0|X=0) &= P(Y=1|X=1) = 1-p \\ P(Y=0|X=1) &= P(Y=1|X=0) = p \end{aligned}, \quad (1.1)$$

где  $p$  – средняя вероятность искажения символа.

Как следует из представленного описания, основной характеристикой ДСК является вероятность искажения символа  $p$ . Запишем выражение, связывающее данную величину с ранее упоминавшимся отношением сигнал-шум  $E_s/N_0$  для случая использования двух противоположных сигналов  $s_0(t) = -s_1(t)$ :

$$p = Q\left(\sqrt{2 \frac{E_s}{N_0}}\right), \quad (1.2)$$

где  $Q(x)$  – функция, определяемая по формуле

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t}{2}} dt. \quad (1.3)$$

Более общей моделью канала является *дискретный канал без памяти* (ДКБП). Входом данного канала являются  $q$ -ичные символы  $X=\{x_0, x_1, \dots, x_{q-1}\}$ , а выходом –  $Q$ -ичные символы

$Y=\{y_0, y_1, \dots, y_{Q-1}\}$ . Термин «без памяти» означает, что выходной символ канала зависит только от текущего входного символа. ДКБП характеризуется набором из  $qQ$  переходных вероятностей

$$P(y_i | x_i) = P(Y = y_i | X = x_i), \quad (1.4)$$

где  $i=0, 1, \dots, Q-1$  и  $j=0, 1, \dots, q-1$ . Для ДКБП переходные вероятности постоянны во времени и переходы различных символов независимы. Графическое представление данной модели канала показано на рис. 1.3.

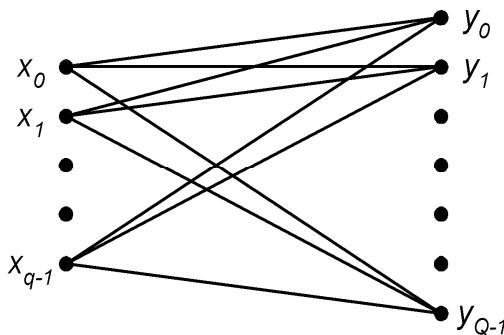


Рис. 1.3. Дискретный канал без памяти

Частным случаем ДКБП является  $q$ -ичный симметричный канал, называемый далее  $q$ СК. Данный канал, получающийся при использовании модулятором  $q$  ортогональных сигналов и вынесении жесткого решения демодулятором, характеризуется дискретным входом  $X=\{x_0, x_1, \dots, x_{q-1}\}$ , дискретным выходом  $Y=\{y_0, y_1, \dots, y_{q-1}\}$  и набором переходных вероятностей

$$P(y_i | x_j) = \begin{cases} 1 - P_q, & \text{если } i = j \\ \frac{P_q}{q-1}, & \text{если } i \neq j \end{cases}, \quad (1.5)$$

где  $P_q$  – вероятность искажения  $q$ -ичного символа, зависимость которой от отношения сигнал-шум выглядит так

$$P_q = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left[ 1 - \left( \frac{1}{\sqrt{2\pi}} \int_{-\infty}^y e^{-\frac{x^2}{2}} dx \right)^{q-1} \right] e^{-\frac{\left(y - \sqrt{2\frac{E_s}{N_0}}\right)^2}{2}} dy. \quad (1.6)$$

Последней рассматриваемой в справочнике моделью канала является канал с *аддитивным белым гауссовским шумом* (АБГШ), получающийся из ДКБП при бесконечном числе уровней квантования выхода детектора (т.е. квантование отсутствует,  $Q=\infty$ ). В данном случае шум является гауссовой случайной величиной с нулевым средним и дисперсией  $\sigma^2 = 1/(2E_s/N_0)$ . Таким образом, канал с АБГШ характеризуется дискретным входом  $X=\{x_0, x_1, \dots, x_{q-1}\}$ , непрерывным выходом  $Y=\{-\infty, +\infty\}$  и рядом переходных вероятностей

$$p(y | X = x_j) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-x_j)^2}{2\sigma^2}}, \quad j = 0, 1, \dots, q-1. \quad (1.7)$$

Для данной модели канала зависимость вероятности ошибки  $p$  от отношения сигнал-шум  $E_s/N_0$  определяется в соответствии с выражением (1.2).

### 1.3. Пропускная способность канала связи

Одной из основных характеристик канала является его *пропускная способность*  $C$ , которая определяется как максимальная средняя взаимная информация  $I(X;Y)$  между входом  $X$  и выходом  $Y$  канала. Важность этого понятия основана на теореме о кодировании для канала с шумами [25], которая утверждает, что если скорость передачи двоичных сообщений (или кодовая скорость  $r$ , определяемая в параграфе 1.4) меньше пропускной способности канала  $C$ , то с помощью правильно выбранных кодера и декодера можно вести передачу по каналу с шумом со сколь угодно малой вероятностью ошибки.

Для ДКБП пропускная способность вычисляется следующим образом:

$$\begin{aligned}
 C &= \max_{P(x_j)} I(X;Y) = \\
 &= \max_{P(x_j)} \sum_{j=0}^{q-1} \sum_{i=0}^{Q-1} P(x_j) P(y_i | x_j) \log [P(y_i | x_j) / P(y_i)]. \quad (1.8)
 \end{aligned}$$

Здесь максимизация осуществляется по всем возможным наборам вероятностей входных символов  $P(x_j)$ . В том случае, если канал является симметричным, максимизация достигается при равновероятных входных символах, т.е. при  $P(x_j)=1/q$ ,  $j=0, 1, \dots, q-1$ .

Для ДСК пропускная способность равна

$$C = 1 + p \log_2 p + (1-p) \log_2 (1-p). \quad (1.9)$$

Графики зависимости пропускной способности ДСК от вероятности ошибки  $p$  и от отношения сигнал-шум на символ  $E_s/N_0$  в канале связи представлены на рис. 1.4 и рис. 1.5.

Для канала с АБГШ пропускная способность  $C$  определяется так:

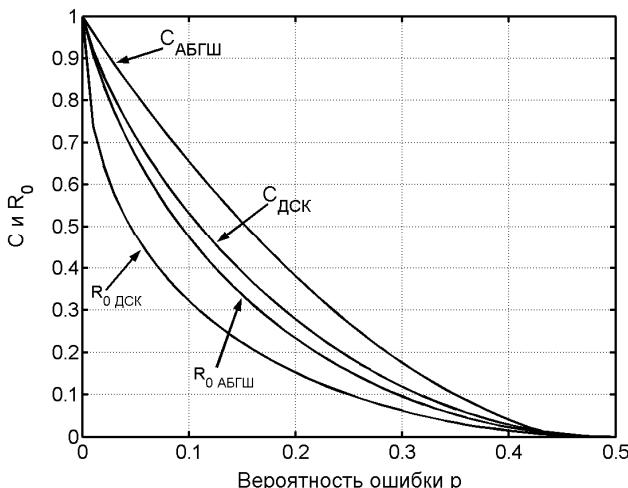


Рис. 1.4. Зависимость пропускной способности канала  $C$  и вычислительной скорости  $R_0$  от вероятности ошибки  $p$

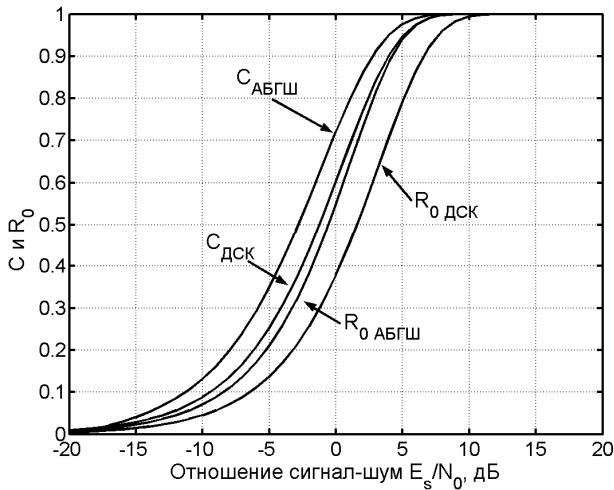


Рис. 1.5. Зависимость пропускной способности канала  $C$  и вычислительной скорости  $R_0$  от отношения сигнала-шума  $E_s/N_0$

$$C = \max_{P(x_i)} \sum_{i=0}^{q-1} \int_{-\infty}^{\infty} p(y|x_i)P(x_i) \log_2 \frac{p(y|x_i)}{p(y)} dy, \quad (1.10)$$

где  $p(y|X=x_i)$  вычисляется по формуле (1.7), а

$$p(y) = \sum_{k=0}^{q-1} p(y|x_k)P(x_k). \quad (1.11)$$

В случае работы в канале с АБГШ с двумя возможными входами  $X=\{-1, 1\}$  выражение (1.10) преобразуется к виду

$$\begin{aligned} C = & \frac{1}{2} \int_{-\infty}^{\infty} p(y|1) \log_2 \frac{p(y|1)}{p(y)} dy + \\ & + \frac{1}{2} \int_{-\infty}^{\infty} p(y|-1) \log_2 \frac{p(y|-1)}{p(y)} dy, \end{aligned} \quad (1.12)$$

где  $p(y) = \frac{1}{2}(p(y|1) + p(y|-1))$ .

Графики зависимости пропускной способности канала с АБГШ с двумя возможными входами от вероятности ошибки  $p$  и от отношения сигнал-шум  $E_s/N_0$  в канале связи также представлены на рис. 1.4 и рис. 1.5. Отметим, что одинаковой пропускной способности с ДСК канал с АБГШ достигает при отношении сигнал-шум примерно на 2 дБ меньшем, что показывает существенное преимущество использования мягких решений демодулятора.

Для  $q$ -ичного симметричного канала пропускная способность вычисляется следующим образом:

$$C = \frac{1}{q} \sum_{j=0}^{q-1} \sum_{i=0}^{q-1} P(y_i | x_j) \log [P(y_i | x_j) / P(y_i)], \quad (1.13)$$

где  $P(y_i | x_j)$  вычисляется по формуле (1.5), а

$$P(y_i) = \sum_{j=0}^{q-1} P(x_j) P(y_i | x_j) = \frac{1}{q}.$$

Графики зависимости пропускной способности  $q$ СК от вероятности символьной ошибки  $P_q$  и отношения сигнал-шум  $E_s/N_0$  представлены на рис. 1.6 и рис. 1.7 соответственно.

Еще одной важной характеристикой канала (особенно для методов последовательного декодирования, рассматриваемых в параграфе 3.2) является его *вычислительная скорость*  $R_0$  (иначе называемая параметром экспоненциальной границы), используемая при определении границы случайного кодирования

$$\bar{P}_s < 2^{-n(R_0 - R)}, \quad (1.14)$$

показывающей, что если использовать достаточно длинные коды, то при кодовой скорости  $R$ , меньшей вычислительной скорости  $R_0$ , среднюю по всевозможным кодам вероятность ошибки  $\bar{P}_s$  при использовании достаточно длинных кодов можно сделать сколь угодно малой.

Для ДКБП вычислительная скорость рассчитывается следующим образом:

$$R_0 = \max_{\{P(x_j)\}} \left\{ -\log_2 \sum_{i=0}^{Q-1} \left[ \sum_{j=0}^{q-1} P(x_j) \sqrt{P(y_i | x_j)} \right]^2 \right\}. \quad (1.15)$$

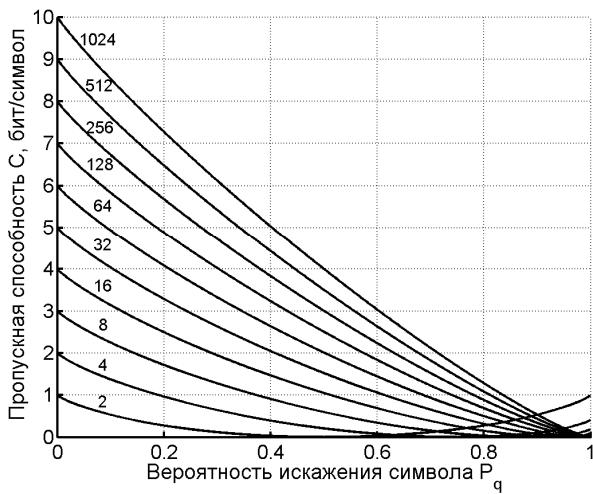


Рис. 1.6. Пропускная способность  $q$ -ичного канала с ортогональными сигналами для  $q$  от 2 до 1024

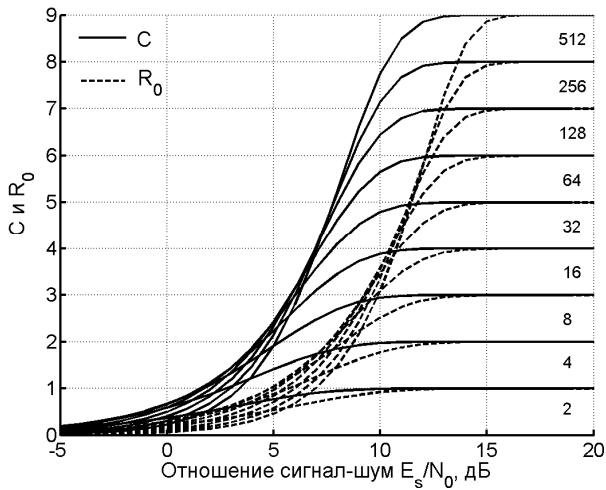


Рис. 1.7. Пропускная способность  $q$ -ичного канала с ортогональными сигналами для  $q$  от 2 до 512

При работе в ДСК выражение (1.15) принимает вид

$$R_0 = -\log_2 \left( \frac{1}{2} \left[ \sqrt{1-p} + \sqrt{p} \right]^2 \right). \quad (1.16)$$

Для канала с АБГШ и двоичным входом выражение для определения вычислительной скорости  $R_0$  выглядит следующим образом:

$$R_0 = 1 - \log_2 \left( 1 + e^{-E_s / N_0} \right). \quad (1.17)$$

Для  $q$ СК с ортогональными сигналами и жестким решением демодулятора вычислительная скорость равна

$$R_0 = -\log_2 \left( \frac{1}{q} \left[ \sqrt{1-P_q} + \sqrt{P_q(q-1)} \right]^2 \right). \quad (1.18)$$

Графики зависимости вычислительной скорости для вышеописанных моделей каналов связи представлены на рис. 1.4, рис. 1.5 и рис. 1.7 кривыми  $R_0$ .

Полезно получить зависимости пропускной способности канала  $C$  и вычислительной скорости  $R_0$  от отношения сигнал-шум на информационный бит (далее именуемое отношение сигнал-шум на бит), определяемого как отношение энергии, приходящейся на один информационный символ к спектральной мощности шума  $E_b/N_0$ . Для этого сначала необходимо нормализовать зависимость  $C(E_s/N_0)$  делением на  $N/2$  (здесь  $N$  – размерность пространства, используемого для передачи символов)

$$C = \frac{C(E_s / N_0)}{N/2},$$

а затем вычислить соответствующее полученному  $C$  значение  $E_b/N_0$ :

$$E_b / N_0 = C E_s / N_0.$$

Аналогично можно получить зависимость вычислительной скорости  $R_0$  от отношения сигнала-шум на бит  $E_b/N_0$ . Для примера на рис. 1.8 представлены графики зависимости  $C(E_b/N_0)$  и  $R_0(E_b/N_0)$  для ДСК и канала с АБГШ.

Более детальную информацию о расчете пропускной способности и вычислительной скорости различных моделей каналов связи можно найти в книге Дж. Прокиса [20].

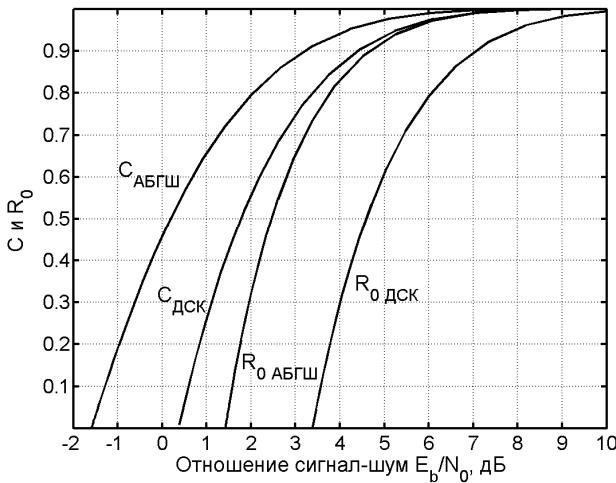


Рис. 1.8. Зависимость пропускной способности канала  $C$  и вычислительной скорости  $R_0$  от отношения сигнала-шума на бит  $E_b/N_0$

#### 1.4. Помехоустойчивые коды

На сегодняшний день известно много различных классов помехоустойчивых кодов, отличающихся друг от друга структурой, функциональным назначением, энергетической эффективностью, алгоритмами кодирования и декодирования и многими другими параметрами. Рассмотрим основные подходы к классификации данных кодов, представленные на рис. 1.9.

При одном подходе коды можно разбить на две группы: *блоковые* [19], в которых кодирование и декодирование производится в пределах кодовой комбинации или блока, и *древовидные* [18], в которых обработка символов производится непрерывно, без разделения на блоки. Кодер для блокового кода является устройством без памяти, отображающим последовательности из  $k$  входных символов в последовательности из  $n$  выходных символов. Термин «без памяти» указывает, что каждый вы-

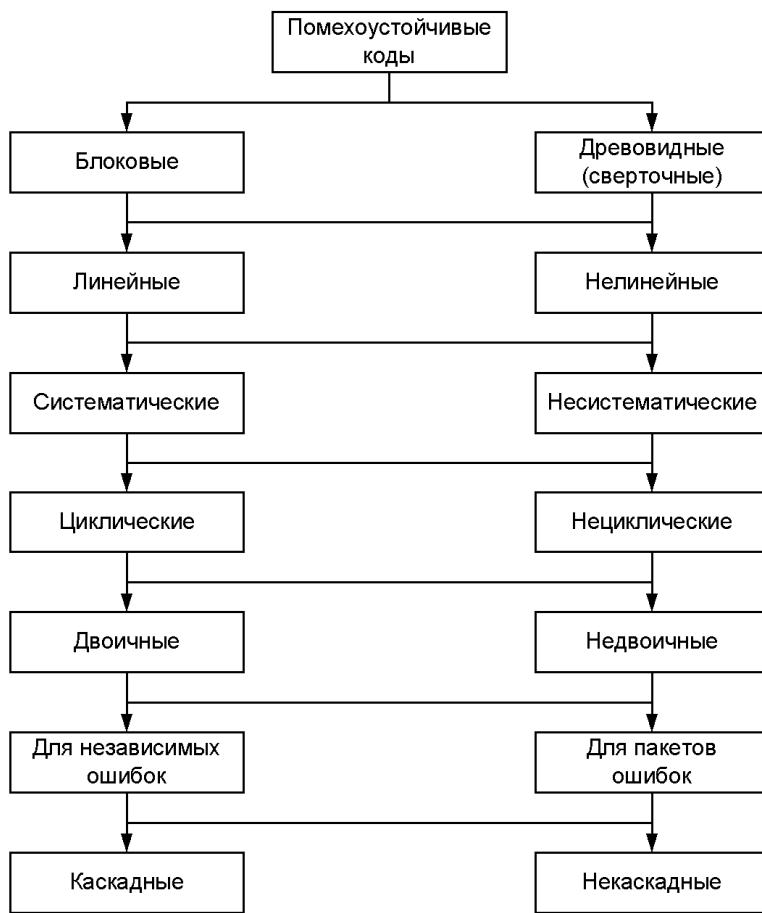


Рис. 1.9. Классификация помехоустойчивых кодов

ходной блок из  $n$  символов зависит только от соответствующего входного блока из  $k$  символов и не зависит от других блоков. Основными параметрами блокового кода являются *длина кода*  $n$ , *длина информационной последовательности*  $k$ , *скорость кода*  $r=k/n$  и *минимальное кодовое расстояние*  $d_{min}$ . Последний пара-

метр определяется минимальным расстоянием (расстоянием Хэмминга) между любыми двумя кодовыми словами длины  $n$ .

Кодер для древовидного кода является устройством с памятью, в которое поступают наборы из  $k_0$  входных символов, а на выходе появляются наборы из  $n_0$  выходных символов. Каждый набор  $n_0$  выходных символов зависит от текущего входного набора и от  $K-1$  предыдущих входных наборов. Параметр  $K$  называется *конструктивной длиной* сверточного кода, а величину  $n_A=Kn_0$  – *длиной кодового ограничения*. Древовидные коды характеризуются также скоростью  $r=k_0/n_0$  и *свободным расстоянием*  $d_{free}$ , смысл которого будет пояснен далее при описании сверточных кодов.

Из вышеизложенного описания видно, что в процессе работы кодер помехоустойчивого кода в соответствии с некоторыми правилами осуществляет обработку входных символов. В зависимости от количества возможных значений  $q$  каждого из символов все коды можно разделить на *двоичные* (при  $q=2$ ) и *недвоичные* (при  $q>2$ ).

Еще при одном подходе коды можно разделить на *линейные* и *нелинейные* [19]. Линейные коды образуют векторное пространство и обладают следующим важным свойством: два кодовых слова можно сложить, используя поддающее определение суммы, и получить третье кодовое слово. В случае обычных двоичных кодов эта операция является посимвольным сложением двух кодовых слов по модулю 2. Данное свойство существенно упрощает процедуры кодирования и декодирования, а также задачу вычисления параметров кода, поскольку минимальное расстояние между двумя кодовыми словами при этом эквивалентно минимальному расстоянию между кодовым словом, состоящим целиком из нулей, и некоторым другим кодовым словом. Кроме того, при вычислении характеристик линейного кода достаточно рассмотреть, что происходит при передаче кодового слова, состоящего целиком из нулей. Линейные древовидные коды обычно называют *сверточными*.

Помехоустойчивые коды также можно разбить на коды, исправляющие *случайные* или *независимые* ошибки, и коды, исправляющие *пакеты* ошибок. На практике, в основном, приме-

няются коды, исправляющие случайные ошибки, поскольку для исправления пакетов ошибок часто оказывается легче использовать коды для исправления независимых ошибок вместе с устройствами перемежения и восстановления. Первое из них осуществляет перемешивание порядка символов в закодированной последовательности перед передачей в канал, а второе – восстановление исходного порядка символов после приема. При правильном проектировании данных устройств можно считать, что образующиеся в канале связи пакеты ошибок перед декодированием будут разбиты на случайные ошибки.

Почти все схемы кодирования, применяемые на практике, основаны на линейных кодах, поэтому более подробно рассмотрим способ их описания [1]. Блоковый код длины  $n$  с  $2^k$  словами называется линейным  $(n, k)$  кодом, если его кодовые слова образуют  $k$ -мерное подпространство  $V_k$  векторного  $n$ -мерного пространства  $V_n$ . Подпространство  $V_k$  порождается базисом из  $k$  линейно независимых векторов, которые образуют строки *порождающей* матрицы  $(n, k)$  кода

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}. \quad (1.19)$$

Процесс кодирования блокового кода состоит в разбиении информационной последовательности на сообщения  $\bar{U} = (u_0, u_1, \dots, u_{k-1})$  длины  $k$  и отображении этих сообщений в кодовые слова  $\bar{C} = (c_0, c_1, \dots, c_{n-1})$  длины  $n$  следующим образом:

$$\bar{C} = \bar{U} G. \quad (1.20)$$

Часто кодовые слова лучше представлять в *систематической* форме  $\bar{C} = (\bar{U}, \bar{V})$ , образуя отдельно информационную часть  $\bar{U}$  из  $k$  символов и проверочную часть  $\bar{V}$  из  $m=n-k$  символов. Порождающая матрица такого систематического кода будет иметь вид

$$G = [I_k P] = \begin{bmatrix} 1 & 0 & \dots & 0 & g_{00} & g_{01} & \dots & g_{0,n-k-1} \\ 0 & 1 & \dots & 0 & g_{10} & g_{11} & \dots & g_{1,n-k-1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & g_{k-1,0} & g_{k-1,1} & \dots & g_{k-1,n-k-1} \end{bmatrix}. \quad (1.21)$$

В данном случае матрица  $G$  содержит единичную матрицу  $I_k$  размером  $k \times k$ , формирующую информационную часть слова, и матрицу  $P$  размером  $k \times (n-k)$ , определяющую проверочные символы.

С порождающей матрицей тесно связано понятие *проверочной* матрицы  $H$ , пространство строк которой ортогонально пространству строк порождающей матрицы, т. е.

$$G H^T = 0. \quad (1.22)$$

Отметим, что каждое кодовое слово линейного кода  $\bar{C}$  также будет удовлетворять условию ортогональности, так как

$$\bar{C} H^T = \bar{U} G H^T = 0. \quad (1.23)$$

Если порождающая матрица задана в форме (1.21), то для выполнения условия ортогональности проверочная матрица  $H$  должна иметь вид

$$H = \left| P^T I_{n-k} \right|. \quad (1.24)$$

Проверочная матрица обычно используется в процессе декодирования кода следующим образом. Пусть при передаче по каналу из-за влияния шумов  $\bar{E}$  кодовые символы  $\bar{C}$  искажаются. В результате принятые слова имеют вид  $\bar{Y} = \bar{C} + \bar{E}$ , где  $\bar{Y} = (y_0, y_1, \dots, y_{n-1})$  – принятый вектор, а  $\bar{E} = (e_0, e_1, \dots, e_{n-1})$  – вектор ошибок. Для обнаружения или исправления ошибок на приемной стороне сначала вычисляют *синдром*

$$\bar{S} = \bar{Y} H^T = (s_0, s_1, \dots, s_{n-k-1}), \quad (1.25)$$

зависящий только от вектора ошибок, поскольку

$$\bar{S} = \bar{Y} H^T = (\bar{C} + \bar{E}) H^T = \bar{C} H^T + \bar{E} H^T = \bar{E} H^T. \quad (1.26)$$

Если  $\bar{S} = 0$ , то принятое слово  $\bar{Y}$  принадлежит к множеству кодовых слов, иначе слово  $\bar{Y}$  содержит ошибки, и по значениям

символов синдрома  $\bar{S}$  определяют их местоположение. Данный принцип лежит в основе методов синдромного декодирования.

Важный подкласс линейных кодов составляют *циклические коды*, обладающие циклическими свойствами: если  $\bar{C} = (c_0, c_1, \dots, c_{n-1})$  – кодовое слово циклического кода, тогда и  $\bar{C}' = (c_1, c_2, \dots, c_{n-1}, c_0)$ , полученное циклическим сдвигом элементов  $\bar{C}$ , также является кодовым словом. Эти коды обладают большим количеством структурных особенностей, которые значительно упрощают реализацию операций кодирования и декодирования.

Порождающую матрицу циклического кода можно записать в виде

$$G = \begin{bmatrix} g_0 & g_1 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_{n-k} & \dots & 0 \\ \dots & \dots & \dots & & \dots & \dots & \dots \\ 0 & \dots & 0 & g_0 & g_1 & \dots & g_{n-k} \end{bmatrix}. \quad (1.27)$$

Для компактной записи циклического  $(n, k)$  кода часто используется *порождающий или образующий полином*  $g(x)$  степени  $n-k$  с коэффициентами из поля  $GF(q)$  (для двоичных кодов  $g_i = \{0,1\}$ )

$$g(x) = g_0 + g_1x + g_2x^2 + \dots + g_{n-k}x^{n-k}. \quad (1.28)$$

Если в виде полинома записать информационное сообщение  $U(x)$  и кодовое слово  $C(x)$ , то кодовые слова кода образуются путем умножения сообщения на порождающий полином

$$C(x) = U(x)g(x). \quad (1.29)$$

Обычно для удобства записи коэффициенты образующих полиномов объединяют в двоичное слово и представляют в восьмеричной системе счисления. Также образующие полиномы можно записывать путем перечисления степеней с ненулевыми коэффициентами. Например, запись  $g(x) = 1 + x + x^4 + x^6$  равносильна следующим:  $g = 1010011_2 = 123_8$ ;  $g = (0, 1, 4, 6)$ .

Для примера на рис. 1.10 показан кодер систематического блокового кода, соответствующий порождающей матрице вида

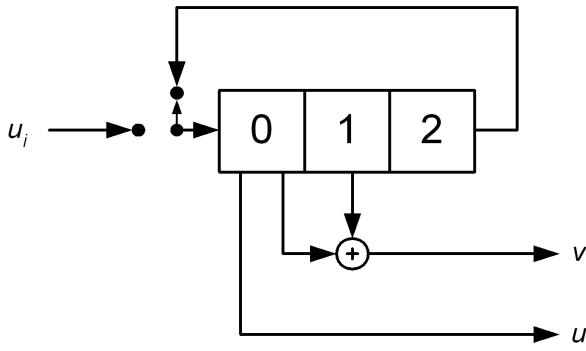


Рис. 1.10. Кодер блокового систематического кода

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (1.30)$$

Проверочная матрица данного кода в соответствии с (1.24) записывается следующим образом:

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (1.31)$$

Получившийся код характеризуется следующими параметрами: количество кодовых символов  $n=6$ , количество информационных символов  $k=3$ , кодовая скорость  $r=1/2$ , минимальное кодовое расстояние  $d_{min}=3$ .

Далее рассмотрим способы описания сверточных кодов. Для построения кодера сверточного кода необходимо  $k_0$  регистров сдвига, связи между элементами которых определяются набором образующих полиномов  $g_{(i,j)}(x)$ , где  $i=0, 1, \dots, k_0-1$  – номер входного потока, а  $j=0, 1, \dots, n_0-1$  – номер выходного потока. Поскольку на практике чаще используются коды с единственным входным потоком ( $k_0=1$ ), индекс  $i$  обычно опускается. Еще одной характеристикой сверточного кода является его кон-

структуривная длина  $K$ , влияющая на сложность его декодирования и равная длине самого длинного регистра сдвига.

Для примера на рис. 1.11 представлен кодер несистематического сверточного кода, заданный образующими полиномами  $g_0(x)=1+x^2$  и  $g_1(x)=1+x+x^2$ . Данный код характеризуется следующими параметрами: количество информационных ветвей  $k_0=1$ , количество проверочных ветвей  $n_0=2$ , кодовая скорость  $r=1/2$ , длина кодового ограничения  $n_A=6$ , конструктивная длина кода  $K=3$ .

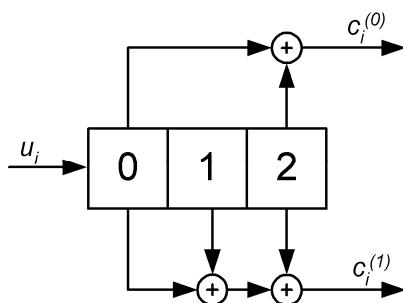


Рис. 1.11. Кодер сверточного несистематического кода

Так же как и блоковые, сверточные коды можно представить в систематической форме, частным случаем которой являются *рекурсивные систематические сверточные* (Recursive Systematic Convolutional – RSC) коды. Характерной особенностью кодеров данных кодов является наличие обратной связи. В качестве примера на рис. 1.12 представлен RSC кодер, заданный полиномом обратной связи (feedback polynomial)  $g_0(x)=1+x+x^2$  и выходным полиномом (feedforward polynomial)  $g_1(x)=1+x^2$ .

Удобным способом представления процесса функционирования кодера сверточных кодов является *кодовое дерево*. Пример кодового дерева для кодера с рис. 1.11 показан на рис. 1.13. Как видно из рисунка, кодовое дерево состоит из узлов, связанных между собой ребрами. При этом каждому узлу соответствует од-

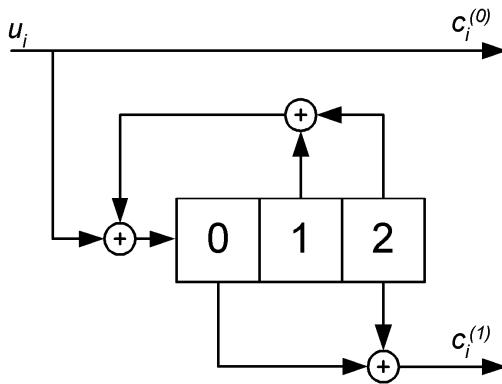


Рис. 1.12. Кодер рекурсивного систематического сверточного кода

но из  $2^{(K-1)k_0}$  состояний кодера (в данном случае 4), определяемое содержимым всех, кроме самых правых, ячеек регистров сдвига кодера (десятичное представление содержимого этих ячеек показано на рисунке в виде меток 0, 1, 2 и 3), а каждое ребро характеризуется входной и выходной последовательностью (верхнему ребру соответствует входной символ 0, нижнему – 1, а выходная последовательность представлена метками 00, 01, 10 и 11). В результате любая входная последовательность задает некоторый путь на дереве, по которому можно определить выходную последовательность. Например, входной последовательности 11001 соответствует путь, показанный на рис. 1.13 жирной линией и задающий выходную последовательность 1110101111.

Недостатком данного представления сверточных кодов является экспоненциальный рост числа ветвей с ростом длины входной последовательности. Поэтому часто кодовое дерево изображают в виде кодовой решетки, при построении которой учитывается конечное число возможных состояний кодера, что дает возможность отождествлять вершины дерева, характеризуемые одинаковыми состояниями. Изображение кодовой решетки для вышеприведенного примера и путь, соответствующий входной последовательности 11001, показаны на рис. 1.14.

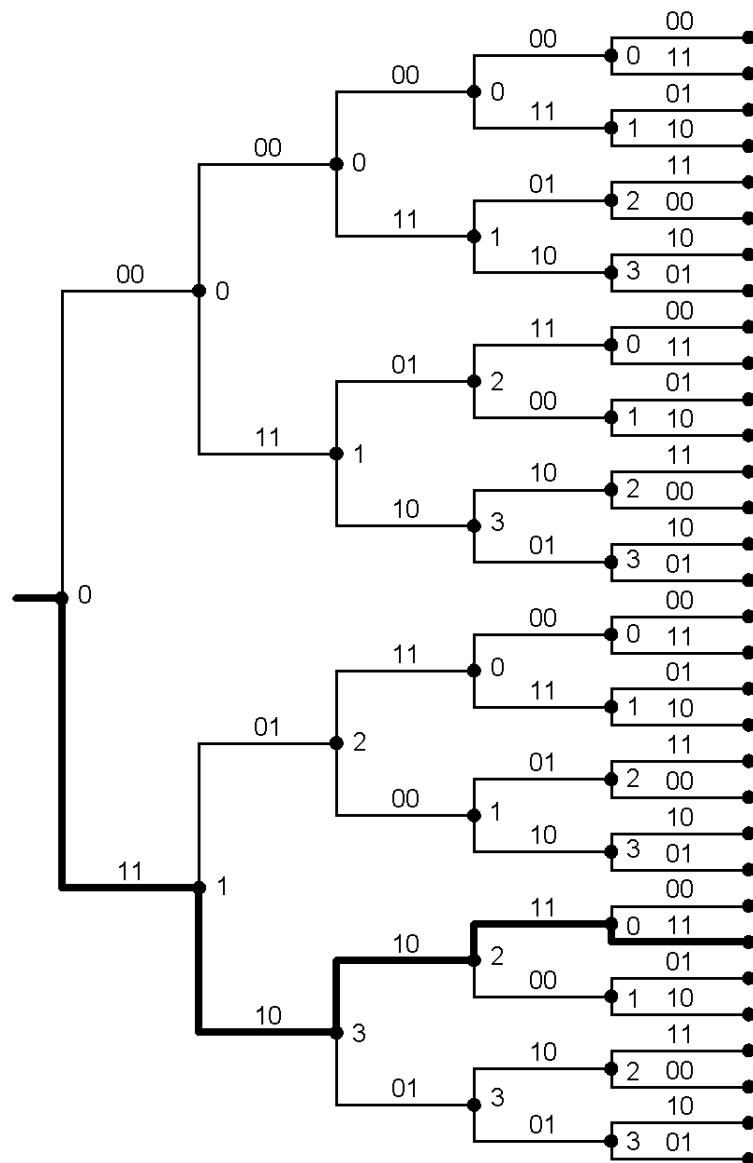


Рис. 1.13. Кодовое дерево сверточного кода

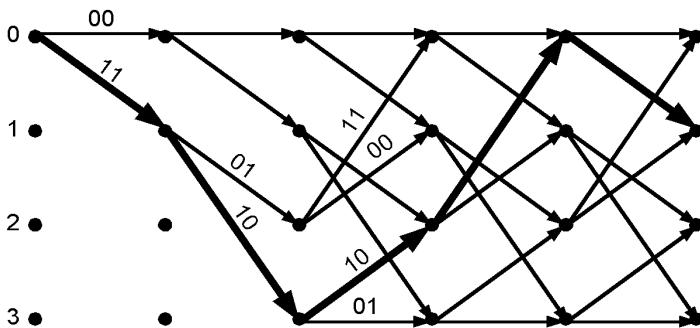


Рис. 1.14. Решетчатая диаграмма сверточного кода

Используя решетчатое представление, определим ранее встречавшееся понятие минимального свободного расстояния сверточного кода  $d_{free}$  как минимальное расстояние между двумя различными путями по кодовой решетке, начинающимися и заканчивающимися в одном состоянии. Например, из рис. 1.14 следует, что минимальное свободное расстояние рассматриваемого кода равно 5 (общий вес пути 0→1→2→0, заданного вершинами на диаграмме).

Заметим, что далее, в случае применимости выводов, формул и т.д. как к сверточным, так и блоковым кодам, вместо минимального кодового расстояния  $d_{min}$  и свободного расстояния  $d_{free}$  используется общий термин кодовое расстояние  $d$ .

Теперь рассмотрим основные характеристики помехоустойчивости линейных кодов и методы их оценки.

## 1.5. Основные характеристики методов коррекции ошибок

Для оценки помехоустойчивости системы связи часто используют среднюю вероятность ошибки в информационном бите  $P_b$  или последовательности бит (кодовом блоке)  $P_B$  при определенном отношении сигнал-шум в канале связи. Данные вероятности ошибки могут быть определены как по точным или приближенным формулам, так и получены с помощью статистического моделирования системы связи на ЭВМ.

Для оценки вероятностей ошибки в бите  $P_b$  и блоке  $P_B$  линейного блокового кода длиной  $n$ , декодируемого с помощью декодера *максимального правдоподобия* (т.е. декодера, выбирающего из всех возможных кодовых слов то, которое находится на минимальном расстоянии от принятой из канала последовательности), можно воспользоваться аддитивной оценкой [14], характеризующейся достаточной точностью при большом отношении сигнал-шум

$$\begin{aligned} P_B &\leq \sum_{j=1}^n N_j P_c(j) \\ P_b &\leq \sum_{j=1}^n \frac{j}{k} N_j P_c(j), \end{aligned} \quad (1.32)$$

где  $k$  – количество информационных символов кода,  $N_j$  – число кодовых слов веса  $j$ ,

$$P_c(j) = \begin{cases} \sum_{i=(j+1)/2}^j C_j^i p^i (1-p)^{j-i}, & -\text{ДСК, } j \text{ нечетно} \\ \frac{1}{2} C_j^{j/2} p^{j/2} (1-p)^{j/2} + \sum_{i=(j/2)+1}^j C_j^i p^i (1-p)^{j-i}, & -\text{ДСК, } j \text{ четно} \\ Q\left(\sqrt{2 j E_s / N_0}\right), & -\text{канал с АБГШ} \end{cases} \quad (1.33)$$

где  $p$  – вероятность битовой ошибки на входе декодера (т.е. в канале),  $E_s/N_0$  – отношение сигнал-шум в канале,  $Q(x)$  – функция, определяемая формулой (1.3).

Недостатком данных оценок вероятности ошибки является их неточность при малом отношении сигнал-шум и необходимость определения спектра кода, что часто является достаточно трудной задачей. Поэтому для некоторых кодов при их описании будут рассмотрены более простые и точные границы вероятностей ошибки.

Еще одной, не менее важной характеристикой, является энергетический выигрыш кодирования (ЭВК), показывающий величину снижения энергии, необходимой для передачи одного бита данных при некоторой выбранной средней вероятности ошибки на бит  $P_b$  в случае использования тех или иных алго-

ритмов кодирования и декодирования, по сравнению со случаем, когда кодирования нет. Зарубежные специалисты оценивали каждый децибел ЭВК в миллионы долларов более 20 лет назад [2]. Сейчас ценность ЭВК еще более возросла, поскольку он позволяет уменьшать размеры очень дорогих антенн, повышать дальность связи, увеличивать скорость передачи, снижать необходимую мощность передатчика и улучшать другие важные свойства систем связи.

При оценке эффективности алгоритмов декодирования различных кодов полезно знать предельные (или асимптотические) для малого шума характеристики данных кодов.

Асимптотический ЭВК зависит только от скорости кода  $r$  и кодового расстояния  $d$ , и для системы с двоичной фазовой модуляцией при приеме без квантования он равен

$$G_a = 10 \cdot \lg(Rd). \quad (1.34)$$

В случае использования двоичного симметричного канала

$$G_a = 10 \cdot \lg(R(t+1)), \quad (1.35)$$

где  $t$  – максимальное целое, меньшее, чем  $d/2$ , определяющее число исправляемых кодом ошибок. Из сопоставления выражений (1.34) и (1.35) видно, что при приеме в целом в пределе обеспечивается на 3 дБ больший ЭВК, чем при использовании жестких решений демодулятора. Однако при реальных отношениях сигнал-шум и умеренных значениях  $d$  разница обычно близка к 1÷2 дБ. Использование же мягкого модема с квантованием выходного значения на 8 и 16 уровней уменьшает ЭВК по сравнению с приемом в целом примерно на 0.25 и 0.1 дБ соответственно.

Среди других характеристик алгоритмов декодирования выделим сложность их реализации (как программной, так и аппаратной). Данная характеристика имеет чрезвычайно большое значение, так как применяя очень сложные методы декодирования можно получить весьма высокие значения ЭВК, но при этом практическое использование данных алгоритмов будет невозможно.

В следующих главах рассмотрим наиболее важные с точки зрения авторов коды и методы их декодирования.

## 2. БЛОКОВЫЕ КОДЫ И МЕТОДЫ ИХ ДЕКОДИРОВАНИЯ

### 2.1. Коды Хэмминга

Одними из самых простых являются *коды Хэмминга*, представленные Хэммингом в 1950 г. [44]. К данным кодам относятся линейные блоковые коды с параметрами  $(n, k)$  вида  $(2^m-1, 2^m-m-1)$ , где  $m=n-k$  – число проверочных символов кода. Коды Хэмминга обладают кодовым расстоянием  $d_{min}=3$  и поэтому способны исправлять только одну или обнаружить две ошибки.

Для задания кодов Хэмминга обычно используется проверочная матрица  $H$ , содержащая  $m$  строк и  $2^m-1$  столбцов, причем столбцами являются все возможные ненулевые двоичные векторы длины  $m$ . Для примера приведем проверочную матрицу  $(7, 4)$  кода Хэмминга, записанную в виде (1.24):

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (2.1)$$

Порождающая матрица данного кода формируется из единичной матрицы размером  $k \times k$  и транспонированной подматрицы, состоящей из первых  $k$  строк матрицы  $H$ :

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \quad (2.2)$$

Коды Хэмминга являются одними из немногих кодов, для которых полностью известно распределение весов кодовых слов, определяемое весовой функцией

$$N(x) = \sum_{i=0}^n N_i x^i, \quad (2.3)$$

где  $N_i$  – число кодовых слов веса  $i$ . Для кодов Хэмминга с  $d_{min}=3$  и  $n=2^m-1$  весовая функция имеет вид [19]

$$N(x) = \frac{1}{n+1} \left[ (1+x)^n + n(1+x)^{(n-1)/2} (1-x)^{(n+1)/2} \right]. \quad (2.4)$$

Знание спектра кода, как было показано в параграфе 1.5, может оказаться полезным при получении границ его эффективности.

Кроме описанных кодов Хэмминга существуют так называемые *укороченные* и *расширенные коды Хэмминга*. Укороченные коды Хэмминга получаются при исключении какой-либо проверки (т.е. удалении одной строки проверочной матрицы). Расширенные коды Хэмминга получаются путем введения дополнительной проверки на четность всех символов кодового слова. В результате их минимальное расстояние увеличивается до  $d_{min}=4$ , что позволяет данным кодам исправлять одну и обнаруживать две или только обнаруживать три ошибки. Проверочная матрица  $H$  расширенного кода Хэмминга, заданного матрицей (2.1), имеет вид

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (2.5)$$

Для удобства работы данную матрицу можно привести к канонической (систематической) форме (1.24) с помощью приведения к последней строке всех остальных строк:

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.6)$$

Спектр расширенных кодов Хэмминга задается весовой функцией

$$N(x) = \frac{1}{2n} \left[ (1+x)^n + (1-x)^n + 2(n-1)(1-x^2)^{n/2} \right]. \quad (2.7)$$

Далее рассмотрим характеристики кодов Хэмминга. На рис. 2.1 и рис. 2.2 представлены экспериментальные зависимости вероятности битовой ошибки  $P_b$  от вероятности ошибки в канале  $p$  и от отношения сигнал-шум на бит  $E_b/N_0$  для кодов Хэмминга с  $m=3 \div 7$  при работе в ДСК, декодируемых с помощью

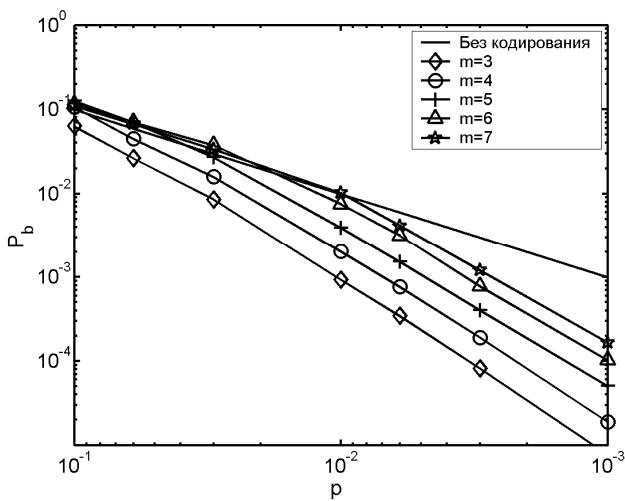


Рис. 2.1. Характеристики кодов Хэмминга в ДСК для различных значений  $m$

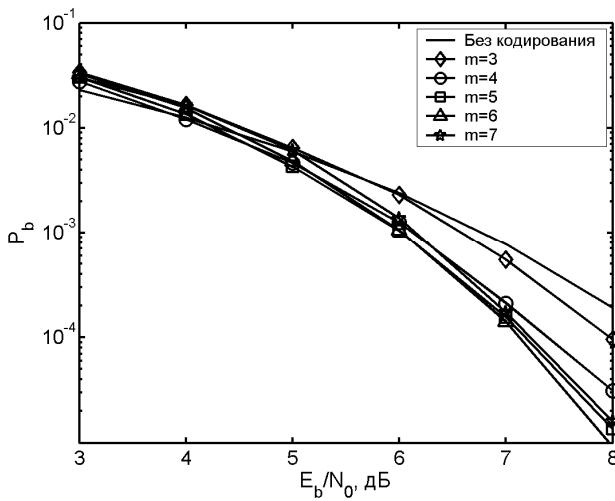


Рис. 2.2. Характеристики кодов Хэмминга в ДСК для различных значений  $m$

декодера Меггита, описываемого в параграфе 2.5. Заметим, что между вероятностью ошибки  $p$  в канале и отношением сигнал-шум на бит  $E_b/N_0$  существует следующая зависимость:

$$p = Q\left(\sqrt{2\frac{E_s}{N_0}}\right) = Q\left(\sqrt{2r\frac{E_b}{N_0}}\right), \quad (2.8)$$

где  $Q(x)$  – функция, определенная (1.3),  $r$  – кодовая скорость кода ( $r=k/n$ ).

Как видно из представленных рисунков, коды Хэмминга обладают очень слабой корректирующей способностью и отдельно практически не используются. Однако применение данных кодов в составе каскадных схем кодирования (например, Turbo Product Codes [49, 60]) позволяет получить очень хорошие результаты.

Более подробное описание кодов Хэмминга можно найти в [15, 19].

## 2.2. Коды Боуза-Чоудхури-Хоквингема

*Коды Боуза-Чоудхури-Хоквингема* (БЧХ) [3] представляют собой класс линейных циклических кодов, исправляющих кратные ошибки, и являются обобщением ранее описанных кодов Хэмминга. Коды БЧХ обычно задаются через корни порождающего многочлена  $g(x)$  степени  $n-k$ . Данные коды определяются представленным ниже образом [15].

*Примитивным кодом БЧХ*, исправляющим  $t$  ошибок, называется блоковый код длиной  $n = q^m - 1$  над полем Галуа  $GF(q)$ , для которого элементы  $\alpha^{m_0}, \alpha^{m_0+1}, \dots, \alpha^{m_0+2t-1}$  (для произвольного  $m_0$ ) являются корнями порождающего многочлена  $g(x)$ , где  $\alpha$  – примитивный элемент поля  $GF(q^m)$ .

Основываясь на представленном определении порождающий многочлен кода БЧХ можно записать в виде

$$g(x) = HOK(m_{m_0}(x), m_{m_0+1}(x), \dots, m_{m_0+2t-1}(x)), \quad (2.9)$$

где  $m_c(x)$  – минимальная функция  $\alpha^c$  в поле  $GF(q)$ , причем в случае двоичных кодов БЧХ наименьшее общее кратное ищется только для минимальных функций с нечетными индексами.

*Непримитивные коды БЧХ* определяются аналогично, но примитивный элемент  $\alpha$  заменяется непримитивным элементом  $\beta$  поля  $GF(q^n)$ , и длина блока становится равной порядку  $\beta$ .

Для большого количества практически используемых значений  $n$ ,  $k$  и  $t$  порождающие полиномы уже получены, и их можно найти, например, в [15, 19].

Можно получить асимптотические оценки зависимости вероятности ошибки в кодовом блоке  $P_B$ ,  $q$ -ичном символе  $P_s$  и бите  $P_b$  при декодировании кодов БЧХ по максимуму правдоподобия. Данные оценки не требуют знания спектра кода и имеют вид [15, 20]

$$P_B = \sum_{i=t+1}^n C_n^i P_q^i (1 - P_q)^{n-i}, \quad (2.10)$$

$$P_s \leq \frac{1}{n} \sum_{i=t+1}^n (i+t) C_n^i P_q^i (1 - P_q)^{n-i}, \quad (2.11)$$

$$P_b \approx \frac{q/2}{q-1} P_s. \quad (2.12)$$

При выводе данных оценок предполагалось, что кодовый блок декодируется неверно при наличии в нем более чем  $t$  ошибочных символов, а неправильно декодированное кодовое слово отличается от переданного не более чем в  $(i+t)$   $q$ -ичных символах. Также считалось, что в неправильно декодированном символе примерно половина кодовых битов будет ошибочной.

Для некоторых  $(n, k)$  кодов БЧХ с  $r=k/n \approx 1/2$  оценка вероятности символьной ошибки представлена на рис. 2.3. На рис. 2.4 показаны графики зависимости оценки вероятности ошибки на символ  $P_s$  от отношения сигнал-шум на бит  $E_b/N_0$  для кодовых скоростей  $r$ , близких к  $1/3$  и  $2/3$ . Как следует из сопоставления представленных графиков, энергетическая эффективность кодов БЧХ с уменьшением кодовой скорости понижается.

Поскольку коды БЧХ относятся к циклическим блоковым кодам, задаваемым образующим полиномом, кодирование информационной последовательности осуществляется в соответствии с (1.29). Декодирование кодов БЧХ может выполняться с

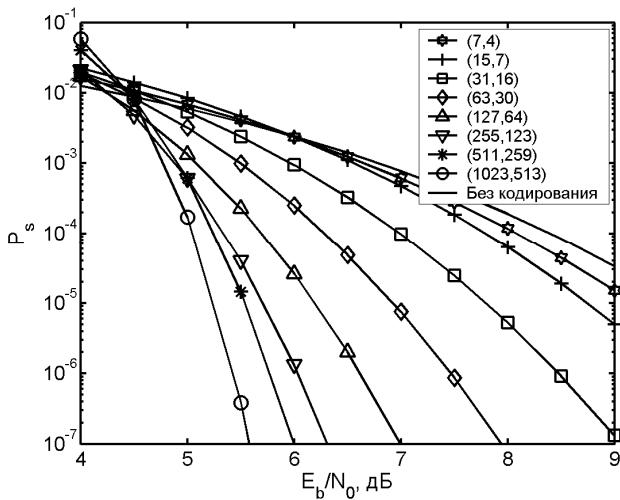


Рис. 2.3. Оценка характеристик различных кодов БЧХ в ДСК для кодовых скоростей, близких к 1/2

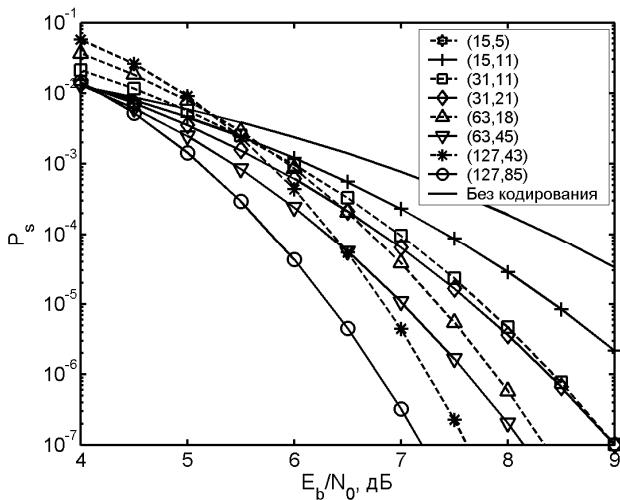


Рис. 2.4. Оценка характеристик различных кодов БЧХ в ДСК для кодовых скоростей, близких к 1/3 и 2/3

помощью алгебраических методов, достаточно подробно описаных в [15, 19].

На рис. 2.5 представлены экспериментальные зависимости вероятности ошибки на бит  $P_b$  от отношения сигнал-шум на бит  $E_b/N_0$  для кодов БЧХ с кодовой скоростью  $r \approx 1/2$  при работе в ДСК. Из сопоставления данного рисунка с рис. 2.3 (для двоичных кодов вероятность ошибки на бит  $P_b$  и на символ  $P_s$  совпадают) видно, что аналитическая зависимость вероятности ошибки (2.12) вполне подходит для оценивания сверху ее реального значения.

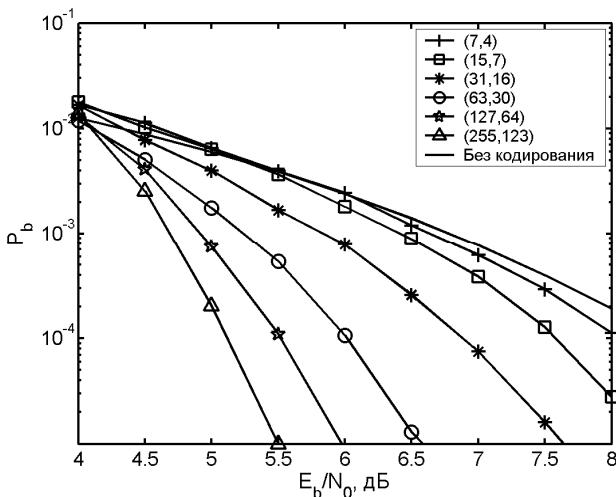


Рис. 2.5. Характеристики различных кодов БЧХ в ДСК для кодовых скоростей, близких к 1/2

Как видно из представленных характеристики, коды БЧХ обладают существенно лучшей эффективностью по сравнению с кодами Хэмминга, однако эти характеристики все еще очень далеки от предельных значений (см. рис. 1.8). Кроме того, сложность декодирования данных кодов при большом значении  $n$  очень велика. Поэтому коды БЧХ небольшой длины, так же

как и коды Хэмминга, в основном применяются в качестве составляющих элементов более эффективных каскадных кодов.

### 2.3. Коды Рида-Соломона

Важный подкласс кодов БЧХ составляют *коды Рида-Соломона* [21], для которых  $m=m_0=1$ . Эти недвоичные коды характеризуются следующими параметрами:

- длина блока  $n=q-1$ , выраженная количеством  $q$ -ичных символов;
- количество информационных символов  $k$  от 1 до  $n-1$ ;
- минимальное кодовое расстояние  $d_{min}=n-k+1$ ;
- кодовая скорость  $r=k/n$ .

Для задания кодов Рида-Соломона используется порождающий многочлен вида

$$g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{2t}),$$

$$\text{где } t = \left\lfloor \frac{1}{2}(d_{min} - 1) \right\rfloor.$$

Поскольку образующий полином имеет степень  $2t$ , возможно использование всего лишь  $2t$  проверочных символов для исправления пакетов из  $t$  ошибок. Последнее свойство позволило найти данным кодам достаточно широкое применение в каскадных методах кодирования, описанных в четвертой главе. Кроме того, для декодирования кодов Рида-Соломона существуют достаточно эффективные алгоритмы декодирования жестких решений, что позволяет использовать относительно длинные коды во многих практических приложениях, особенно в системах с  $q$ -ичной модуляцией.

Так как коды Рида-Соломона являются одной из разновидностей кодов БЧХ, для оценки их эффективности можно воспользоваться формулами (2.10) – (2.12). Для примера на рис. 2.6 представлены зависимости оценки вероятности символной ошибки  $P_s$  от вероятности ошибки  $P_q$  в  $q$ -ичном симметричном канале  $q$ СК для различных значений  $q$  и кодовых скоростей  $r$ . Кривые на рис. 2.7 отражают зависимость оценок  $P_s$  для тех же кодов от отношения сигнал-шум  $E_b/N_0$  на информационный бит.

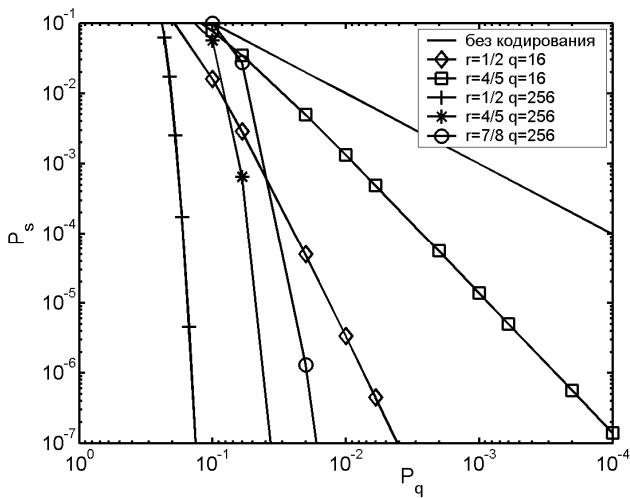


Рис. 2.6. Оценка характеристик кодов Рида-Соломона в  $q$ СК

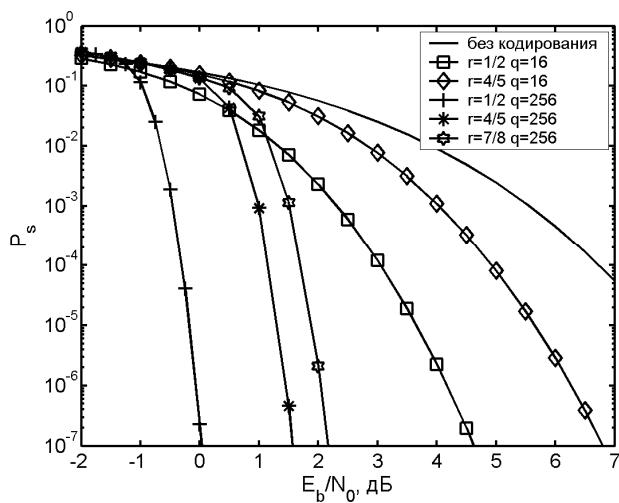


Рис. 2.7. Оценка характеристик кодов Рида-Соломона в  $q$ СК

## 2.4. Мажоритарно декодируемые коды

*Мажоритарно декодируемые коды* образуют класс линейных кодов (как сверточных, так и блоковых), имеющих ортогональные проверки и допускающих чрезвычайно простое мажоритарное декодирование, основанное на принципе «голосования по большинству».

Для разъяснения понятия ортогональности рассмотрим (15,7) код БЧХ, проверочная матрица которого имеет вид

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (2.13)$$

Используя 3, 7, 1+5 и 0+2+6 строки данной матрицы можно составить проверочные уравнения (далее иногда называемые просто проверками), зависящие только от вектора ошибки  $\bar{E}$ :

$$\begin{aligned} P_1 &= e_3 + e_{11} + e_{12} + e_{14} \\ P_2 &= e_7 + e_8 + e_{10} + e_{14} \\ P_3 &= e_1 + e_5 + e_{13} + e_{14} \\ P_4 &= e_0 + e_2 + e_6 + e_{14} \end{aligned} \quad . \quad (2.14)$$

Видно, что ошибка в  $e_{14}$  входит во все уравнения, в то время как ни одна другая ошибка не входит более, чем в одну проверку. Такое свойство проверочных уравнений называется *ортогональностью* относительно  $e_{14}$ .

Теперь кратко опишем принцип *мажоритарного декодирования*. Из анализа системы проверок (2.14) следует, что любая комбинация из одной или двух ошибок, не включающих  $e_{14}$ , исказит не более двух проверок, а содержащих  $e_{14}$  – как минимум три. Тогда процесс декодирования состоит в вычислении значения суммы указанных проверок и символа в позиции 14 в случае

превышения данной суммой пороговой величины, равной двум. Далее осуществляется циклический сдвиг кодового слова и начинается декодирование следующего символа.

Среди широкого класса кодов, допускающих мажоритарное декодирование, выделим *самоортогональные коды* (СОК) [17], допускающие очень эффективное многопороговое декодирование. СОК характеризуются тем, что система всех проверок, контролирующих символ  $e_0$ , уже сама является ортогональной относительно данного символа.

Обычно СОК задаются с помощью образующих полиномов  $g(x)$ , *разностные треугольники* (набор разностей между всеми степенями с ненулевыми коэффициентами) которых не содержат одинаковых элементов. Например, для полинома  $g(x)=1+x+x^4+x^6$  разностный треугольник имеет вид

$$\begin{array}{ll} 1-0, 4-0, 6-0 & 1, 4, 6 \\ 4-1, 6-1 & \Rightarrow 3, 5 \\ 6-4 & 2. \end{array}$$

Блоковый СОК характеризуются кодовым расстоянием  $d$ , равным увеличенному на 1 количеству ненулевых компонент образующего полинома. Количество информационных символов  $k$  в блоковом СОК с кодовой скоростью  $r=1/2$  равно как минимум  $2m+1$ , где  $m$  – максимальная степень образующего полинома. Длина  $n$  такого кода будет равна  $2k$ .

Пример кодера блокового СОК представлен на рис. 2.8. Данный код характеризуется параметрами  $n=26$ ,  $k=13$ ,  $r=1/2$ ,  $d=5$ ,  $g(x)=1+x+x^4+x^6$ .

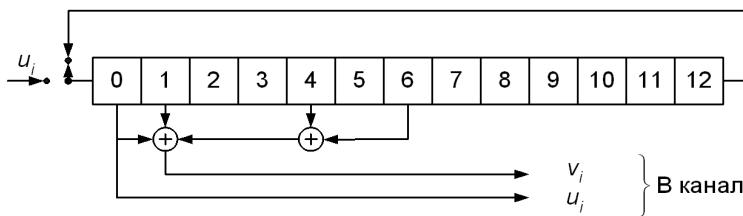


Рис. 2.8. Кодер блокового СОК

Зная порождающий полином  $g(x)$  систематического блокового СОК, можно определить его проверочную матрицу  $H=[P^T I_k]$ , причем строками матрицы  $P^T$  будут всевозможные циклические сдвиги коэффициентов образующего полинома. Например, для полинома  $g(x)=1+x+x^4+x^6$  проверочная матрица будет иметь вид

$$H = \begin{bmatrix} 1100101000000100000000000000 \\ 0110010100000010000000000000 \\ 0011001010000001000000000000 \\ 0001100101000000100000000000 \\ 0000110010100000010000000000 \\ 00000110010100000010000000 \\ 00000011001010000001000000 \\ 000000011001010000001000000 \\ 10000001100100000000100000 \\ 010000001100100000000010000 \\ 10100000011000000000000000000000 \\ 01010000001100000000000000000000 \\ 00101000000110000000000000000000 \\ 10010100000010000000000000000001 \end{bmatrix}. \quad (2.15)$$

Для оценки вероятности ошибки на бит  $P_b$  при оптимальном декодировании блокового СОК, можно воспользоваться следующей формулой

$$P_b = \begin{cases} \sum_{i=(d+1)/2}^d C_d^i p^i (1-p)^{d-i}, & -\text{ДСК}, d \text{ нечетно} \\ \frac{1}{2} C_d^{d/2} p^{d/2} (1-p)^{d/2} + \sum_{i=(d/2)+1}^d C_d^i p^i (1-p)^{d-i}, & -\text{ДСК}, d \text{ четно} \\ Q\left(\sqrt{2 \cdot d \cdot E_s / N_0}\right), & \text{канал с АБГШ} \end{cases} \quad (2.16)$$

где  $d$  –кодовое расстояние,  $p$  – вероятность ошибки на входе декодера,  $E_s/N_0$  – отношение сигнал-шум в канале,  $Q(x)$  – функция, определяемая выражением (1.3).

Графики зависимости оценки вероятности ошибки  $P_b$  от отношения сигнал-шум  $E_s/N_0$  при работе в ДСК и канале с АБГШ для блоковых СОК с  $d$  от 3 до 19 при их оптимальном декодировании представлены на рис. 2.9 и рис. 2.10 соответственно. Из сопоставления данных графиков видно, что при переходе от жесткого декодирования к мягкому можно увеличить выигрыш от кодирования примерно на 2 дБ.

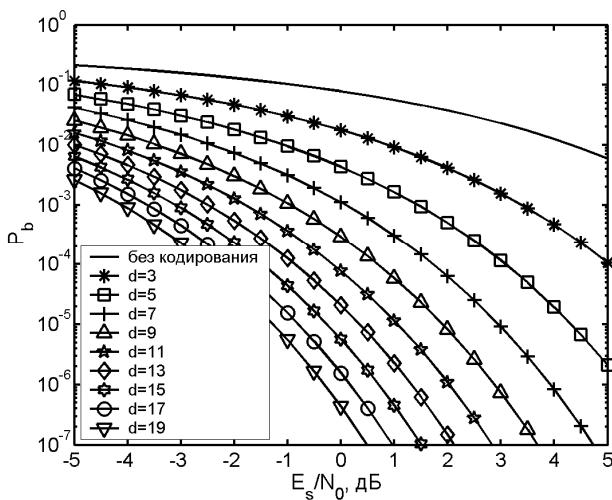


Рис. 2.9. Оценка характеристик оптимального декодирования СОК в ДСК для различных значений  $d$

## 2.5. Декодер Меггита

Декодер Меггита предложенный еще в 1961 г. для исправления пакетов ошибок является простым неалгебраическим методом декодирования систематических циклических кодов. Данный декодер является декодером максимального правдоподобия при декодировании жестких решений и его сложность с ростом числа исправляемых ошибок растет экспоненциально.

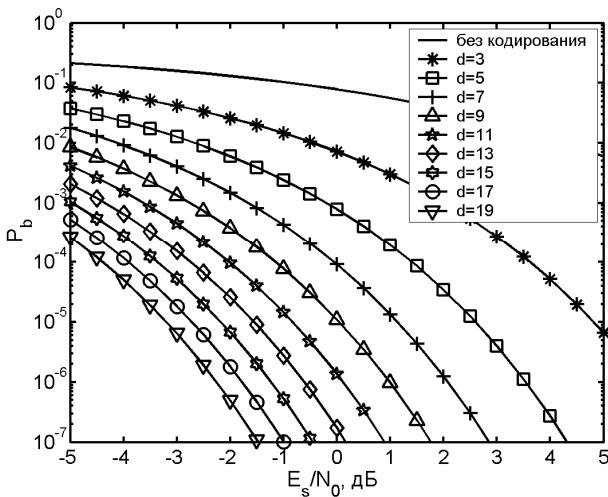


Рис. 2.10. Оценка характеристик оптимального декодирования СОК в канале с АБГШ для различных значений  $d$

Поэтому декодер Меггита обычно используется для коррекции небольшого (1÷3) числа ошибок.

Принцип работы данного декодера [15] основан на взаимно однозначном соответствии между множеством исправляемых ошибок и множеством синдромов, а также на наличии очень простой связи между синдромом, соответствующим некоторой комбинации ошибок и синдромом ее циклического сдвига: если синдром  $S(x)$  соответствует вектору ошибок  $E(x)$ , то синдром  $S'(x)=xS(x) \bmod g(x)$  соответствует комбинации ошибок  $x'$ , равной  $xE(x) \bmod (x^n-1)$ . Из последнего свойства следует, что все возможные синдромы можно разбить на непересекающиеся множества, каждое из которых соответствует циклическому сдвигу одной и той же комбинации ошибок. Таким образом, необходимо запомнить только синдромы, представляющие исправляемые комбинации ошибок, имеющие единицу в старшем разряде. В результате, процесс декодирования кодового блока можно разбить на  $n$  шагов, на каждом из которых текущий син-

дром сравнивается с запомненными, и, в случае совпадения, изменяется старший бит кодового блока. После этого содержимое синдрома сдвигается с помощью регистра с обратными связями и начинается следующий шаг декодирования.

Декодер Мэггита также существует в несколько усовершенствованном варианте, в котором используется обратная связь для удаления из регистра синдрома выявленных ошибок. В этом случае для более полного использования корректирующей способности кода можно осуществлять несколько циклов декодирования.

Схема декодера Мэггита с обратной связью для (15,7) кода БЧХ, заданного образующим полиномом  $g(x)=1+x^4+x^6+x^7+x^8$ , представлена на рис. 2.11 [15]. Поскольку в данной схеме с помощью цепей обратной связи устраняется влияние исправленных ошибок на синдром, после декодирования при исправлении всех возможных ошибок значение синдрома должно быть нулевым. Если же в синдроме будут содержаться ненулевые символы, то выданное пользователю кодовое слово будет содержать обнаруживаемые, но не исправляемые ошибки.

Как уже было сказано, сложность декодера Мэггита экспоненциально зависит от количества распознаваемых комбинаций ошибок. Поэтому он в основном применяется для декодирова-

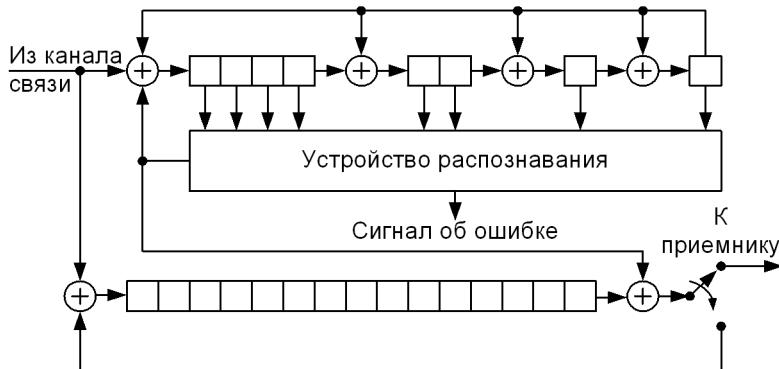


Рис. 2.11. Схема декодера Мэггита с обратной связью для (15, 7) кода БЧХ

ния кодов с небольшим кодовым расстоянием  $d$  (т.е. кодов Хэмминга, коротких кодов БЧХ и т.д.). Например, при декодировании кодов Хэмминга необходимо запомнить всего одно значение синдрома, поскольку с их помощью возможно исправление только одной ошибки.

## 2.6. Мягкое декодирование блоковых кодов

В параграфе 1.3 было показано, что использование при декодировании мягких решений относительно принятых из канала символов может улучшить эффективность декодирования примерно на 2 дБ. К сожалению, многие методы декодирования, такие как декодер Мэггита, алгебраические методы декодирования кодов БЧХ, очень трудно обобщить на случай использования мягких решений. Поэтому при переходе к работе в канале с АБГШ часто применяют методы декодирования, использующие при работе декодеры жестких решений. Одними из наиболее эффективных алгоритмов данного класса являются алгоритмы декодирования Чейза [15, 32], рассматриваемые в данном параграфе.

Принцип работы алгоритмов Чейза основан на жестком декодировании нескольких пробных последовательностей, полученных внесением в вектор жестких решений относительно мягких решений демодулятора различных комбинаций ошибок. Сказанное можно представить в виде следующего алгоритма.

1. Определяется вектор жестких  $\bar{D}$  решений для принятой последовательности  $\bar{Y}$ , т.е.

$$d_i = \begin{cases} 0, & \text{при } y_i < 0 \\ 1, & \text{при } y_i \geq 0 \end{cases}.$$

2. В вектор  $\bar{D}$  вводятся различные комбинации ошибок  $\bar{E}$ , порождая различные пробные последовательности  $\bar{F} = \bar{D} + \bar{E}$ .

3. При помощи декодера жестких решений декодируем каждую из пробных последовательностей  $\bar{F}$ , получая различные кодовые слова  $\bar{C}$ .

4. В качестве решения выбираем кодовое слово  $\bar{C}_0$ , находящееся на наименьшем евклидовом расстоянии от принятой последовательности мягких решений  $\bar{Y}$ , т.е.

$$\bar{C}_0 = \arg \min_{\{\bar{C}\}} \left( \sqrt{\sum_{i=1}^n (y_i - (2c_i - 1))^2} \right).$$

В работе [32] Чейз предложил три варианта данного алгоритма, отличающиеся методом выбора вносимых комбинаций ошибок. В первом методе в качестве векторов  $\bar{E}$  выбираются нулевой вектор и все комбинации ошибок веса  $\lfloor d/2 \rfloor$ . Второй метод предполагает выбор множества векторов, имеющих все возможные символы на  $\lfloor d/2 \rfloor$  наименее достоверных позициях (т.е. позициях с наименьшим значением  $|y_i|$ ) и нулевые на остальных. И, наконец, в третьем методе множество векторов  $\bar{E}$  имеет единицы на  $i$  наименее достоверных позициях и нулевые символы на остальных. Для последнего метода  $i=0, 2, 4, \dots, d-1$  при нечетном  $d$  и  $i=0, 1, 3, \dots, d-1$  при четном  $d$ .

Как следует из описания алгоритмов Чейза, первый метод приводит к наибольшему числу комбинаций ошибок и, соответственно, показывает наилучшие характеристики. Второй метод, практически не уступая по эффективности первому, характеризуется значительно меньшим числом вводимых комбинаций ошибок, что положительно сказывается на его быстродействии. Третий метод, используя очень небольшое число векторов ошибок, несколько уступает по эффективности двум первым, но оказывается существенно быстрее. Характеристики алгоритмов Чейза в канале с АБГШ с 8-ми уровневым квантованием при декодировании (24, 12) расширенного кода Голея представлены на рис. 2.12 [15]. При сравнении данных характеристик с характеристиками кодов БЧХ при декодировании жестких решений (рис. 2.5) видно, что алгебраические декодеры даже для более длинных (31, 15) кодов БЧХ уступают алгоритмам декодирования Чейза порядка 2 дБ.

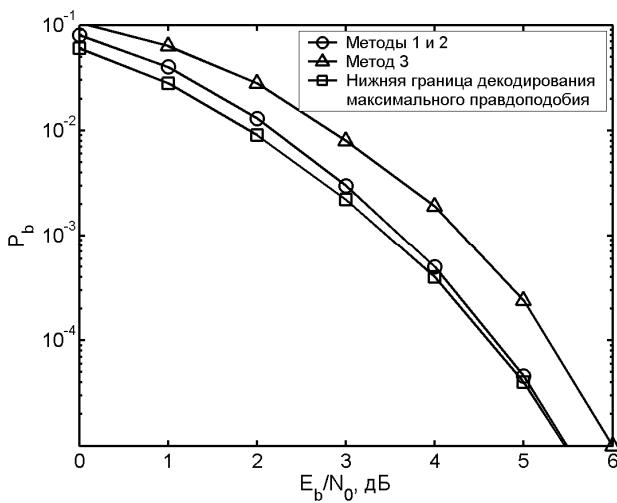


Рис. 2.12. Характеристики алгоритмов Чейза для (24, 12) расширенного кода Голея

## 2.7. Пороговое декодирование

Пороговое или мажоритарное декодирование [17] является очень простым алгоритмом декодирования, применимым к мажоритарно декодируемым кодам, представленным в параграфе 2.4. Данный метод может использоваться для декодирования как двоичных, так и недвоичных кодов.

Рассмотрим работу порогового декодера (ПД) в ДСК на примере двоичного блокового кода, кодер которого был изображен на рис. 2.8. На первом шаге работы ПД (см. рис. 2.13) с помощью входящего в его состав кодера, вычисляет проверочные элементы для принятых из канала информационных символов  $u$  (предположим, что данные символы уже находятся в информационном регистре) и складывает их по модулю 2 с принятыми проверочными  $v$  (при этом ключи  $K_2$  и  $K_3$  находятся в положении 1, а ключ  $K_1$  – в положении 2). В результате в синдромном регистре сформируется синдром. После заполнения

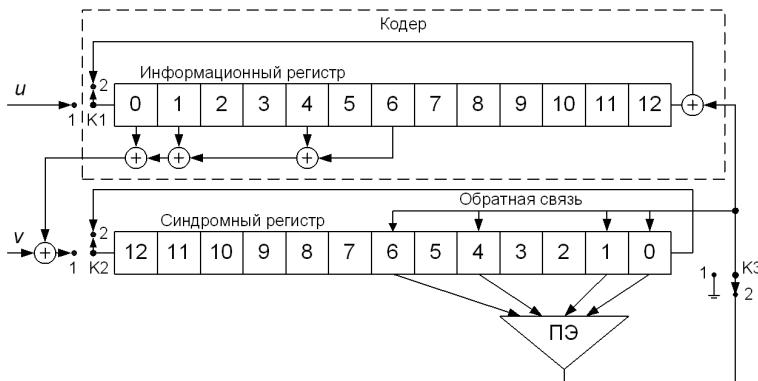


Рис. 2.13. Пороговый декодер блокового СОК

$$c n=26, r=1/2, d=5 \text{ и } g(x)=1+x+x^4+x^6$$

синдромного регистра ключи К2 и К3 переводятся в положение 2 и осуществляется декодирование информационного символа с номером 12, для чего в пороговом элементе (ПЭ) осуществляется сравнение суммы элементов синдромного регистра, соответствующих декодируемому символу (в данном случае элементов с номерами 0, 1, 4 и 6), с некоторым пороговым значением. В случае если указанная сумма окажется больше порога, выход ПЭ устанавливается равным 1, что приводит к изменению информационного символа и связанных с ним проверок. В противном случае на выходе ПЭ будет 0. Затем осуществляется циклический сдвиг содержимого регистров, после чего приступают к декодированию следующего информационного символа.

Рассмотренный ПД является *ПД с обратной связью*, поскольку исправленная в информационном регистре ошибка за счет обратной связи удаляется и из синдромного регистра. Использование обратной связи с одной стороны значительно улучшает эффективность работы ПД при небольшой вероятности ошибки в канале, а с другой – может привести к высокому уровню размножения ошибок (РО), заключающемуся в том, что при неправильном декодировании одного символа значительно

возрастает вероятность ошибочного декодирования нескольких последующих, так как в синдром при этом по цепи обратной связи дополнительно поступают  $d-1$  собственных ошибок ПД. В результате на выходе декодера может сформироваться пачка ошибок. Одним из способов предотвращения РО является использование *дефинитного ПД*, в котором обратная связь отсутствует. К сожалению, эффективность ПД данного типа несколько меньше и поэтому чаще используют другие методы ограничения РО, один из которых основывается на выборе кодов, характеризующихся небольшой долей общих проверок для различных информационных символов. Данный способ подробно рассмотрен в [7, 22].

Для дефинитного ПД блоковых СОК (т.е. ПД без обратной связи) при работе в ДСК существует аналитическая оценка вероятности ошибки в информационном бите [17], задаваемая выражением

$$P_b = (1-p) \sum_{i=T+1}^J C_J^i P^i (1-P)^{J-i} + p \sum_{i=J-T}^J C_J^i P^i (1-P)^{J-i}, \quad (2.17)$$

где  $p$  – вероятность искажения бита в канале связи,  $J$  – количество слагаемых образующего полинома, равное  $J=d-1$  ( $d$  – кодо-

вое расстояние),  $P = \frac{1 - (1-2p)^{J(n-1)}}{2}$  – вероятность наличия не-

четного числа ошибок в проверке относительно информационного символа (другими словами вероятность ошибочности самой проверки),  $T$  – величина порога на пороговом элементе, рав-

ная  $T = \left\lfloor \frac{d-1}{2} \right\rfloor$ . Данная оценка справедлива для СОК со скоро-

стями  $r=1/n$  и  $r=(n-1)/n$ .

Графики зависимости оценки (2.17) от отношения сигнал-шум в канале  $E_s/N_0$  для кодов с минимальным кодовым расстоянием  $d=3 \div 11$  представлены на рис. 2.14. Заметим, что выражение (2.17) подходит и для оценки вероятности ошибки дефинитного ПД сверточных СОК, описываемых в параграфе 3.3.

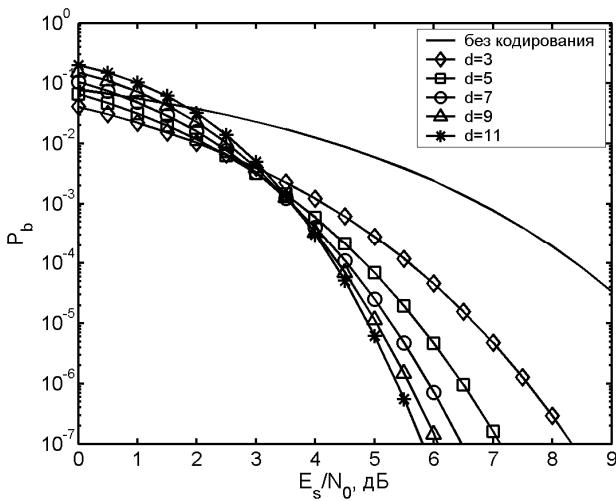


Рис. 2.14. Оценка характеристик дефинитного ПД для СОК с  $r=1/2$  в ДСК

Из анализа представленных графиков следует, что с ростом  $d$  дополнительный выигрыш от его дальнейшего повышения уменьшается. А поскольку сложность ПД, как будет показано далее, пропорциональна кодовому расстоянию, то достаточно быстро увеличение  $d$  оказывается невыгодным. Еще одно свойство ПД состоит в том, что при фиксированном значении кодового расстояния  $d$  лучшими характеристиками при переходе к отношению сигнал-шум на бит  $E_b/N_0$  будут обладать коды с большей кодовой скоростью  $r$ .

Результаты моделирования дефинитного ПД и ПД с обратной связью в случае работы в ДСК представлены на рис. 2.15. Номера кривых на данном рисунке соответствуют номерам СОК в табл. 2.1. В данной таблице образующие полиномы заданы перечислением степеней с ненулевыми коэффициентами.

Табл. 2.1. Список СОК для ПД

№	$n$	$k$	$r=k/n$	$d_{min}$	Коэффициенты образующего полинома $g_{ij}$
1	26	13	1/2	5	$g_{11}=0,1,4,6$
2	162	81	1/2	9	$g_{11}=0,1,3,7,15,24,35,40$
3	446	223	1/2	13	$g_{11}=0,1,3,7,12,25,35,51,65,82,103,111$
4	567	378	2/3	9	$g_{11}=0,1,4,10,21,40,58,90;$ $g_{21}=0,2,7,15,29,53,78,94$
5	99	33	1/3	9	$g_{11}=0,1,4,12;$ $g_{12}=0,2,7,16$

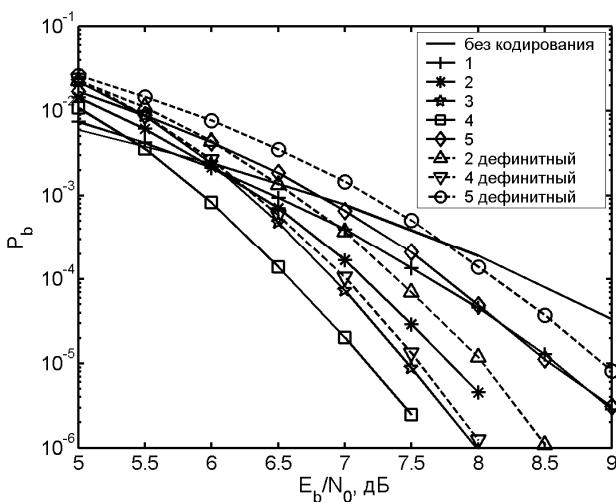


Рис. 2.15. Характеристики порогового декодера в ДСК для СОК из табл. 2.1

Из сопоставления данного рисунка с рис. 2.14 видно, что аналитическая оценка вероятности ошибки (2.17) дефинитного ПД достаточно точна. Также можно заметить, что дефинитные ПД при  $P_b=10^{-5}$  оказываются примерно на 0.25 дБ хуже, чем ПД с обратной связью.

В отличие от ранее рассмотренных методов декодирования, ПД можно легко обобщить для работы с мягкими решениями демодулятора. В этом случае пороговый элемент суммирует элементы синдрома с некоторым весом, зависящим от надежности символов, участвующих в формировании элементов синдрома. Результаты моделирования ПД для тех же СОК в случае работы в канале с АБГШ при квантовании выходного сигнала на 16 уровней представлены на рис. 2.16. Сравнивая данные графики с ранее представленными для ДСК видно, что переход к мягким решениям позволяет увеличить эффективность ПД блокового кода всего на  $0.2 \div 0.3$  дБ.

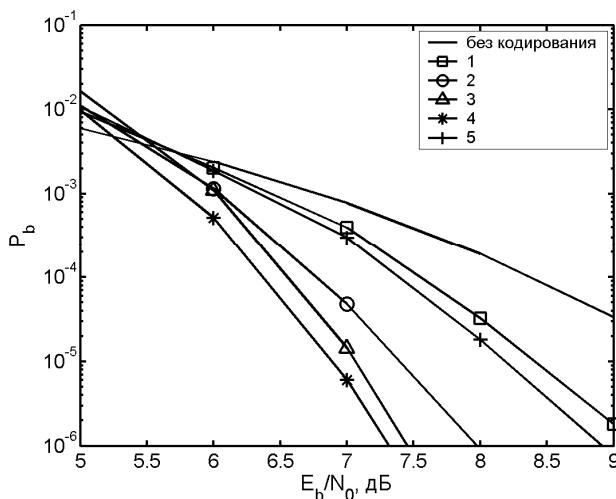


Рис. 2.16. Характеристики порогового декодера в канале с АБГШ при 16-уровневом квантовании для СОК из табл. 2.1

Еще немного увеличить выигрыш от кодирования можно использованием более длинных кодов при сохранении минимального кодового расстояния. Характеристики кодов с длиной до нескольких тысяч символов с  $d_{min}=9$  для кодовых скоростей  $r=1/2$ ,  $2/3$  и  $1/3$  при использовании как жестких, так и мягких решений демодулятора представлены на рис. 2.17.

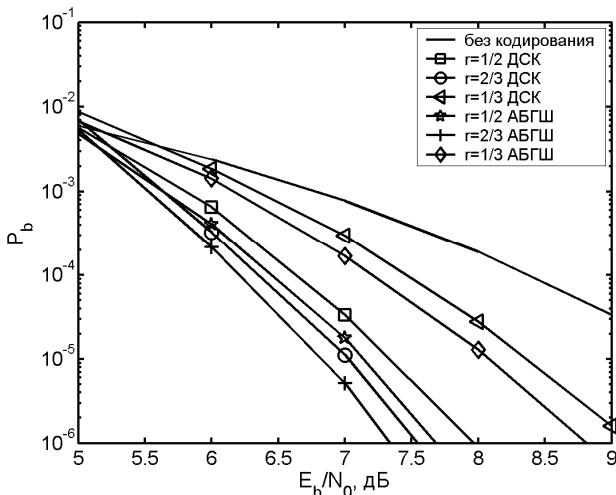


Рис. 2.17. Характеристики порогового декодера в ДСК и канале с АБГШ при 16-уровневом квантовании для длинных СОК

Анализируя сложность программной реализации ПД можно заметить, что при декодировании одного информационного бита требуется  $d-1$  операций сложения целых чисел для подсчета суммы на ПЭ, 1 операция сравнения с порогом и, в случае необходимости коррекции информационного символа и связанных с ним проверок,  $d$  операций сложения по модулю 2 (в канале с АБГШ вместо сложения по модулю 2 выполняется инверсия). Кроме того, при вычислении синдрома выполняются  $d-1$  операций сложения по модулю 2. Поскольку коррекция выполняется с вероятностью  $p \leq 0.1$  ( $p$  – вероятность ошибки в канале) и обычно ПД применяется для декодирования кодов с небольшим значе-

нием  $d$  среднее количество операций  $N_{\text{ПД}}$ , требуемых для декодирования одного информационного символа, будет примерно равно  $N_{\text{ПД}} \approx d+1+d-1=2d$ . Здесь предполагается, что операции сложения, сравнения, сложения по модулю 2 и инверсии имеют примерно одинаковую сложность. При аппаратной реализации ПД можно считать, что скорость его работы  $V_{\text{ПД}}$  в основном будет определяться скоростью продвижения данных  $V_p$  по регистрам сдвига, т.е.  $V_{\text{ПД}} \approx V_p$ .

## 2.8. Многопороговый декодер

Многопороговый декодер (МПД) самоортогональных кодов [12, 22, 46, 62] является развитием простейшего порогового декодера Месси и позволяет декодировать очень длинные коды с линейной от длины кода сложностью исполнения. В основе работы МПД лежит итеративное декодирование, что позволяет вплотную приблизиться к решению оптимального декодера в достаточно широком диапазоне кодовых скоростей и уровней шума в канале. При этом МПД сохраняет простоту и быстродействие обычного порогового декодера, что делает его очень привлекательным для применения в существующих и вновь создаваемых высокоскоростных системах связи.

Рассмотрим принцип работы МПД (см. рис. 2.18). Пусть задан двоичный линейный систематический блоковый или сверточный самоортогональный код, который используется для передачи сообщения из  $k$  двоичных символов. После кодирования общее число кодовых символов равно  $n$ ,  $n > k$ .

В результате передачи по двоичному симметричному каналу (обобщение на канал с АБГШ будет проведено далее) декодер получает вместо кодового слова  $\bar{C}_0$  искаженное шумами сообщение  $\bar{Y}$  длины  $n$ . Сначала, как и в обычном пороговом декодере, вычисляется синдром  $\bar{S} = \bar{Y}H^T$  принятого из сообщения, и для каждого информационного символа  $i_j$ ,  $1 \leq j \leq k$ , выделяется множество  $\{S_{j_k}\}$  элементов синдрома с номерами  $\{j_k\}$ , называемых проверками относительно символа  $i_j$  и содержащих в качестве слагаемого ошибку  $e_j$  в этом символе.

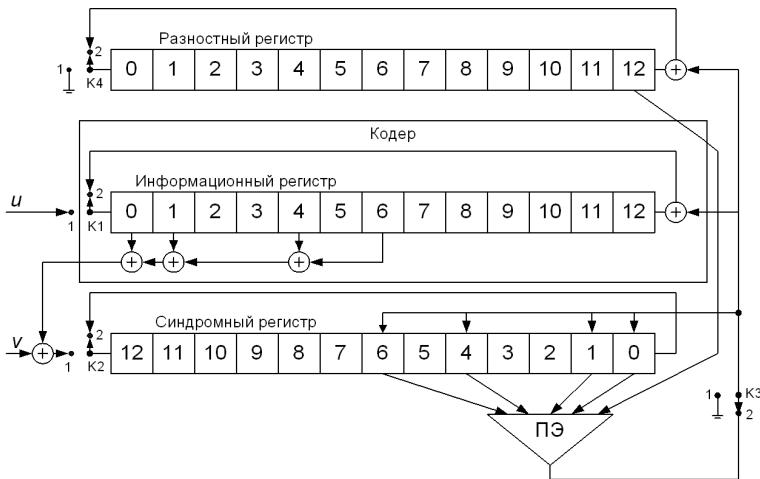


Рис. 2.18. Многопороговый декодер блокового СОК с  $r=1/2$ ,  $d=5$  и  $n=26$ . Верхние входы ячеек 0, 1, 4 и 6 регистра синдрома предполагаются инвертирующими

Дополнительно вводится двоичный вектор  $\bar{D}$  длины  $k$ , называемый *разностным*, первоначально равный нулю. В данном регистре будут отмечаться измененные информационные символы для того, чтобы декодер «помнил» принятное из канала сообщение и всегда мог вычислить разность между этим сообщением и кодовым словом, находящимся в информационном регистре.

Основной шаг декодирования заключается в том, что для произвольно взятого символа  $u_j$  вычисляется функция правдоподобия  $L_j$ , зависящая от относящихся к нему проверок  $S_{j_k}$  и  $j$ -го элемента вектора  $\bar{D}$ :

$$L_j = \sum_{\{j_k\}} S_{j_k} + d_j. \quad (2.18)$$

Общее число слагаемых в (2.18) равно минимальному кодовому расстоянию  $d$ . Если  $L_j > T$ , где  $T = (d-1)/2$  – пороговое значение

ние, то символ  $u_j$ , все проверки  $\{S_{j_k}\}$  и символ  $d_j$  инвертируются, после чего выбирается другой символ  $u_m$ ,  $m \neq j$ , для него снова вычисляется сумма  $L_m$  и т.д. Если же  $L_j \leq T$ , то сразу осуществляется переход к очередному символу  $u_m$ .

При выполнении основного шага декодирования все  $k$  информационных символов сообщения могут перебираться много-кратно и в любом порядке. При этом часть решений декодера на каких-то символах может быть ошибочной. Некоторые из этих ошибок будут исправлены при следующих попытках декодирования тех же символов (т.е. на следующих итерациях).

Основным свойством МПД является постоянная сходимость его решения к решению оптимального (по максимуму правдоподобия) декодера (ОД), поскольку при каждом  $i$ -м изменении декодируемых символов суммарный вес синдрома  $\bar{S}$  и разностного вектора  $\bar{D}$  уменьшается, т.е. происходит переход к кодовому слову  $\bar{C}_i$ , которое более правдоподобно, чем кодовое слово  $\bar{C}_{i-1}$ , находившееся в МПД в предыдущий момент времени. Переходя от одного слова  $\bar{C}_i$  к другому, МПД может получить наиболее правдоподобное слово  $\bar{C}_{OD}$ , которое, собственно, и является решением ОД. Однако нельзя утверждать, что МПД обязательно достигнет решения ОД, так как во многих кодах, допускающих мажоритарное декодирование, МПД на некоторых конфигурациях ошибок веса несколько большего, чем  $d/2$ , прекратит изменять информационные символы раньше, чем достигнет решения ОД.

Отметим, что МПД сохраняет свое основное свойство в некотором диапазоне значений порогов  $T$ , что позволяет осуществлять настройку декодера для минимизации вероятности ошибки. Например, на первых итерациях декодирования лучше увеличивать порог. При этом ПЭ этих итераций вносят меньшее количество собственных ошибок в сообщение, передаваемое следующим итерациям, что значительно повышает общую эффективность данной схемы.

Кроме того, МПД, как и обычный ПД, легко модифицируется для суммирования проверок в (2.18) с некоторыми коэффици-

ентами, в частности, при работе с квантованными на несколько уровней решениями мягкого модема (т.е. при работе в канале с АБГШ), дополнительные выходные биты которого определяют надежность выносимого им решения. Использование мягких решений демодулятора позволяет достигать результатов на 1.4–1.7 дБ лучших, чем при использовании только жестких демодулятора. При этом выражение (2.18) для вычисления функции правдоподобия  $L_j$  принимает вид

$$L_j = \sum_{\{j_k\}} S_{j_k} w_{j_k} + d_j w_j, \quad (2.19)$$

где  $w_i$  – коэффициент, отражающий надежность проверки  $S_{j_k}$ ,  $w_j$  – коэффициент, отражающий надежность принятого символа  $u_j$ .

Как уже было сказано, в процессе работы МПД может прекратить изменение информационных символов, еще не достигнув решения ОД. Одной из основных причин этого является значительная подверженность пороговых декодеров, являющихся составной частью МПД, эффекту размножения ошибок (РО). В результате вторая и последующие итерации декодирования вынуждены работать в основном с потоками пакетов ошибок от декодеров предыдущих итераций, что существенно уменьшает эффективность всего декодера. Следовательно, основным способом приближения решения МПД к решению ОД является уменьшение эффекта РО. Для уменьшения РО необходим тщательный выбор кодов, характеризующихся малой степенью пересечения множеств ошибок, входящих в проверки относительно разных информационных символов, а также настройка параметров декодера (например, величины порогов на разных итерациях). Методика оценки качества кодов в соответствии с указанным выше критерием подробно изложена в [7, 22].

Для примера на рис. 2.19 представлены характеристики МПД в ДСК для коротких кодов из табл. 2.1 с  $d=9$  и длинных кодов, в малой степени подверженных РО (те же коды, что и для рис. 2.17). При получении данных графиков использовалось от 5 до 15 итераций декодирования. Как следует из представленных графиков, применение МПД для декодирования плохо выбранных кодов позволяет добиться лишь небольшого улучшения ха-

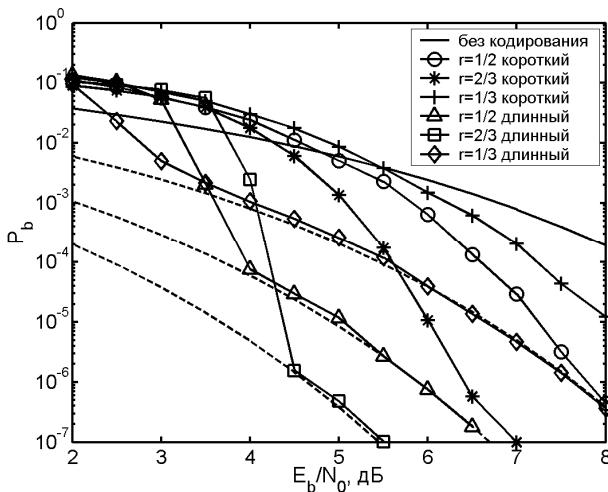


Рис. 2.19. Характеристики многопорогового декодера в ДСК для коротких и длинных СОК

рактеристик по сравнению с обычным ПД, в то время как при использовании кодов с малым РО обеспечивается почти оптимальное их декодирование (для сравнения на рис. 2.19 пунктирными линиями представлены графики зависимости вероятности ошибки оптимального декодера), что позволяет увеличить ЭВК более чем на 2 дБ при  $P_b=10^{-5}$ .

Характеристики МПД в канале с АБГШ при использовании квантования на 16 уровней для тех же кодов представлены на рис. 2.20. При использовании длинных кодов и в данном случае практически достигается эффективность декодера максимального правдоподобия. Заметим, что переход к мягким решениям демодулятора позволяет увеличить ЭВК на 1.5÷2 дБ по сравнению с применением только жестких решений.

Теперь рассмотрим вопросы сложности реализации МПД. Из сопоставления схемы МПД (рис. 2.18) со схемой ПД (рис. 2.13) видно, что они отличаются только наличием разностного регистра, т.е. при работе МПД в случае его программной

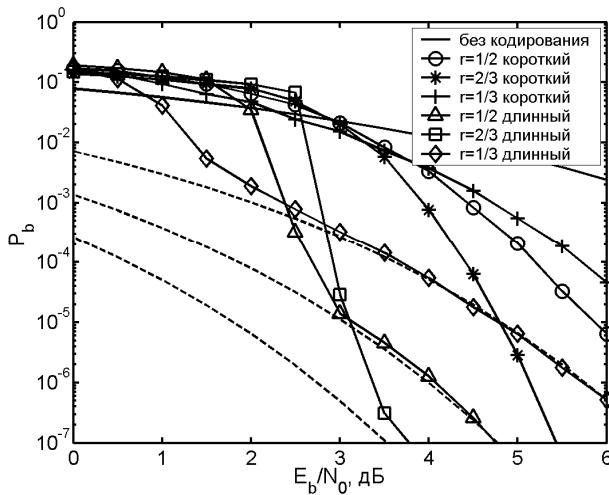


Рис. 2.20. Характеристики многопорогового декодера в канале с АБГШ для коротких и длинных СОК

реализации добавляется всего одна операция сложения при вычислении суммы на ПЭ и одна операция сложения по модулю 2 (или инверсии) при необходимости коррекции информационного символа. Таким образом, количество операций  $N_{\text{МПД}}$ , требуемых для декодирования одного информационного бита, примерно равно  $N_{\text{МПД}} \approx I(d+2) + d - 1 \approx (I+1)(d+2)$ , где  $I$  – количество итераций декодирования, а слагаемое  $d-1$  соответствует вычислению синдрома, выполняемому только на первой итерации декодирования. Заметим, что в большинстве случаев при незначительной потере в эффективности (порядка 0.1 dB) удается снизить общее число операций до величины  $N_{\text{МПД}} \approx 4d + 3I$ . В случае аппаратной реализации блокового МПД скорость его работы  $V_{\text{МПД}}$  также определяется скоростью продвижения данных в регистрах сдвига  $V_p$ , т.е.  $V_{\text{МПД}} \approx k_0 V_p / I$ , где  $k_0$  – количество информационных ветвей. Следует отметить, что МПД, как при программной, так и аппаратной реализации, оказывается на **два-три**

**порядка быстрее** [13, 51] сопоставимых по эффективности турбо кодов, рассматриваемых в четвертой главе.

Дополнительную информацию о многопороговом методе декодирования можно получить на сайте [46].

## 2.9. Недвоичные многопороговые декодеры

В данном параграфе рассмотрим обобщение многопорогового декодирования для  $q$ -ичных симметричных каналов, описанных в параграфе 1.2. По аналогии с соответствующими декодерами для двоичных данных эти алгоритмы названы  $q$ -ичными многопороговыми декодерами ( $q$ МПД). Они также обладают свойством сходимости к решению оптимального декодера (ОД) при сохранении линейной от длины кода сложности реализации, которая свойственна только пороговым процедурам. Ценность  $q$ МПД заключается в том, что в случае больших значений основания кода  $q$ ,  $q > 10$ , практически невозможно создать эффективные ОД, поскольку их сложность в большинстве случаев будет пропорциональна  $q^k$ , где  $k$  – длина информационной части кода, выраженная числом  $q$ -ичных символов.

Далее изложим основные принципы работы  $q$ МПД. Пусть задан  $q$ -ичный ( $q > 2$ ) симметричный канал ( $q$ СК) с вероятностью ошибки  $P_q > 0$  такой, что при передаче любой исходный символ кода переходит в один из оставшихся  $q-1$  символов случайно, независимо и равновероятно. Для этого канала решением ОД будет такое, может быть единственное, кодовое слово из  $q^k$  возможных, которое отличается от принятого из канала сообщения в минимальном числе кодовых символов.

Рассмотрим линейный недвоичный код, проверочная матрица которого имеет такой же вид, как и в двоичном случае, т. е. состоит только из нулей и единиц. Пусть эта матрица соответствует самоортогональному систематическому блоковому коду. Поскольку проверочные (а значит, и порождающие) матрицы кода содержат только нули и единицы, то для выполнения операций кодирования и декодирования принятого сообщения достаточно использовать операции сложения и вычитания по модулю  $q$ . Таким образом, для кодирования и декодирования не требуется наличие недвоичного поля, а достаточно создать только

группу чисел, что существенно упрощает все процедуры кодирования и декодирования. Пусть декодер типа  $q$ МПД (рис. 2.21) устроен так, что после вычисления обычным образом вектора синдрома  $\bar{S}$  принятого сообщения начинается процедура декодирования, заключающаяся в том, что для очередного контролируемого недвоичным пороговым элементом информационного символа кода  $u_j$  происходит подсчет количества и определение значений двух относящихся к нему и наиболее часто встречающихся проверок кода, например,  $b_1$  и  $b_2$ , причем  $b_1$  встречается  $m_1$  раз,  $b_2$  –  $m_2$  раз ( $m_1 > m_2$ ), а остальные значения проверок для декодируемого символа  $u_j$  встречаются не более  $m_2$  раз. Тогда на выходе  $q$ ПЭ будет значение  $b_1$  и  $q$ МПД при каждом изменении символа  $u_j$  будет переходить к все более правдоподобным решениям [5, 8], так как при этом число различий между кодовым словом, соответствующим текущему содержимому информационного регистра, и принятой из канала последовательностью будет уменьшаться. Если окажется, что два наиболее часто встречающихся значения проверок таковы, что  $m_1 = m_2$ , то выход

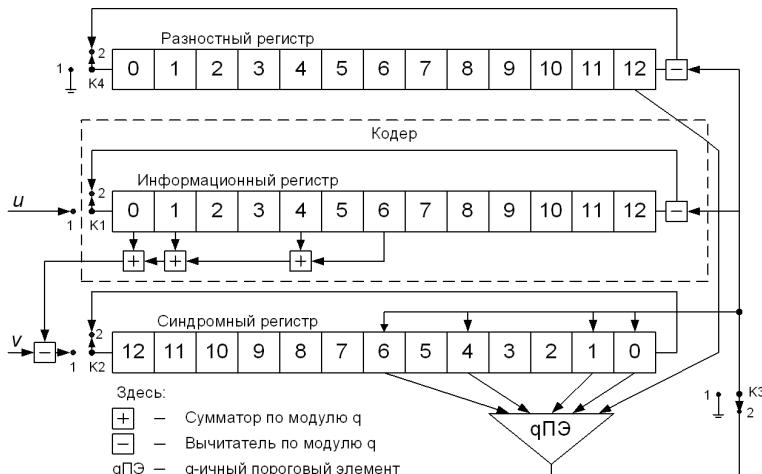


Рис. 2.21.  $q$ МПД блокового СОК с  $r=1/2$ ,  $d=5$  и  $n=26$ . Верхние входы ячеек 0, 1, 4 и 6 регистра синдрома вычтываются по модулю  $q$  из содержимого соответствующих ячеек

$q$ ПЭ устанавливается равным нулю, т.е. символ  $u_j$  не изменяется, и делается попытка декодирования любого другого информационного символа кода. Наиболее существенным обстоятельством, повышающим корректирующие возможности описанного  $q$ МПД, является возможность принимать безошибочные решения при больших значениях  $q$  всего при 2-х правильных проверках относительно  $u_j$  из  $d-1$  возможных, что происходит в том случае, когда все неправильные проверки относительно декодируемого символа  $u_j$  имеют различные значения.

Теперь рассмотрим процесс вычисления нижней оценки вероятности ошибки при оптимальном декодировании кода, задаваемого описанным выше способом. При этом необходимо выявить наиболее часто встречающиеся условия того, что вектор ошибки будет иметь расстояние Хемминга до ближайшего кодового слова меньшее, чем его собственный вес. В силу линейности кода этого достаточно для вынесения неправильного решения даже оптимальным переборным алгоритмом. Рассматривая такой вектор ошибки, будем учитывать, что нужно анализировать только те символы этого вектора, которые соответствуют позициям проверок относительно очередного декодируемого символа  $u_j$ . Выпишем вероятности таких наиболее частых событий, которые приводят к ошибкам оптимального декодера. К искомым векторам ошибки относятся такие, что [5, 8]:

– все проверочные символы и декодируемый символ  $u_j$  ошибочны:

$$P_1 = P_q^{J+1},$$

где  $J$  – число проверок кода, определяющее минимальное кодовое расстояние СОК ( $d=J+1$ );

– все проверочные символы ошибочны, но два из них одинаковы, а  $u_j$  принят верно:

$$P_2 = \frac{(1-P_q)P_q^J J(J-1) \dots (J-2)}{2(q-1)} \prod_{i=1}^{q-1} \left(1 - \frac{i}{q-1}\right);$$

– есть один правильно принятый проверочный символ, а остальные ошибочны, как и  $u_j$ :

$$P_3 = J(1-P_q)P_q^J;$$

– есть один правильно принятый проверочный символ, а также  $u_j$ , но из всех остальных неправильно принятых символов есть 3 одинаковых значения ошибок:

$$P_4 = \frac{(1-P_q)^2 P_q^{J-1} J!}{6(q-1)^2 (J-4)!} \prod_{i=1}^{J-4} \left(1 - \frac{i}{q-1}\right);$$

– есть 2 правильных проверочных символа, а все остальные, включая  $u_j$ , неправильны, причем есть 2 ошибочно принятых проверочных символа с одинаковыми значениями:

$$P_5 = \frac{(1-P_q)^2 P_q^{J-1} J!}{4(q-1)(J-4)!} \prod_{i=1}^{J-4} \left(1 - \frac{i}{q-1}\right);$$

– есть 3 правильных проверочных символа, а все остальные, включая  $u_j$ , неправильны, причем есть 3 ошибочно принятых проверочных символа с одинаковыми значениями:

$$P_6 = \frac{(1-P_q)^3 P_q^{J-2} J!}{36(q-1)^2 (J-6)!} \prod_{i=1}^{J-6} \left(1 - \frac{i}{q-1}\right).$$

Заметим, что если кодовое расстояние  $d < 7$ , то последний случай рассматривать не следует, так как он предполагает наличие  $J=6$  проверок в коде, тогда как для СОК  $d=J+1$ . Таким образом, нижняя оценка вероятности символьной ошибки оптимального декодирования определяется суммой найденных выше вероятностей  $P_s = \sum P_i, i=1 \div 6$ .

Перечисленных событий вполне достаточно, чтобы для большинства реальных условий применения кодов получать удовлетворительные по точности вероятностные оценки потенциальной помехоустойчивости кода. А поскольку  $q$ МПД на каждом шаге стремится к решению ОД, то можно ожидать, что при некотором достаточно высоком уровне шума он в большинстве случаев достигнет искомого оптимального решения.

Теперь сравним вероятностные характеристики кодов Рида-Соломона (РС) с возможностями  $q$ МПД. Подчеркнем, что для  $q$ МПД никаких ограничений по длине кода нет, поскольку он выполняет только операции сложения и сравнения в группе целых чисел. Очевидно, что недвоичный пороговый элемент, рассмотренный выше при описании операций в  $q$ МПД, – простей-

шее устройство или подпрограмма с числом операций  $N$  сложения и сравнения небольших целых чисел  $N \sim 10 \div 50$  для всех тех значений минимального кодового расстояния  $d$  ( $d < 15$ ), которое следует применять в таком декодере.

На рис. 2.22 представлены характеристики  $q$ МПД для блоковых СОК с кодовыми скоростями  $1/2$  (код с  $d=9$ ) в  $q$ СК при  $q=16$  и  $256$ . По горизонтальной оси отложены вероятности искажения символа  $P_q$  в указанном канале, а по оси ординат – средние вероятности ошибки на символ  $P_s$  после декодирования. На данном рисунке графики с пометкой «длинный» соответствуют кодам в наименьшей степени подверженных размножению ошибок, а без нее – кодам с длиной блока  $n$  порядка  $400$ . Для достижения решения, обычно совпадающего с оптимальным или близкого нему (кривая ОД для кода с  $R=1/2$ ,  $d=9$  при  $q=256$ , полученная ранее описанным способом),  $q$ МПД требовалось от  $5$  до  $20$  итераций декодирования принятого сообщения. Для более удобного сравнения на рис. 2.22 также показаны оценки характеристик кодов Рида-Соломона с теми же параметрами. Как

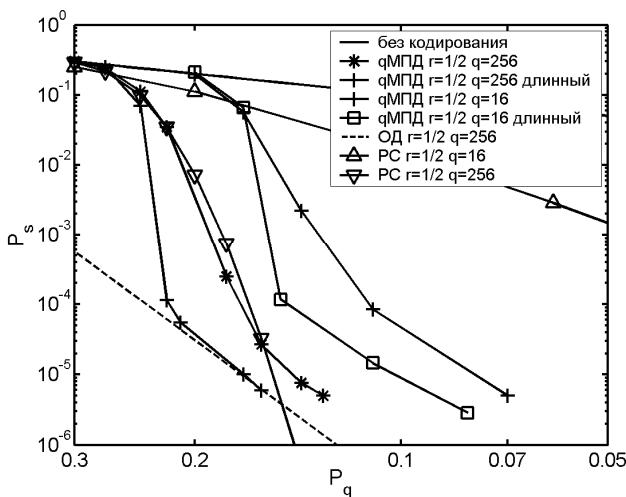


Рис. 2.22. Характеристики МПД в  $q$ СК для коротких и длинных СОК с кодовыми скоростями  $1/2$

следует из вида графиков зависимостей средней вероятности ошибки декодирования на символ  $P_s$  от вероятности ошибки  $P_q$  канала  $q$ СК на входе декодеров, простейший по своему устройству  $q$ МПД обеспечивает гораздо более высокие характеристики, чем декодеры для кода РС (особенно при небольших значениях  $q$ ), благодаря несколько большей длине  $n$  используемых кодов и хорошей сходимости решений  $q$ МПД к решению ОД. Заметим, что в настоящее время неизвестны другие алгоритмы декодирования с приемлемой сложностью реализации, которые могут обеспечить такие же характеристики.

На рис. 2.23 представлены характеристики  $q$ МПД для блоковых СОК с кодовыми скоростями  $4/5$  и  $7/8$  в  $q$ СК при  $q=256$ . Видно, что и в данном случае  $q$ МПД показывает сопоставимые с РС или лучшие характеристики.

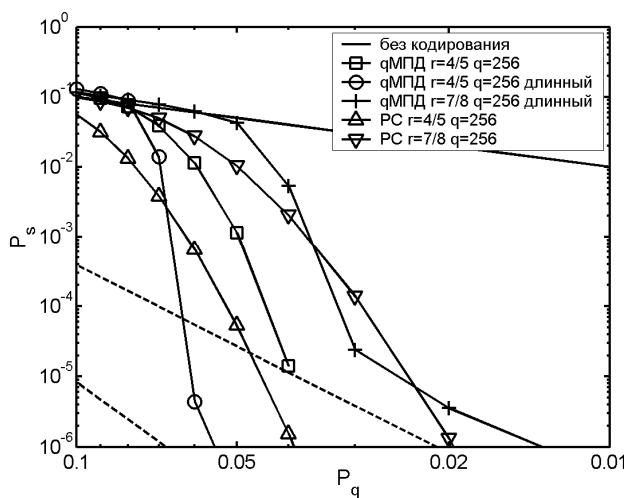


Рис. 2.23. Характеристики МПД в  $q$ СК для коротких и длинных СОК с кодовыми скоростями  $4/5$  и  $7/8$

### 3. СВЕРТОЧНЫЕ КОДЫ И МЕТОДЫ ИХ ДЕКОДИРОВАНИЯ

#### 3.1. Алгоритм декодирования Витерби

Алгоритм декодирования Витерби [4] предназначен для декодирования сверточных кодов, описанных в параграфе 1.4, и является оптимальным в смысле минимизации вероятности ошибки последовательности. Основная идея алгоритма Витерби состоит в пошаговом сравнении всех путей по кодовой решетке с принятой из канала последовательностью  $\bar{Y}$  и отбрасывании тех из них, которые точно будут находиться на большем расстоянии, чем другие пути. В случае работы в ДСК под расстоянием между двумя последовательностями понимается расстояние Хэмминга. В дальнейшем будем рассматривать алгоритм Витерби применительно к сверточным кодам со скоростью  $r=1/n_0$ , так как сложность декодера с ростом  $k_0$  значительно увеличивается. При необходимости использовать коды с кодовой скоростью вида  $r=k_0/n_0$  ( $k_0 > 1$ ), можно применять выкалывание, т.е. удаление части кодовых символов перед передачей сообщения в канал и вставкой на их место нулей на приемной стороне.

Опишем работу алгоритма Витерби во время приема из канала  $i$ -й  $n_0$ -символьной группы последовательности  $\bar{Y}$ . К данному моменту времени рассматриваемые пути могут проходить через  $2^{K-1}$  узлов (состояний) решетчатой диаграммы (здесь  $K$  – конструктивная длина кода), и для каждого из них вычислено расстояние от принятой последовательности (данное расстояние далее будем называть метрикой). На  $i$ -м шаге необходимо:

1. Вычислить расстояние Хэмминга между принятой  $n_0$ -символьной группой и всевозможными ветвями решетчатой диаграммы. Поскольку из каждого из  $2^{K-1}$  узлов выходит по 2 ветви, то необходимо вычислить  $2^K$  таких расстояний.

2. Расстояния Хэмминга для каждой из ветвей добавляются к метрикам путей, из которых они выходят. В результате получаются  $2^K$  возможных путей, ведущих в  $2^{K-1}$  состояний.

3. Для каждого из  $2^{K-1}$  состояний сравниваются метрики двух входящих в него путей и путь с меньшей метрикой, т.е. находящийся на меньшем расстоянии от входной последовательности.

тельности, становится выжившим. Путь с большей метрикой отбрасывается и не участвует в дальнейших вычислениях.

4. Запомнить все  $2^{K-1}$  выживших путей вместе с их метриками и перейти к выполнению  $(i+1)$ -го шага.

Теперь рассмотрим работу декодера Витерби кода со скоростью  $r=1/2$ , заданного образующими полиномами  $g_0(x)=1+x^2$  и  $g_1(x)=1+x+x^2$ , на простом примере. Пусть в канал передавалась нулевая кодовая последовательность  $\bar{C}=(0000\dots)$ , а из канала принято сообщение  $\bar{Y}=(1001000000\dots)$ . Поскольку свободное расстояние кода  $d_{free}$  равно 5, две полученные ошибки должны исправиться. Проследим за работой алгоритма Витерби при декодировании данной последовательности.

Первые два шага работы алгоритма тривиальны (см. рис. 3.1). На первом шаге из состояния 00 выходят два пути: в состояние 00 с расстоянием Хэмминга до принятой из канала  $n_0$ -символьной последовательности (далее именуемого метрикой), равным 1 и в состояние 10 с метрикой 1. На втором шаге формируются уже четыре пути, причем метрики состояний 00, 10, 01 и 11 становятся равными 2, 2, 1 и 3 соответственно.

Третий шаг работы является ключевым для понимания всего алгоритма. Рассмотрим состояние 00. В данное состояние можно прийти по двум различным путям: из состояния 00 и из состояния 01. Первый из них будет иметь метрику 2, поскольку метрика ветви  $(00 \rightarrow 00)$  равна нулю и предыдущее значение метрики пути равно 2, а второй – 3, так как метрика ветви  $(01 \rightarrow 00)$  равна 2 и предыдущая метрика пути равна 1. Сравнение метрик этих двух путей позволяет выбрать выживший путь (путь из состояния 00, так как он имеет меньшую метрику). Таким же образом выбираются выжившие пути для остальных узлов решетки.

Шаги 4,5,6 полностью аналогичны третьему шагу. Особенностью шага 7 является равенство метрик конкурирующих путей, входящих в одно состояние (состояние 01). Простейшим способом разрешения данной неопределенности является случайный выбор выжившего пути. После выполнения 11 шага

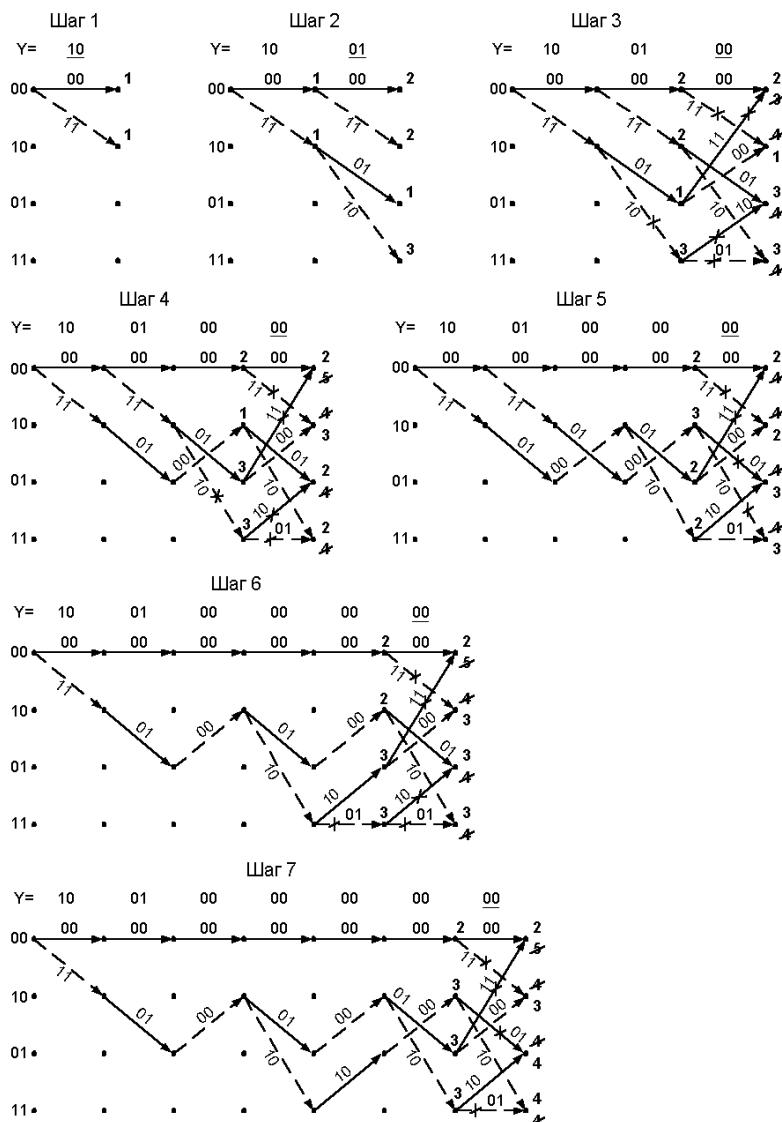
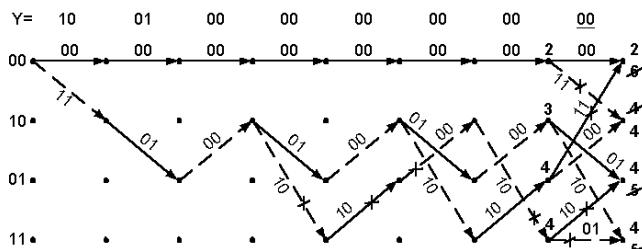
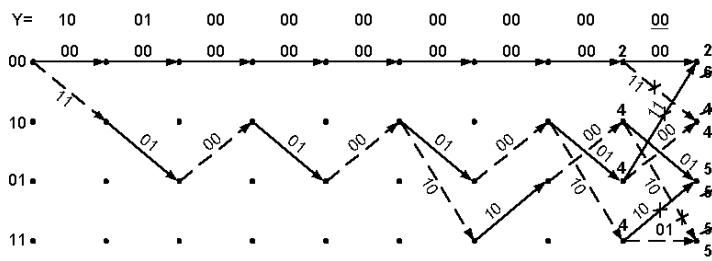


Рис. 3.1. Иллюстрация исправления ошибок с помощью алгоритма Витерби

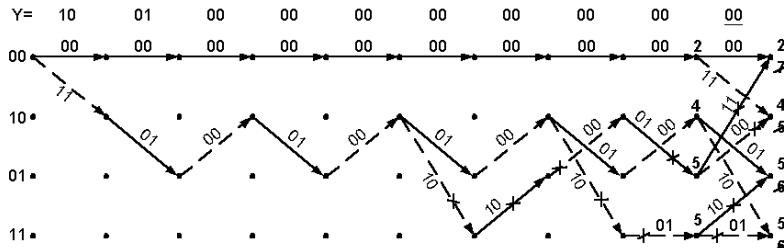
## Шаг 8



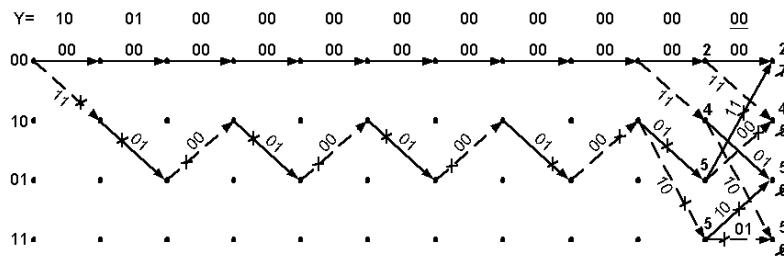
## Шаг 9



## Шаг 10



## Шаг 11



### Продолжение рис. 3.1

работы сложилась характерная ситуация, когда начальные части (первые 9 ветвей) всех выживших путей слились и теперь можно принимать решение о первых 9 информационных битах. Начальная часть выживших путей определяет информационную последовательность 000000000, т.е. две канальные ошибки были исправлены.

В рассмотренном примере решение о значении информационных символов было принято после 11 шага декодирования. Иногда начальные части выживших путей могут отличаться друг от друга в течение достаточно долгого времени, поэтому при практической реализации декодера обычно используют фиксированное число шагов с момента приема канальной последовательности, соответствующей информационному символу, до момента вынесения решения о его значении. Данную величину называют *глубиной декодирования*. Экспериментально было установлено, что при конструктивной длине кода  $K \leq 9$  глубина декодирования  $L$  должна быть большей, чем пять конструктивных длин кода  $K$ . При такой глубине декодирования ухудшение характеристик алгоритма Витерби по сравнению с бесконечной глубиной декодирования пренебрежимо мало. С ростом  $K$  глубину декодирования  $L$  выбирать несколько большей (например, при  $K=12$  значительное улучшение характеристик прекращается при  $L > 9K$ ).

Замечательной особенностью алгоритма Витерби является очень простой переход к декодированию мягких решений, для чего при вычислении метрики ветвей вместо расстояния Хэмминга (пункт 1 алгоритма) нужно использовать Евклидово расстояние. Остальные шаги алгоритма при этом не меняются.

Основным недостатком алгоритма Витерби является экспоненциальный рост числа просматриваемых путей с ростом конструктивной длины кода  $K$ . Например, при программной реализации алгоритма Витерби, как показано в табл. 4.1, для декодирования одного информационного символа требуется выполнение порядка  $3 \cdot 2^{K-1}$  операций, эквивалентных сложению. Поэтому на практике часто приходится выбирать коды с небольшим значением  $K \leq 9$  [34, 40, 57].

Далее рассмотрим характеристики алгоритма Витерби. Для сверточных кодов можно получить аддитивную границу вероятности битовой ошибки, имеющей вид

$$P_b < \frac{1}{k_0} \sum_{j=d_{\text{free}}}^{\infty} w_j P_j, \quad (3.1)$$

где  $w_j$  – полный информационный вес всех путей веса  $j$ ,  $P_j$  – вероятность выбрать неправильный путь, отличающийся от правильного в  $j$  позициях. Для часто используемых кодов известны несколько первых ненулевых значений  $w_j$ , представленные в табл. 3.1 – табл. 3.3 [15, 41]. Заметим, что в данных таблицах порождающие многочлены заданы в восьмеричной системе счисления. Вероятность  $P_j$  в выражении (3.1) можно определить по формуле (1.33).

Табл. 3.1. Порождающие многочлены и спектры наилучших кодов с  $r=1/2$

K	Порождающие многочлены	d	w <sub>d</sub>	w <sub>d+1</sub>	w <sub>d+2</sub>	w <sub>d+3</sub>	w <sub>d+4</sub>
3	7, 5	5	1	4	12	32	80
4	17, 15	6	2	7	18	49	130
5	37, 23	7	4	12	20	72	225
6	75, 53	8	2	36	32	62	332
7	171, 133	10	36	0	211	0	1404
8	371, 247	10	2	22	60	148	340
9	753, 561	12	33	0	281	0	2179
10	1151, 1753	12	2	21	100	186	474
11	3345, 3613	14	56	0	656	0	3708
12	5261, 7173	15	66	98	220	788	2083
13	12767, 16461	16	60	188	288	952	2754

Табл. 3.2. Порождающие многочлены и спектры наилучших кодов с  $r=1/3$ 

K	Порождающие многочлены	d	W <sub>d</sub>	W <sub>d+1</sub>	W <sub>d+2</sub>	W <sub>d+3</sub>	W <sub>d+4</sub>
3	7, 7, 5	8	3	0	15	0	58
4	17, 15, 13	10	6	0	6	0	58
5	37, 33, 25	12	12	0	12	0	56
6	75, 53, 47	13	1	8	26	20	19
7	171, 165, 133	15	7	8	22	44	22
8	367, 331, 225	16	1	0	24	0	113

Табл. 3.3. Порождающие многочлены и спектры наилучших кодов с  $r=2/3$ 

K	Порождающие многочлены	d	W <sub>d</sub>	W <sub>d+1</sub>	W <sub>d+2</sub>	W <sub>d+3</sub>	W <sub>d+4</sub>
3	7, 5, 7	3	1	10	54	226	853
4	15, 13, 15	4	8	34	180	738	2989
5	31, 33, 31	5	25	112	357	1858	8406
6	73, 41, 73	6	75	0	1571	0	31474
7	163, 135, 163	6	1	81	402	1487	6793
8	337, 251, 337	8	395	0	6695	0	235288
9	661, 473, 661	8	97	0	2861	0	56633

Аддитивная граница вероятности ошибки для представленных в таблицах кодов в случае работы в ДСК и канале с АБГШ показана на рис. 3.2 – рис. 3.7. Из данных графиков следует, что с повышением конструктивной длины кода  $K$  на единицу выигрыш от кодирования увеличивается примерно на 0.3÷0.5 дБ, а при одинаковых значениях  $K$  лучшими характеристиками обладают коды с меньшей кодовой скоростью. Также отметим, что переход от декодирования жестких решений к мягким позволяет увеличить эффективность на 2÷2.2 дБ.

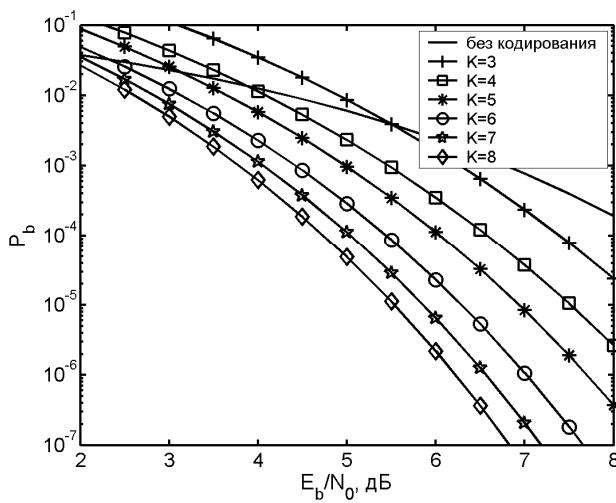


Рис. 3.2. Оценка характеристик декодера Витерби для кодов с кодовой скоростью  $r=1/3$  в ДСК

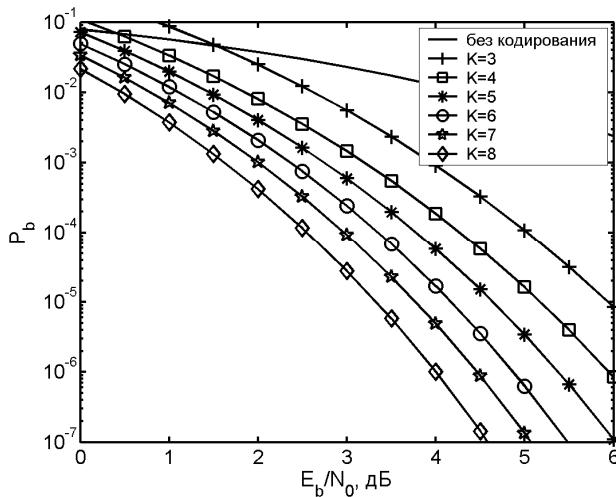


Рис. 3.3. Оценка характеристик декодера Витерби для кодов с кодовой скоростью  $r=1/3$  в канале с АБГШ

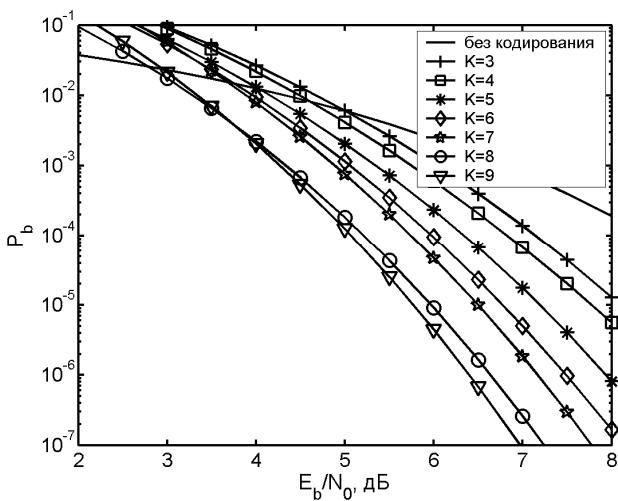


Рис. 3.4. Оценка характеристик декодера Витерби для кодов с кодовой скоростью  $r=1/2$  в ДСК

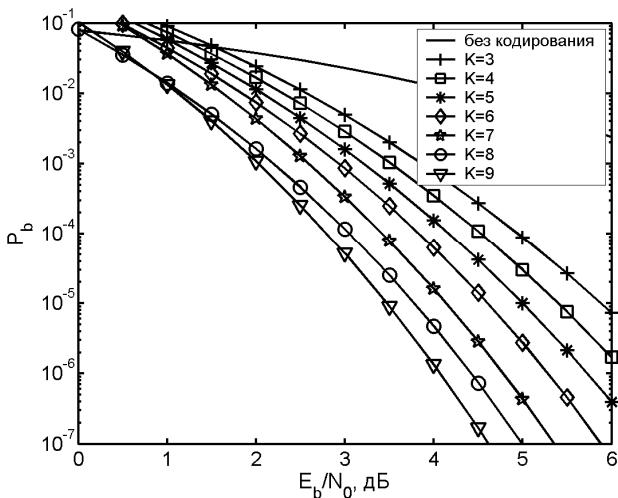


Рис. 3.5. Оценка характеристик декодера Витерби для кодов с кодовой скоростью  $r=1/2$  в канале с АБГШ

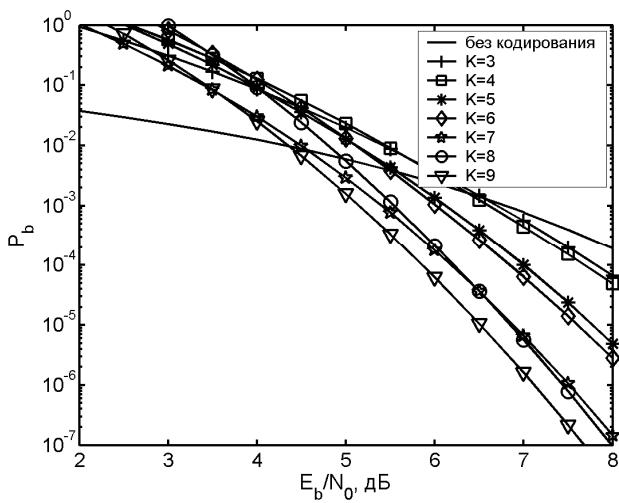


Рис. 3.6. Оценка характеристик декодера Витерби для кодов с кодовой скоростью  $r=2/3$  в ДСК

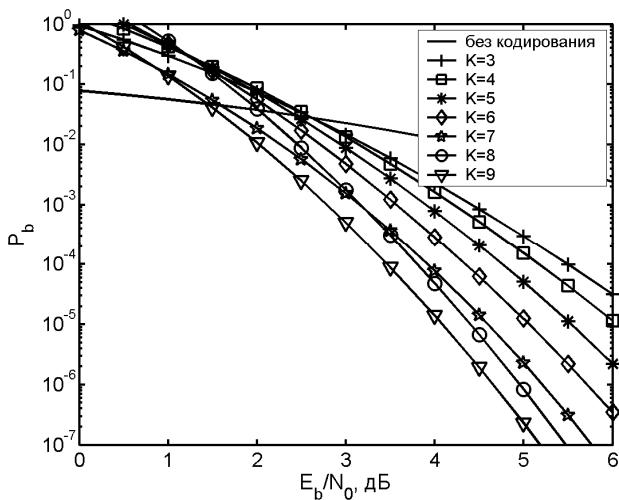


Рис. 3.7. Оценка характеристик декодера Витерби для кодов с кодовой скоростью  $r=2/3$  в канале с АБГШ

На рис. 3.8 отражены результаты моделирования алгоритма декодирования Витерби для кодов с кодовой скоростью 1/2 в ДСК. Их сравнение с ранее представленными оценками показывает достаточную точность последних при среднем и высоком отношении сигнал-шум. Характеристики алгоритма Витерби для тех же кодов в канале с АБГШ при квантовании выходного сигнала на 16 уровней показаны на рис. 3.9.

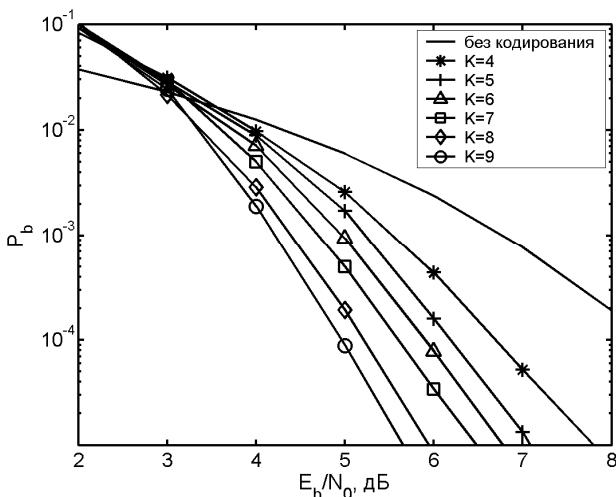


Рис. 3.8. Характеристики декодера Витерби для кодов с кодовой скоростью 1/2 в ДСК

### 3.2. Последовательные алгоритмы декодирования

Другим методом декодирования сверточных кодов является алгоритм последовательного декодирования, впервые предложенный Возенкрафтом, а затем усовершенствованный Фано. На сегодняшний день известно еще несколько модификаций последовательного декодирования [15]. В данном параграфе будет рассматриваться алгоритм, предложенный Фано.

Основным недостатком рассмотренного выше алгоритма Витерби является необходимость анализа всех путей, выходя-

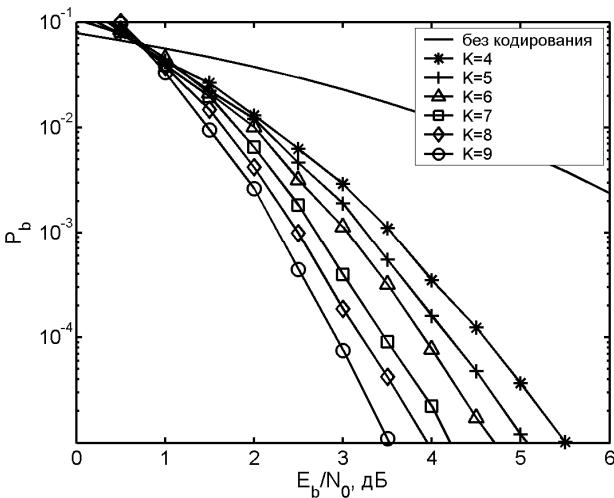


Рис. 3.9. Характеристики декодера Витерби для кодов с  $r=1/2$  в канале с АБГШ при 16-уровневом квантовании

щих из рассматриваемого узла кодовой решетки. При этом большинство выполняемых вычислений оказываются бесполезными и, следовательно, отказ от анализа маловероятных путей позволит кардинально уменьшить сложность декодирования при незначительных потерях в эффективности. Данный подход лежит в основе последовательных алгоритмов декодирования.

В процессе работы последовательный алгоритм декодирования, в отличие от алгоритма Витерби, осуществляет обработку лишь наиболее вероятного пути по кодовой решетке. Если декодер на каком-либо шаге декодирования совершил ошибку, расстояние между текущим путем и принятой последовательностью начнет быстро увеличиваться. Заметив это, декодер возвращается на один или несколько шагов назад и пытается найти более правильное решение. Для быстрого обнаружения неправильного пути и его быстрого исправления необходимо очень тщательно выбирать параметры декодера.

При поиске правильного пути на  $i$ -м шаге декодирования последовательный декодер стремится выбрать ветвь с максимальным значением функции правдоподобия  $L_i$  (также называемой далее метрикой пути), т.е. находящуюся на минимальном расстоянии от принятой  $n_0$ -символьной последовательности. При этом функция правдоподобия вычисляется следующим образом:

$$L_i = \sum_{j=0}^i \lambda_j, \quad (3.2)$$

где

$$\lambda_j = \sum_{k=1}^{n_0} \left\{ \log_2 \left[ \frac{P(y_j^k | c_j^k)}{P(y_j^k)} \right] - B \right\}, \quad (3.3)$$

где  $P(y_j^k | c_j^k)$  – вероятность приема из канала символа  $y_j^k$  при передаче символа  $c_j^k$ ,  $P(y_j^k)$  – вероятность приема символа  $y_j^k$ ,  $B$  – смещение. Величину смещения  $B$  обычно выбирают так, чтобы значение метрики при принятии правильного решения возрастало, а при принятии ошибочного решения – убывало. Достаточно хорошо указанным требованиям удовлетворяет выбор  $B=r$ , где  $r$  – кодовая скорость.

Несмотря на то, что метрика правильного пути на достаточно больших интервалах имеет тенденцию к возрастанию, из-за всплесков шума в канале метрика какого-либо неправильного пути может оказаться больше функции правдоподобия для правильного пути. При снижении шума метрика неправильного пути начинает уменьшаться, и задачей алгоритма декодирования становится выявление такой ситуации и поиск пути с возрастающей метрикой. Для этого метрика текущего пути  $L_i$  сравнивается с некоторым переменным порогом  $T$ , и, если  $L_i < T$ , то рассматриваемый путь является плохим и необходимо начать поиск лучшего пути. Отметим, что в процессе работы алгоритма значение порога может меняться.

Главным преимуществом последовательного алгоритма декодирования по сравнению с алгоритмом Витерби является меньшая зависимость сложности декодера от конструктивной

длины кода. В результате с помощью данного метода можно декодировать более длинные коды, обладающие большим свободным расстоянием, а, соответственно, и лучшей эффективностью. Однако при практической реализации алгоритма последовательного декодирования сталкиваются с рядом трудностей, вызванных тем, что требуемое для декодирования одного информационного бита количество операций является случайной величиной и быстро растет с ухудшением канала передачи данных. В результате возникает необходимость в наличии большого количества буферной памяти, предназначеннной для хранения поступившей, но еще не обработанной канальной последовательности. Причем, так как объем данной памяти конечен, всегда существует вероятность ее переполнения, что приведет к стиранию потерянных символов.

Важной характеристикой канала для последовательных алгоритмов декодирования является вычислительная скорость  $R_0$ , рассмотренная в параграфе 1.3. Данная величина определяет предельную скорость работы последовательного декодера, т.е. условием нормальной работы является  $r \leq R_0$ . Если же  $r > R_0$ , то в декодере будут возникать вычислительные проблемы, вызванные частым переполнением буфера. Анализируя графики зависимости  $R_0(E_b/N_0)$  можно определить наименьшее значение отношения сигнал-шум  $E_b/N_0$ , при котором возможна работа последовательного декодера. Например, из графиков на рис. 1.8 следует, что для ДСК и канала с АБГШ в случае использования кодов с  $r=1/2$  требуемые значения  $E_b/N_0$  равны соответственно 4.6 dB и 2.5 dB, что существенно ниже теоретически достижимых границ, определяемых пропускной способностью канала  $C$ .

Теперь рассмотрим характеристики данного метода. На рис. 3.10 представлены графики зависимости вероятности битовой ошибки  $P_b$  от отношения сигнала-шума на бит  $E_b/N_0$  для последовательного декодера кода с конструктивной длиной  $K=41$  в канале с АБГШ [20]. Для сравнения здесь также представлены оценки характеристик сверточного кода с  $K=7$ , декодируемого с помощью декодера Виттерби. Как следует из данного рисунка, последовательные декодеры за счет применения кодов с большим  $K$  оказываются значительно эффективнее декодера Виттерби.

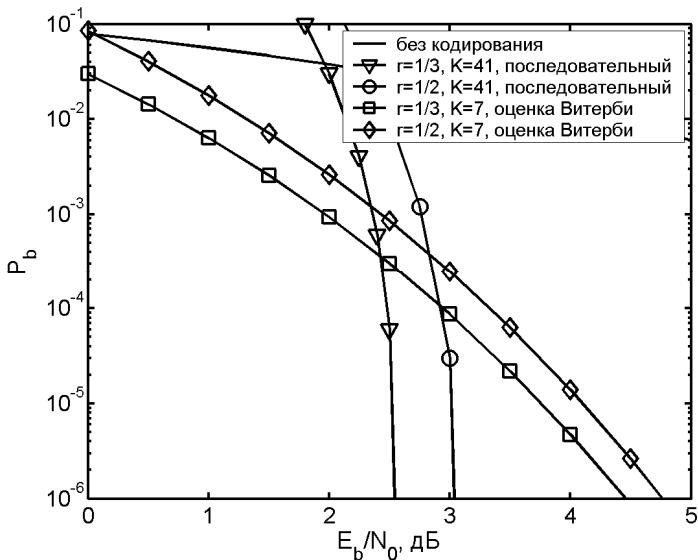


Рис. 3.10. Характеристики последовательного декодирования сверточных кодов в канале с АБГШ

би при малых вероятностях ошибки декодирования  $P_b$ . К сожалению, практическая реализация последовательных алгоритмов декодирования все равно остается громоздкой, поскольку декодер проектируется на выполнение наибольшего количества операций, которое может понадобиться для декодирования сильно зашумленной последовательности.

Более подробную информацию о последовательных алгоритмах декодирования можно найти в книге [15].

### 3.3. Пороговый декодер

Перед описанием порогового декодера (ПД) сверточных кодов кратко рассмотрим способы задания сверточных самоортогональных кодов. Эти коды, так же как и ранее описанные блоковые СОК, обычно определяются с помощью образующих полиномов, разностные треугольники которых не содержат одинак-

ковых элементов. Для примера на рис. 3.11 представлена схема кодера сверточного самоортогонального кода, заданного образующим полиномом  $g(x) = 1 + x + x^4 + x^6$ . Проверочная матрица данного кода определяется так:

$$H = \left( \begin{array}{ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{array} : I \right). \quad (3.4)$$

Данный код является самоортогональным, поскольку множество компонент  $s_i$  синдрома  $\bar{S}$ , содержащих в качестве слагаемых ошибку  $e_0$  в первом информационном символе

$$\begin{aligned} s_0 &= e_0 + e_{0v} \\ s_1 &= e_0 + e_1 + e_{1v} \\ s_4 &= e_0 + e_3 + e_4 + e_{4v} \\ s_6 &= e_0 + e_2 + e_5 + e_6e_{6v}, \end{aligned} \quad (3.5)$$

само является ортогональным относительно ошибки  $e_0$ . В выражении (3.5) ошибка  $e_i$  определяет ошибку в  $i$ -м информационном символе, а  $e_{iv}$  — в  $i$ -м проверочном.

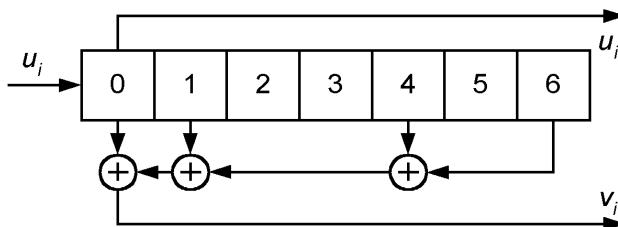


Рис. 3.11. Кодер сверточного самоортогонального кода

Так же как и к блоковым, к сверточным самоортогональным кодам применимо пороговое декодирование. Схема порогового декодера сверточного СОК, кодер которого представлен на рис. 3.11, показана на рис. 3.12. Как видно из рисунка, декодер состоит из кодера, синдромного регистра и порогового элемента. Процедура исправления ошибок данным декодером в целом совпадает с работой порогового декодера блокового СОК, за тем исключением, что при переходе к декодированию следующего информационного символа осуществляется обычный, а не циклический сдвиг составляющих регистров.

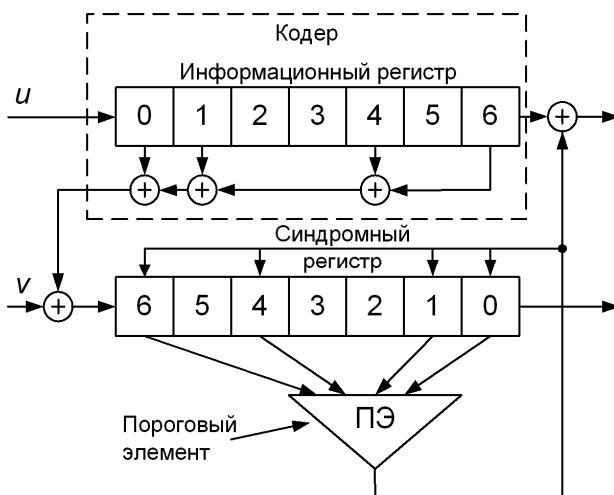


Рис. 3.12. Пороговый декодер сверточного СОК

$$c_{n_d=13}, r=1/2, d=5 \text{ и } g(x)=1+x+x^4+x^6$$

Для оценки вероятности ошибки дефинитного ПД сверточного СОК можно воспользоваться выражением (2.17). В случае ПД с обратной связью возможна оценка лишь вероятности первой ошибки, совпадающая с вероятностью ошибки оптимального декодера и определяемая по формуле (2.16).

Результаты моделирования ПД с обратной связью и дефинитного ПД для коротких СОК из табл. 2.1 представлены на рис. 3.13 и рис. 3.14. При сравнении характеристик ПД блоковых и сверточных кодов последние по эффективности оказываются лучше на 0.3÷0.5 dB, а соотношение между дефинитными ПД и ПД с обратной связью, между пороговыми декодерами для СОК с разной кодовой скоростью и свободным расстоянием остаются прежними. При переходе к более длинным кодам, характеристики которых представлены на рис. 3.15, не заметно значительного улучшения эффективности. В данном случае использовались те же коды, что и на рис. 2.17.

Сложность ПД сверточного СОК, как при программной, так и при аппаратной реализации, полностью совпадает со сложностью ПД блокового кода.

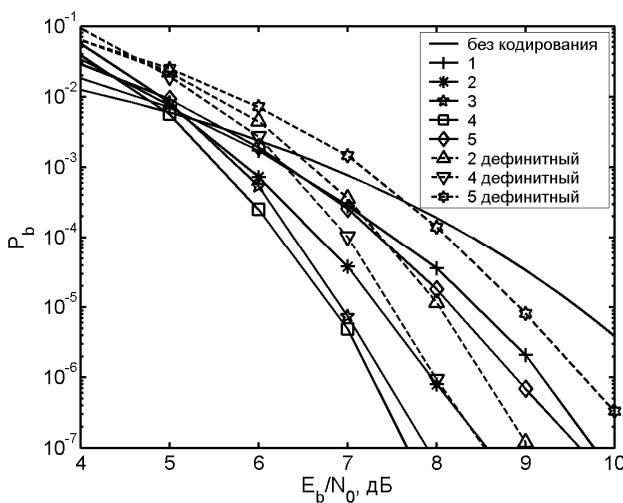


Рис. 3.13. Характеристики порогового декодера в ДСК для СОК из табл. 2.1

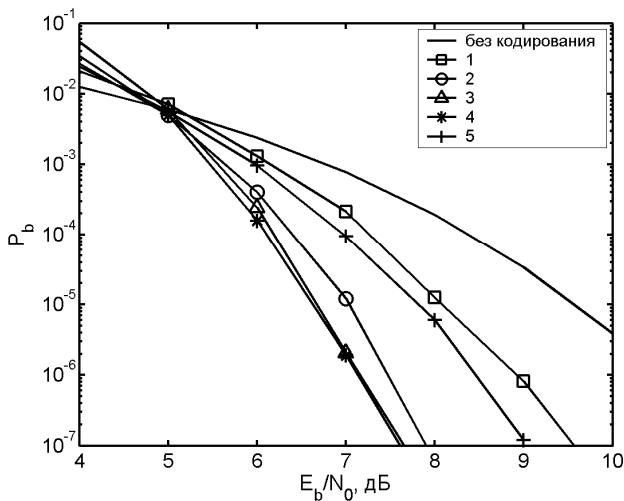


Рис. 3.14. Характеристики порогового декодера в канале с АБГШ при 16-уровневом квантования для СОК из табл. 2.1

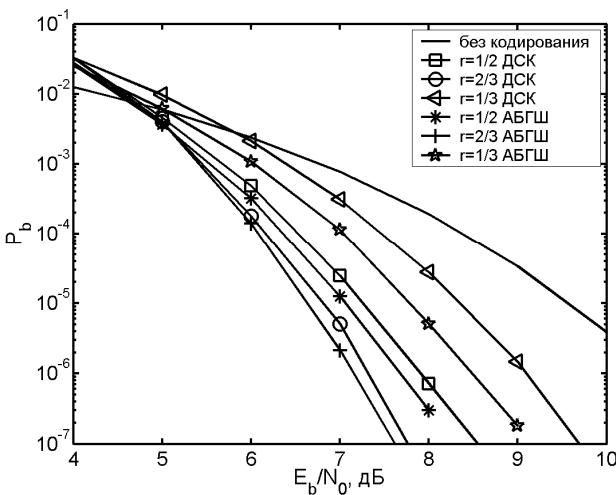


Рис. 3.15. Характеристики порогового декодера в ДСК и канале с АБГШ при 16-уровневом квантования для длинных СОК

### 3.4. Многопороговый декодер

Многопороговый метод декодирования, представленный в параграфе 2.8, может применяться для декодирования сверточных самоортогональных кодов. При этом декодер будет состоять из нескольких последовательно связанных блоков декодирования (рис. 3.16). Представленный на рис. 3.16 МПД сверточного СОК содержит всего две итерации декодирования, но он легко может быть преобразован в МПД с большим количеством итераций простым добавлением еще нескольких блоков декодирования, структурная схема которых полностью совпадает со вторым блоком декодирования.

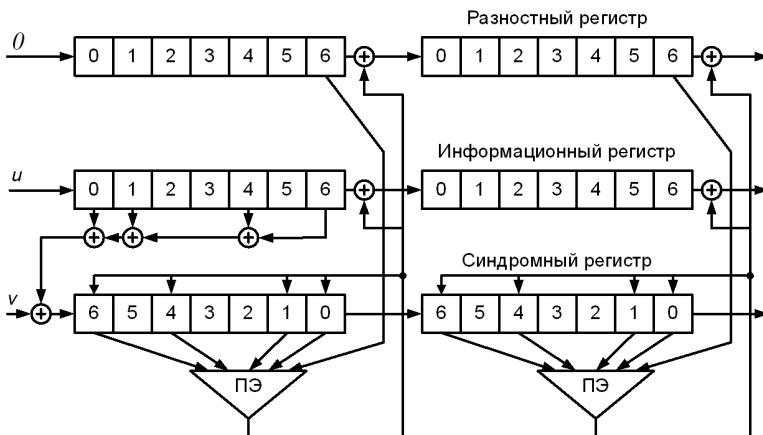


Рис. 3.16. Многопороговый декодер сверточного СОК с  $r=1/2$ ,  $d=5$  и  $n_A=13$  для двух итераций декодирования. Верхние входы ячеек 0, 1, 4 и 6 регистра синдрома предполагаются инвертирующими

Характеристики многопорогового декодера сверточных СОК при работе в канале типа ДСК и канале с АБГШ представлены на рис. 3.18 и рис. 3.17 соответственно. На данных рисунках отражены результаты моделирования тех же кодов, что и на рис. 2.19 и рис. 2.20. Пунктиром снова показаны кривые зави-

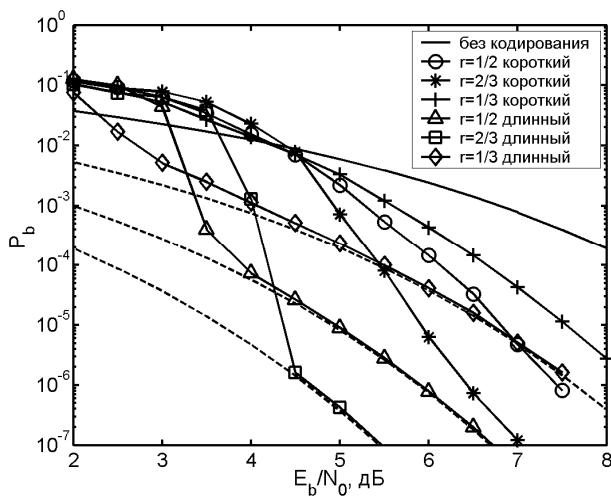


Рис. 3.18. Характеристики многопорогового декодера в ДСК для коротких и длинных СОК

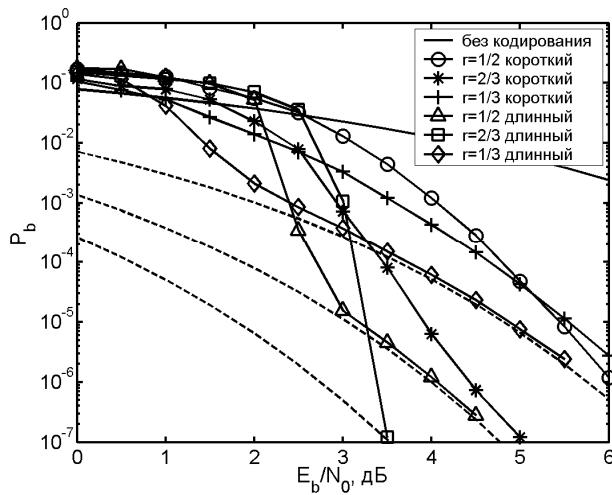


Рис. 3.17. Характеристики многопорогового декодера в канале с АБГШ для коротких и длинных СОК

симости вероятности ошибки при оптимальном декодировании используемых кодов. Из анализа данных графиков следует, что эффективность МПД для длинных блоковых и сверточных СОК примерно одинакова, а небольшое отличие обусловлено различной степенью оптимизации величин порогов на пороговых элементах разных итераций декодирования. Подчеркнем, что получаемые с помощью МПД результаты недостижимы для практически реализуемых декодеров Витерби, поскольку МПД даже при достаточно высоком уровне шума фактически оптимально декодирует длинные коды. Эффективность сверточных  $q$ МПД в  $q$ -ичном симметричном канале, как следует из сопоставления их характеристиками, представленных на рис. 3.19 и рис. 3.20, с характеристиками на рис. 2.22 и рис. 2.23, также примерно равна эффективности блоковых  $q$ МПД.

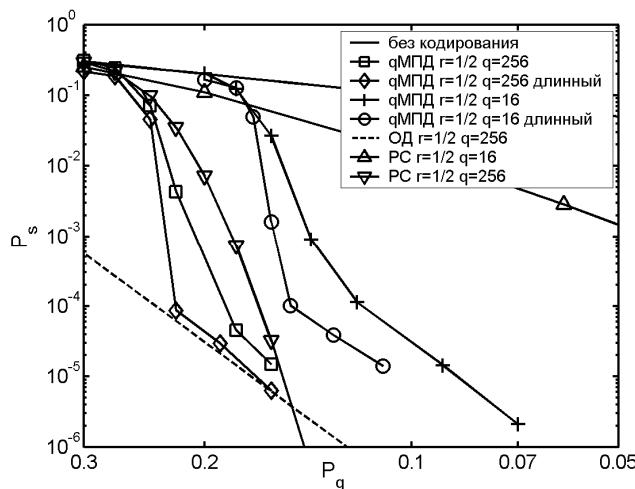


Рис. 3.19. Характеристики МПД в  $q$ СК для коротких и длинных СОК с кодовыми скоростями 1/2

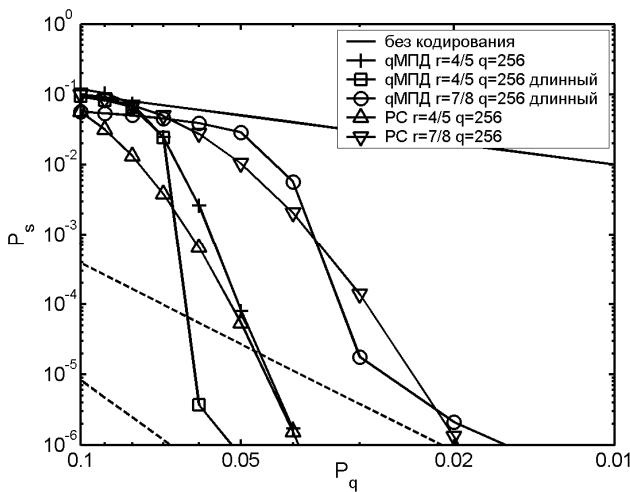


Рис. 3.20. Характеристики МПД в  $q$ СК для коротких и длинных СОК с кодовыми скоростями 4/5 и 7/8

Сложность программной реализации МПД сверточного СОК равна сложности МПД блокового кода, т.е.  $N_{МПД} \approx (I+1)(d+2)$  или  $N_{МПД} \approx 4d+3I$  при пренебрежимо малых потерях в энергетике ( $\sim 0.1$  дБ). В последнем случае МПД при одинаковой эффективности с достаточно мощными турбо кодами, описанными в параграфе 4.4, оказываются быстрее почти в 100 раз. При аппаратной реализации МПД сверточного кода скорость его работы, как и в блоковом варианте, будет определяться скоростью продвижения данных по регистрам сдвига, поскольку пороговые элементы для многих кодов, декодируемых с помощью МПД, можно реализовать в виде простой однотактной схемы. Причем в данном случае скорость уже не будет зависеть от количества итераций декодирования, т.е.  $V_{МПД} \approx k_0 V_P$ , что позволяет сверточному МПД при его аппаратной реализации быть в некоторых случаях в 1000 и более раз быстрее турбо кодов.

## 4. КАСКАДНЫЕ СХЕМЫ КОДИРОВАНИЯ

Важным этапом в развитии теории кодирования является появление каскадных кодов [24], в основе построения которых лежит идея совместного использования нескольких составляющих кодов. Данный подход позволил существенно повысить эффективность применения кодирования по сравнению с базовыми некаскадными методами.

Пример использования каскадного кода, состоящего из двух составляющих кодов, показан на рис. 4.1. Здесь данные источника сначала кодируются внешним  $(n_1, k_1)$  кодом. В качестве внешнего кода часто используются недвоичные коды, например, коды Рида-Соломона. Затем закодированные символы внешнего кода кодируются кодером внутреннего  $(n_2, k_2)$  кода. Общая длина кодового слова каскадного кода оказывается равной  $N=n_1n_2$  двоичных символов, причем  $K=k_1k_2$  из них являются информационными. Следовательно, кодовая скорость полученного каскадного кода оказывается равной

$$R = \frac{K}{N} = \frac{k_1k_2}{n_1n_2} = r_1r_2,$$

где  $r_1, r_2$  – кодовые скорости составляющих кодеров.

Также отметим, что минимальное расстояние сформированного каскадного кода будет равно  $D=d_1d_2$ , где  $d_1$  и  $d_2$  – минимальные расстояния составляющих кодов.

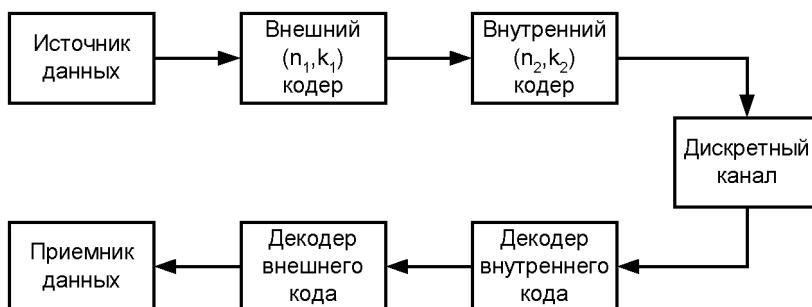


Рис. 4.1. Пример системы связи, использующей каскадный код с двумя составляющими кодами

Декодирование каскадного кода осуществляется в обратном порядке, т.е. принятая из канала последовательность сначала декодируется декодером внутреннего кода, а затем полученная последовательность декодируется декодером внешнего кода. Подчеркнем, что хотя общая длина кода равна  $N$ , структура каскадного кода позволяет применять для декодирования два декодера кодов с длинами всего лишь  $n_1$  и  $n_2$  соответственно. Данное свойство позволяет существенно снизить сложность декодирования по сравнению с сопоставимыми по эффективности декодерами некаскадных блоковых или сверточных кодов.

#### **4.1. Каскадные коды, построенные с использованием кода Рида-Соломона**

Наиболее широкое распространение в реальных системах связи нашла каскадная схема, в которой внешним кодом является код Рида-Соломона, а внутренним – сверточный код, обычно декодируемый с помощью оптимального алгоритма Витерби. Часто в данной схеме между внешним и внутренним кодером/декодером включаются устройства перемежения и восстановления (деперемежения), осуществляющие псевдослучайную перестановку символов внешнего кода и восстановление исходного порядка символов соответственно. Рассмотренные устройства предназначены для разбиения пакетов ошибок, появляющихся при декодировании принятого из канала сообщения с помощью декодера внутреннего кода (алгоритма Витерби), что позволяет существенно улучшить эффективность всей каскадной конструкции.

Для оценки вероятности ошибки каскадного кода, состоящего из кода Рида-Соломона и сверточного кода, можно воспользоваться формулами (2.10) – (2.12), но в данном случае вместо вероятности ошибки в канале  $P_q$  необходимо использовать оценку вероятности ошибки декодирования сверточного кода (3.1). Полученные оценки будут точны только при независимом характере ошибок на входе декодера кода Рида-Соломона, что справедливо лишь при очень большой длине перемежителя. На рис. 4.2 и рис. 4.3 представлены вычисленные вышеописанным образом оценки вероятности битовой ошибки  $P_b$  от отношения

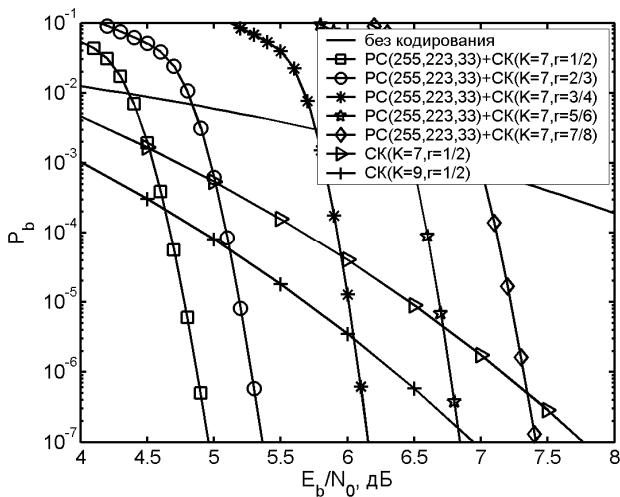


Рис. 4.2. Оценка эффективности каскадного кода, состоящего из кода Рида-Соломона и сверточного кода с  $K=7$  в ДСК

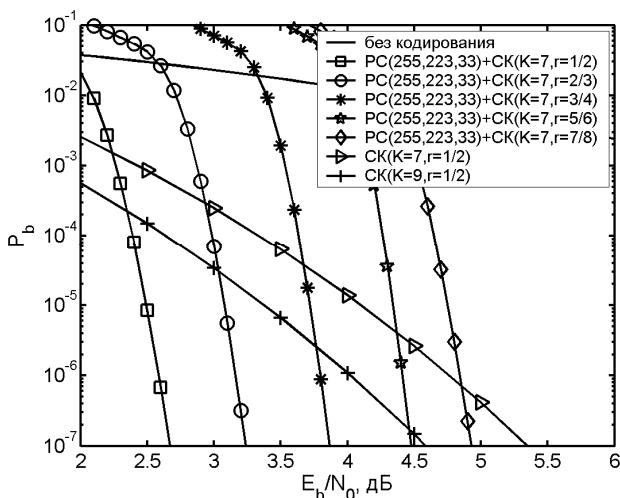


Рис. 4.3. Оценка эффективности каскадного кода, состоящего из кода Рида-Соломона и сверточного кода с  $K=7$  в канале с АБГШ

сигнал-шум на бит  $E_b/N_0$  для каскадного кода, состоящего из кода Рида-Соломона (255, 223, 33) и сверточного кода с конструктивной длиной  $K=7$  в ДСК и канале с АБГШ. На этих рисунках параметры кода Рида-Соломона записаны в виде  $(n, k, d)$ , где  $n$  – общее количество  $q$ -ичных символов в кодовом блоке,  $k$  – количество информационных  $q$ -ичных символов в кодовом блоке,  $d$  – минимальное расстояние кода, а параметрами сверточного кода являются кодовая скорость  $r$  и конструктивная длина  $K$ . Из сопоставления данных графиков с оценкой характеристик сверточного кода с  $K=9$ , также приведенных на рисунках, следует, что уже при  $P_b < 4 \cdot 10^{-5}$  каскадная схема оказывается более эффективной, чем более сложный сверточный код, и с уменьшением  $P_b$  преимущество каскадной схемы становится все более очевидным.

Результаты моделирования каскадного кода, состоящего из кода Рида-Соломона (255, 223, 33) и сверточного кода с  $K=7$  и  $r=1/2$ , при разной глубине *перемежения I* представлены на рис. 4.4. При получении данных графиков использовался обычный прямоугольный перемежитель, представляющий собой массив, состоящий из  $I$  строк (определяющих глубину перемежения) по 255  $q$ -ичных символов (в каждом символе содержится 8 бит, поскольку  $q=256$ ). В процессе кодирования кодовые слова кода Рида-Соломона построчно записывались в массив перемежителя, а затем считывались по столбцам и кодировались с помощью кодера сверточного кода. При декодировании принятая последовательность сначала обрабатывалась декодером внутреннего кода (декодером Витерби). Далее информация с выхода декодера Витерби группировалась в  $q$ -ичные символы и по столбцам записывалась в массив деперемежителя (т.е. первый символ записывается в первый столбец первой строки, второй – в первый столбец второй строки и т.д.), а затем построчно считывалась декодером кода Рида-Соломона и декодировалась. Анализируя данные графики можно заметить, что оценка эффективности рассматриваемого каскадного кода оказывается несколько завышенной. Это можно объяснить неточностью границы для характеристик сверточного кода при очень высоком уровне шума.

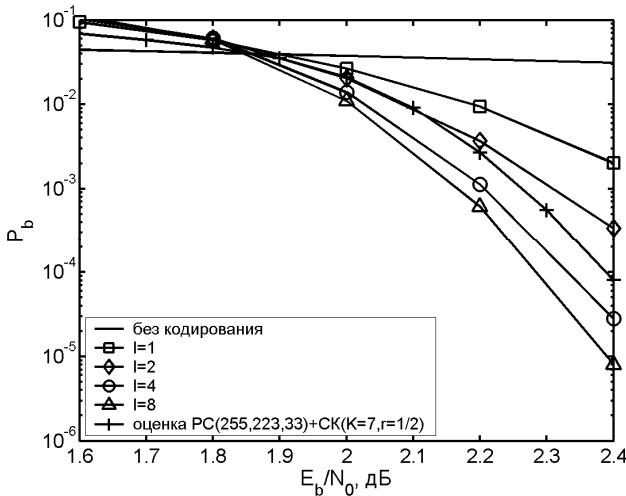


Рис. 4.4. Характеристики каскадного кода, состоящего из кода Рида-Соломона (255,223, 33) и сверточного кода ( $K=7, r=1/2$ ) в канале с АБГШ

#### 4.2. Каскадные коды, декодируемые с использованием многопорогового декодера

Высокие характеристики МПД способствуют его широкому применению в составе различных кодовых конструкций, поскольку эффективность последних непосредственно связана с эффективностью их составных элементов.

Среди возможных подходов к улучшению характеристик МПД следует выделить такие схемы кодирования, как параллельные коды [9], коды с неравномерной энергетикой и коды с выделенными ветвями. Хотя отдельное использование каждого из данных подходов приводит лишь к незначительному ( $0.3 \div 0.7$  дБ) повышению эффективности по сравнению с обычным МПД, совместное применение этих схем позволяет добиться превосходных результатов.

Особое место среди кодовых схем на базе МПД занимает каскадирование с кодами контроля по четности (ККЧ) [11, 12] использование которых также позволяет повысить эффективность применения кодирования. Особенность данной схемы заключается в том, что такое каскадирование практически не требует дополнительных затрат на оборудование (в схему кодирования требуется добавить лишь один сумматор по модулю 2), тогда как использование в каскадном коде, например, кодов Рида-Соломона несравненно труднее.

Кратко опишем принцип работы декодера каскадного кода, внешним кодом которого является ККЧ, а внутренним – СОК, декодируемый с помощью МПД. При этом длину  $n_1$  ККЧ необходимо выбирать достаточно большой (порядка 25÷100) для того, чтобы потери в энергетике из-за уменьшения кодовой скорости были незначительными.

На первом этапе, как в любом каскадном коде, осуществляется декодирование принятой из канала последовательности с помощью декодера внутреннего кода, т.е. МПД. Пусть после последней итерации МПД помнит все суммы проверок относительно всех декодированных символов. Тогда для исправления одиночных ошибок с помощью ККЧ для всех  $n_1$  символов блока данного кода вычисляется достоверность решения  $\Delta_i = |m_i - T|$ , где  $m_i$  – сумма проверок относительно  $i$ -го символа,  $T$  – значение порога на пороговом элементе последней итерации декодирования,  $i=1..n_1$ . Затем, в случае обнаружения с помощью ККЧ ошибки (т.е. в случае отличия от нуля суммы элементов блока ККЧ по модулю 2), исправляется символ, достоверность  $\Delta_i$  которого минимальна. Если есть несколько символов с минимальной достоверностью, то изменение символов не производится.

Рассмотрим методику оценки вероятности ошибки данной схемы при работе в ДСК. При вычислениях предполагаем, что на последней итерации декодирования вероятность ошибки МПД  $P_{MПД}$  равна вероятности ошибки оптимального декодера (данное предположение, как было показано в параграфах 2.8 и 3.4, вполне обосновано), т.е.

$$P_{MПД} = \sum_{i=\frac{d+1}{2}}^d C_d^i p^i (1-p)^{d-i}, \quad (4.1)$$

где  $p$  – вероятность ошибки в канале связи.

Ошибка при использовании ККЧ может произойти в следующих случаях:

1. В блоке ККЧ длиной  $n_1$  находятся две ошибки. Вероятность такого события (ошибки предполагаются независимыми) равна

$$P_{2e} = C_{n_1}^2 (P_{MПД})^2 (1 - P_{MПД})^{n_1-2} = \\ = \frac{n_1(n_1-1)}{2} (P_{MПД})^2 (1 - P_{MПД})^{n_1-2}. \quad (4.2)$$

2. В блоке ККЧ длиной  $n_1$  находится одна ошибка, но достоверность одного или нескольких других символов меньше либо равна достоверности ошибочного символа, т.е. в проверочных символах относительно какого-либо правильно принятого символа не меньше ошибок, чем безошибочных проверок относительно ошибочно принятого символа. Вероятность данного события оценивается сверху, как

$$P_{le} \leq n_1(n_1-1) \sum_{i=\frac{d+1}{2}}^d \left( C_d^i p^i (1-p)^{d-i} \sum_{j=d-i}^{\frac{d-1}{2}} C_d^j p^j (1-p)^{d-j} \right). \quad (4.3)$$

Другими сочетаниями ошибок канала при работе в области оптимального декодирования МПД можно пренебречь.

Так как в данных случаях количество ошибок либо остается прежним, либо увеличивается, вероятность ошибки декодирования каскадной схемы будет не больше, чем

$$P_b \leq \frac{2(P_{2e} + P_{le})}{n_1}. \quad (4.4)$$

На рис. 4.5 представлены верхние оценки вероятностей ошибки для СОК с  $d=7, 9, 11$  (здесь и далее считается, что данные СОК декодируются с помощью МПД) и каскадной схемы на его основе (длина ККЧ  $n_1=25$ ). На этом же рисунке показаны результаты моделирования каскадного кода на базе СОК с  $r=1/2$ ,

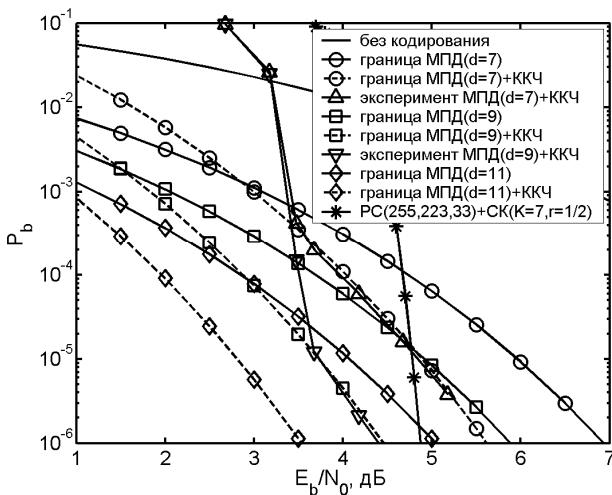


Рис. 4.5. Оценка эффективности и результаты моделирования каскадного кода на базе МПД в ДСК

$d=7$  и  $d=9$ . Заметим, что использование простейшего ККЧ совместно с СОК позволяет получить дополнительный энергетический выигрыш порядка  $1\div 1.5$  дБ при вероятности ошибки на выходе декодера  $P_b=10^{-5}$ .

Эффективность работы каскадной схемы, состоящей из тех же СОК, что и на рис. 4.5 и ККЧ с  $n_1=50$ , для канала с АБГШ отражена на рис. 4.6. Как видно из рисунков, и в данном случае каскадный код оказывается значительно лучше некаскадного. Следует отметить, что при получении представленных графиков ККЧ использовался на нескольких итерациях декодирования, тем самым как бы «помогая» МПД при декодировании внутреннего СОК. Также отметим, что каскадный код, состоящий из кода Рида-Соломона (255, 223, 33) и сверточного кода с кодовой скоростью  $1/2$  и длиной кодового ограничения  $K=7$ , декодируемый с помощью оптимального алгоритма Витерби, даже при меньшей общей кодовой скорости ( $R\approx 0.437$ ) сильно уступает

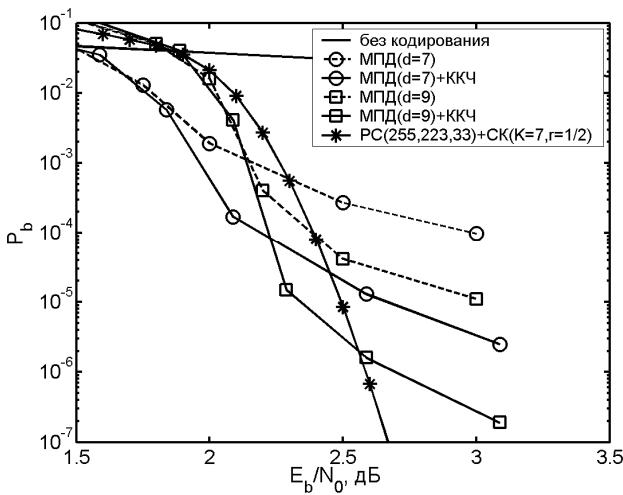


Рис. 4.6. Результаты моделирования каскадного кода на базе МПД в канале с АБГШ

каскадной схеме на базе МПД при  $P_b > 10^{-6}$ . К сожалению, рассмотренный способ каскадирования позволяет улучшить характеристики МПД только в области его эффективной работы.

Сложность декодирования рассмотренной каскадной схемы по сравнению со сложностью обычного МПД увеличивается на сложность декодера ККЧ. Из описания принципа работы данного декодера ясно, что при декодировании блока ККЧ, длиной  $n_1$  символов, требуется выполнить  $n_1-1$  сложений по модулю 2 и 1 сравнение при определении наличия ошибки в блоке,  $n_1-1$  операций выбора максимума из двух чисел для определения символа с минимальной достоверностью и  $d+1$  операций сложения по модулю 2 (инверсий) в случае необходимости коррекции информационного символа (также корректируются элементы разностного и синдромного регистров). Причем, так как коррекция осуществляется с вероятностью  $P_{МПД} \ll 1$ , при определении общей сложности ей можно пренебречь. В результате при деко-

дировании одного информационного бита декодером ККЧ в среднем требуется всего  $\frac{2n_1-1}{n_1-1} \approx 2$  операции, эквивалентных сложению.

### 4.3. Применение многопорогового декодера в схемах с параллельным кодированием

Для приближения границы эффективной работы МПД к пропускной способности канала возможно его применение в ранее упомянутых схемах параллельного кодирования [9]. В основе построения данных схем лежит выделение в СОК  $C_0$  с кодовым расстоянием  $d_0$  и кодовой скоростью  $r_0$  некоторого составляющего кода  $C_1$  с кодовой скоростью  $r_1 > r_0$ , тоже являющегося СОК. Кодовое расстояние  $d_1$  выделенного кода выбирается значительно меньшим  $d_0$ , и, следовательно, область его эффективной работы будет ближе к границе Шеннона. При декодировании параллельного кода сначала выполняются несколько итераций декодирования составляющего кода  $C_1$ , позволяющие примерно на порядок снизить вероятность ошибки в принятой из канала информационной последовательности, после чего в процесс декодирования включается оставшаяся часть кода  $C_0$ . Отличительной особенностью данной схемы кодирования является то, что здесь внешний код работает с кодовой скоростью  $r_0$ , в то время как в обычных каскадных кодах кодовая скорость внешнего кода близка к единице. Данное свойство обеспечивает существенное преимущество МПД перед другими каскадными конструкциями.

Для примера на рис. 4.7 и рис. 4.8 представлены результаты моделирования схем с параллельным кодированием в ДСК и канале с АБГШ для СОК с  $r_0=6/12$ ,  $d_0=13$  и  $r_0=5/10$ ,  $d_0=15$ . В параллельном коде с  $d_0=13$  в данном случае был выделен внешний код с  $r_1=6/11$ ,  $d_1=7$ , а в коде с  $d_0=15$  был выделен код с  $r_1=5/9$ ,  $d_1=9$ . Кривые «составляющий» на рисунках отражают вероятность ошибки на выходе выделенных кодов параллельной схемы. Пунктирными линиями без маркеров на данных рисунках показаны вероятности ошибки оптимального декодирования

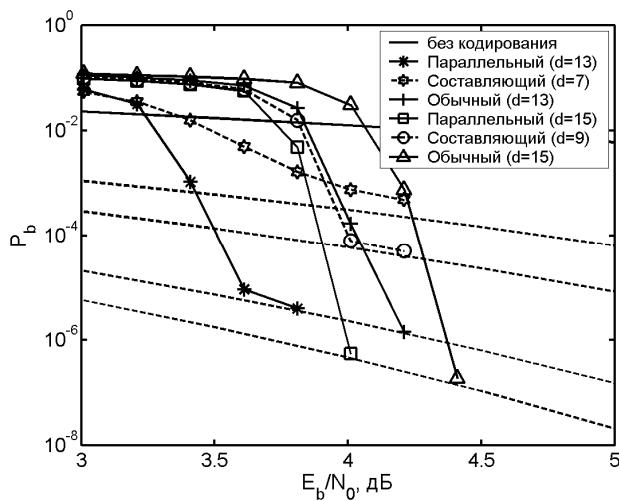


Рис. 4.7. Результаты моделирования параллельного кода на базе МПД в ДСК

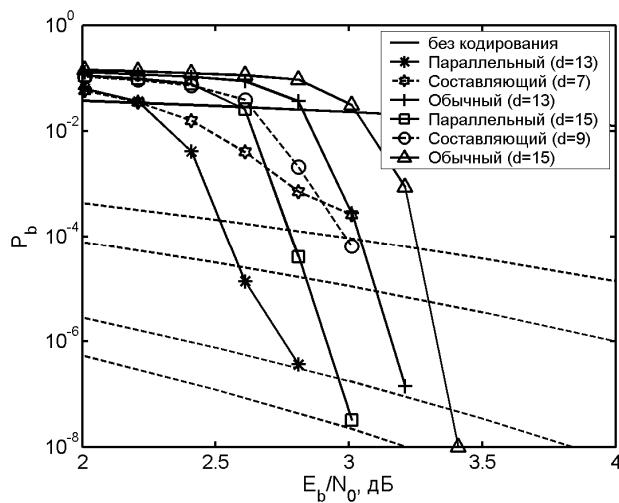


Рис. 4.8. Результаты моделирования параллельного кода на базе МПД в канале с АБГШ

кодов с  $d=7, 9, 11, 13$  и  $15$ . Для сравнения на рис. 4.7 и рис. 4.8 также показаны характеристики декодируемых с помощью МПД обычных СОК с аналогичными  $d$  и  $r$ .

Как следует из анализа представленных графиков, применение параллельного кодирования позволяет приблизить границу эффективной работы МПД к пропускной способности канала примерно на  $0.5$  дБ.

Сложность МПД при параллельном кодировании (в смысле количества выполняемых операций) оказывается даже меньше сложности обычного МПД, поскольку в данном случае на первых итерациях декодирования некоторые элементы синдромного регистра просто не участвуют в процессе вычисления суммы на пороговом элементе.

#### 4.4. Турбо коды

Турбо коды [30] образуются при параллельном каскадировании двух или более составляющих систематических кодов. Рассмотрим общую схему кодера турбо кода, изображенную на рис. 4.9. Как показано на рисунке, передаваемые данные перед кодированием каждым из составляющих кодов перемешиваются с помощью входящих в состав кодера перемежителей. Поскольку систематические выходы каждого из кодеров с учетом перемежения идентичны, в канал можно передавать только исходную последовательность и проверочные выходы каждого из ко-

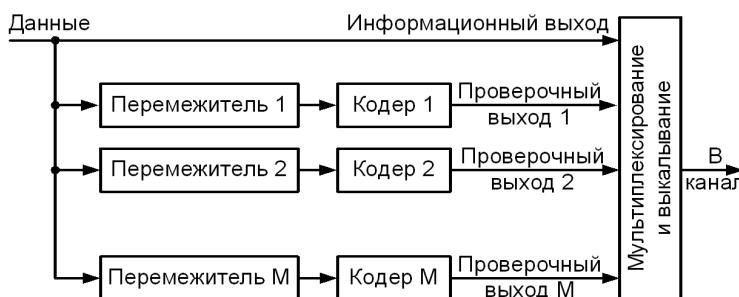


Рис. 4.9. Общая схема кодера турбо кода

деров. В результате общая кодовая скорость турбо кода при использовании составляющих кодов со скоростью  $1/2$  оказывается равной  $R=1/(M+1)$ , где  $M$  – количество составляющих кодеров. Затем систематические и проверочные биты от всех составляющих кодеров объединяются в одно кодовое слово, передаваемое по каналу связи.

Обычно при построении турбо кодера используются два одинаковых рекурсивных систематических сверточных (RSC) кодера без перемежения в первой ветви. Применение таких составляющих кодов совместно с перемежителем позволяет уменьшить число кодовых слов низкого веса, определяющих эффективность турбо кода при большом уровне шума в канале связи [53].

Рассмотрим турбо код, кодер которого представлен на рис. 4.10. Данный код является стандартным турбо кодом с кодовой скоростью  $1/3$ , включенным в стандарты систем мобиль-

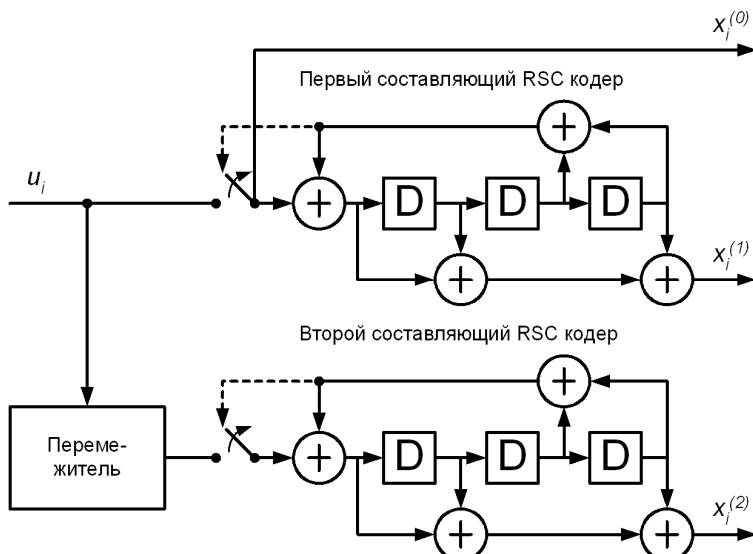


Рис. 4.10. Схема кодера турбо кода, состоящего из двух рекурсивных систематических сверточных кодеров

ной связи третьего поколения [40, 57]. Для построения указанного кода используются два одинаковых RSC кода с конструктивной длиной  $K=4$ , заданных образующими полиномами  $g_0=13$ ,  $g_1=15$ . Как видно из рис. 4.10, выходом турбо кодера является последовательность, состоящая из систематических и проверочных бит первого кодера и проверочных бит второго кодера. Таким образом, общая кодовая скорость турбо кода равна  $R=1/3$ . При необходимости использования более высоких кодовых скоростей обычно применяют *выкалывание*, при котором часть проверочных символов не передается в канал. В этом случае перед декодированием вместо пропущенных проверочных символов вставляются нули.

В начале кодирования информационного блока, длина которого определяется размером  $L$  используемого перемежителя, оба составляющих кодера находятся в нулевом состоянии, т.е. их регистры сдвига заполнены нулями. После кодирования последнего информационного бита в блоке каждый декодер может оказаться в любом из  $2^{K-1}$  (в данном случае 8) состояний. Однако эффективность декодера будет значительно лучше, если будет известно не только начальное, но и конечное состояния кодеров. Поэтому в турбо кодер добавляется схема перевода составляющих кодеров в нулевое состояние, при работе которой (т.е. при переводе ключей в нижние позиции) на вход каждого из кодеров поступают биты из ветви обратной связи (данные биты называются хвостовыми). Заметим, что из-за присутствия перемежителя составляющие кодеры в общем случае будут находиться в разных состояниях, т.е. для их перевода в нулевое состояние требуются разные хвостовые биты. Поэтому в передаваемый поток добавляются хвостовые биты каждого из декодеров и соответствующие им проверочные биты, что приводит к снижению общей кодовой скорости рассматриваемого турбо кода до величины

$$R = \frac{L}{3L+12}.$$

Однако при больших значениях  $L$  данное уменьшение кодовой скорости пренебрежимо мало.

Для декодирования турбо кода с рис. 4.10 можно использовать декодер, изображенный на рис. 4.11 [61]. Декодер представляет собой каскадное соединение двух элементарных декодеров, двух перемежителей, идентичных используемым в кодере, и двух деперемежителей, осуществляющих восстановление исходного (до перемежения) порядка символов. Как видно из рисунка, декодер каждого составляющего кода имеет три входа: принятые из канала информационные биты, канальные проверочные биты от соответствующего кодера и информация от другого декодера, оценивающая значение информационных бит (данная информация называется *априорной*). Единственным выходом компонентного декодера являются мягкие решения относительно декодируемых бит, причем для представления мягких решений обычно используется логарифм отношения правдоподобия, знак которого определяет значение декодированного бита (отрицательное значение соответствует нулю, положительное – единице), а модуль – надежность этого значения.

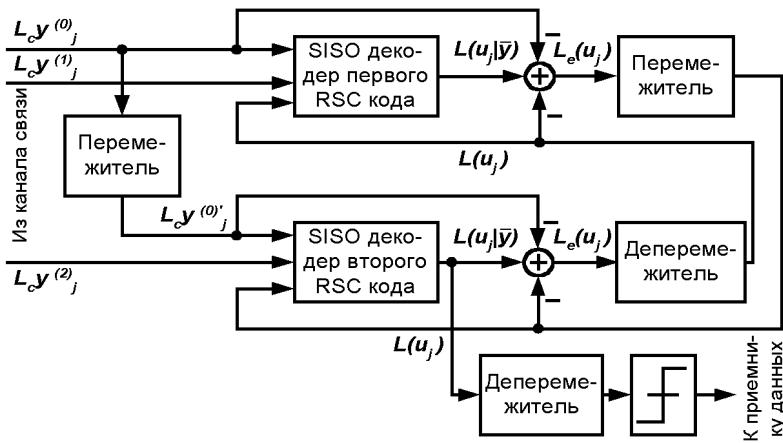


Рис. 4.11. Схема декодера турбо кода

Логарифм отношения правдоподобия  $L(u_k)$  для информационного символа  $u_k$ , как следует из его названия, определяется следующим образом:

$$L(u_k) = \ln \left[ \frac{P(u_k = +1)}{P(u_k = -1)} \right], \quad (4.5)$$

где  $P(u_k = m)$  – вероятность того, что  $u_k = m$  ( $m = \pm 1$ ).

Принцип работы декодера турбо кода заключается в выполнении нескольких итераций декодирования, для первой из которых на входе декодера первого составляющего кода отсутствует априорная информация, т.е. он формирует оценку информационных бит, используя только принятую из канала последовательность. Далее из полученной оценки выделяется так называемая *внешняя* информация, определяемая исключением из оценки декодируемых символов априорной информации (на первой итерации априорная информация равна нулю) и принятых из канала систематических символов:

$$L_e(u_k) = L(u_k | \bar{y}) - L_{C_{y_{ks}}} - L(u_k),$$

где  $L_c$  определяет надежность канала (для канала с АБГШ  $L_c = 2/\sigma^2$ , где  $\sigma^2$  – дисперсия шума).

Затем декодер второго кода использует эту внешнюю информацию в качестве априорной для получения своей оценки информационных бит.

На второй итерации первый декодер снова обрабатывает принятую из канала последовательность, но уже с учетом априорной информации, сформированной из оценки второго декодера на первой итерации. Эта дополнительная информация позволяет первому декодеру получить более точную оценку декодируемых бит, используемую затем вторым декодером в качестве априорной.

Далее декодер турбо кода выполняет еще несколько итераций декодирования, на каждой из которых осуществляется уточнение оценки информационной последовательности (т.е. средняя вероятность ошибки декодирования постепенно уменьшается). К сожалению, с каждой новой итерацией улучшение от ее применения становится все меньше. Поэтому, для снижения сложности декодера, итеративный процесс прекращается после

выполнения некоторого небольшого (10 – 20) количества итераций. Также для завершения декодирования можно использовать некоторый критерий, определяющий целесообразность применения дополнительной итерации, что часто позволяет существенно сократить среднее число итераций и сложность декодера при незначительных потерях в эффективности. После выполнения заданного количества итераций или вынесении решения о нецелесообразности продолжения итеративного процесса жесткое решение о восстановленной после перемежения оценке второго декодера будет определять результат декодирования.

В качестве декодеров составляющих кодов турбо кода, применяемых в описанной схеме, можно использовать такие алгоритмы как MAP (Maximum A Posteriori) [27], Log-MAP [51], Max-Log-MAP [48, 39], SOVA (Soft Output Viterby Algorithm) [42]. Сложность реализации последних трех методов в смысле количества выполняемых операций для декодирования одного информационного символа представлена в табл. 4.1. MAP алгоритм значительно превосходит остальные по сложности при одинаковой с Log-MAP алгоритмом эффективностью и поэтому

Табл. 4.1. Сложность алгоритмов декодирования  
помехоустойчивых кодов

Вид операции	Алгоритм Витерби	Max-Log-MAP	Log-MAP	SOVA
Выбор максимума		$5 \cdot 2^{K-1} - 2$	$5 \cdot 2^{K-1} - 2$	$2^{K-1} + 3K$
Сложение	$2 \cdot 2^{K-1} + 5$	$10 \cdot 2^{K-1} + 11$	$15 \cdot 2^{K-1} + 9$	$2 \cdot 2^{K-1} + 8$
Умножение на $\pm 1$		8	8	8
Сравнение	$2^{K-1}$			$6K$
Поиск в таблице			$5 \cdot 2^{K-1} - 2$	

на практике обычно не используется. Зная сложность реализации алгоритмов декодирования составляющих кодов  $N_{\text{сост}}$  можно оценить сложность декодирования всего турбо кода:

$$N_{\text{турбо}} = I N_{\text{сост}} M, \quad (4.6)$$

где  $I$  – количество итераций декодирования,  $M$  – число составляющих кодов.

Для аналитической оценки эффективности турбо кодов можно воспользоваться выражением для аддитивной границы вероятности битовой ошибки блокового кода

$$P_b \leq \sum_{i=d}^N \frac{w_i}{L} P_i, \quad (4.7)$$

где  $N$  – длина кодового блока,  $L$  – количество информационных символов блока,  $d$  – минимальное кодовое расстояние,  $w_i$  – общий информационный вес всех кодовых слов веса  $i$ ,  $P_i$  – вероятность выбрать неправильное кодовое слово, отличающееся от правильного в  $i$  позициях, определяемая по формуле (1.33).

Для турбо кода с кодовой скоростью  $R=1/2$ , состоящего из двух одинаковых кодеров с конструктивной длиной  $K$  и переключателя длиной  $L$ , длина кодового блока будет равна  $N=2(L+K-1)$ . Учитывая, что общий информационный вес  $w_i$  всех кодовых слов веса  $i$  равен  $w_i = \tilde{w}_i N_i$ , где  $\tilde{w}_i$  – средний информационный вес кодовых слов веса  $i$ ,  $N_i$  – общее количество кодовых слов веса  $i$ , перепишем выражение (4.6) следующим образом:

$$P_b \leq \sum_{i=d}^N \frac{\tilde{w}_i N_i}{L} P_i. \quad (4.8)$$

При вычислении границы (4.8) часто ограничиваются только первым слагаемым, хорошо аппроксимирующим вероятность битовой ошибки при средних и высоких значениях отношения сигнал-шум, и называемым *асимптотой кодового расстояния*:

$$P_b \approx \frac{\tilde{w}_d N_d}{L} P_d. \quad (4.9)$$

Для примера рассмотрим оценку эффективности турбо кода с кодовой скоростью  $R=1/2$ , полученного из двух одинаковых составляющих RSC кодов с образующими полиномами  $g_0=37_8$  и

$g_1=21_8$  и псевдослучайного перемежителя длиной  $L=65536$  бит. Данный код, используемый в работе [30], в дальнейшем будем обозначать как  $(37, 21, 65536)$ . Поскольку для  $(37, 21, 65536)$  турбо кода  $d=6$ ,  $\tilde{w}_d = 2$ ,  $N_d=3$  [53], выражение (4.8) для канала с АБГШ принимает вид

$$P_b = \frac{2 \cdot 3}{65536} Q\left(\sqrt{6 \frac{E_b}{N_0}}\right). \quad (4.10)$$

График зависимости оценки вероятности ошибки от отношения сигнал-шум на бит  $E_b/N_0$ , полученный по формуле (4.10) и результаты моделирования  $(37, 21, 65536)$  турбо кода [53] для случая использования Log-MAP алгоритма декодирования составляющих кодов и 18 итераций декодирования представлены на рис. 4.12. Превосходные характеристики турбо кода в области больших шумов объясняются очень малым отношением  $\tilde{w}_i N_i / L$  для кодовых слов низкого веса, вызванного использо-

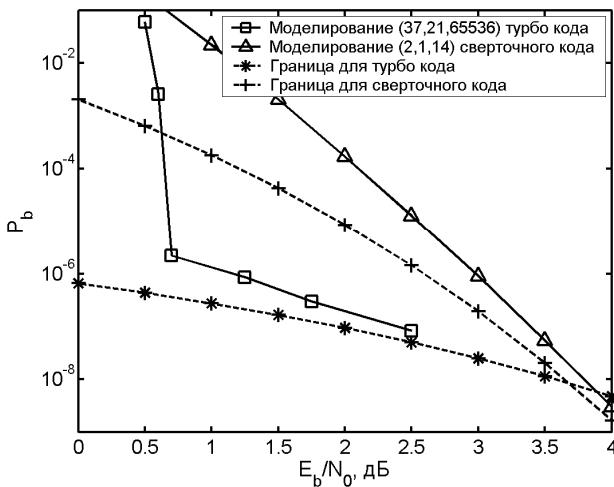


Рис. 4.12. Границы вероятности ошибки и результаты моделирования  $(31, 27, 65536)$  турбо кода и  $(2, 1, 14)$  сверточного кода

ванием рекурсивных составляющих кодеров, часто позволяющих получить для информационных последовательностей низкого веса проверочные последовательности с высоким весом, и наличием перемежителя, с высокой вероятностью отображающего проверочные последовательности низкого веса первого составляющего кода в проверочные последовательности высокого веса второго составляющего кода [53].

Из рисунка следует, что асимптота кодового расстояния при умеренных и высоких значениях отношения сигнал-шум хорошо подходит для нижней оценки характеристик турбо кода. На рис. 4.12 также можно наблюдать проявление свойственного турбо кодам эффекта насыщения вероятности ошибки, заключающегося в уменьшении угла наклона кривой характеристик турбо кода после достижения некоторого значения вероятности ошибки на выходе декодера. Данный эффект, как следует из анализа выражения (4.10), возникает из-за относительно небольшой величины кодового расстояния турбо кода, поскольку указанная величина влияет на наклон графика асимптоты. Кроме кодового расстояния на область насыщения вероятности ошибки большое влияние оказывает коэффициент  $\tilde{w}N_d/L$ , поскольку его увеличение поднимает асимптоту кодового расстояния, а уменьшение – понижает. Таким образом, для увеличения эффективности турбо кода в области средних и малых шумов можно увеличивать кодовое расстояние  $d$  или длину используемого перемежителя  $L$ .

На рис. 4.12 также представлена граница вероятности ошибки и результаты моделирования (2, 1, 14) сверточного кода с кодовой скоростью  $r=1/2$  и конструктивной длиной кода  $K=14$ , декодируемого с помощью алгоритма Витерби. Сложность декодера Витерби для данного кода сопоставима со сложностью декодера (31, 27, 65536) турбо кода. При вычислении границы сверточного кода также использовалось только первое ненулевое слагаемое в (3.1), которое для данного кода принимает вид

$$P_b = 187 Q\left(\sqrt{18 \frac{E_b}{N_0}}\right), \quad (4.11)$$

поскольку  $d_{free}=18$  и  $w_{18}=187$  [53].

Сравнение асимптот рассмотренных кодов показывает, что из-за большего кодового расстояния угол наклона асимптоты сверточного кода значительно больше, чем асимптота турбо кода. Однако, так как количество кодовых слов низкого веса для турбо кода гораздо меньше, чем для сверточного, две асимптоты не пересекаются до отношения сигнал-шум порядка 3.5 dB. При этом вероятность ошибки при декодировании рассматриваемых кодов оказывается ниже  $10^{-7}$ , что меньше требуемой вероятности ошибки для многих реальных систем связи. Следовательно, несмотря на то, что при сопоставимой сложности сверточные коды асимптотически лучше турбо кодов, во многих практических случаях целесообразно использовать именно турбо коды.

Большое значение при разработке турбо кодов уделяют выбору типа перемежителя, поскольку он оказывает огромное влияние на минимальное расстояние и количество кодовых слов низкого веса, определяющих эффективность турбо кода. Все типы перемежителей в зависимости от способа построения можно условно разделить на два класса: псевдослучайные (например,  $S$ -random [38]) и регулярные (блочный, циклический, диагональный и т.д. [33, 45, 56, 28, 35]). Зависимость позиций битов в перемеженном блоке от их позиций в исходном блоке для регулярных перемежителей имеет ярко выраженную структуру (рис. 4.13). При использовании псевдослучайных перемежителей биты исходного блока в перемеженном блоке располагаются в псевдослучайном порядке (рис. 4.14), что положительно сказывается на общей эффективности турбо кода.

Представим результаты моделирования турбо кода в канале с АБГШ при использовании двоичной фазовой модуляции. Как следует из описания кодера и декодера турбо кодов, на их эффективность оказывают влияние много параметров, основными из которых являются алгоритм декодирования составляющих кодов, количество итераций декодирования, тип и длина перемежителя, составляющие коды. Далее рассмотрим характеристики турбо кодов, по умолчанию имеющих следующие параметры: кодовая скорость  $R=1/2$ , конструктивная длина составляющих рекурсивных систематических сверточных кодов  $K=4$ ,

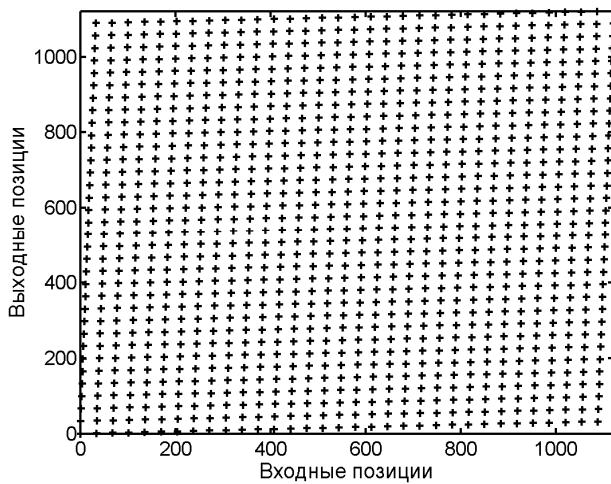


Рис. 4.13. Зависимость выходной позиции бита в блоке от его входной позиции для регулярного перемежителя

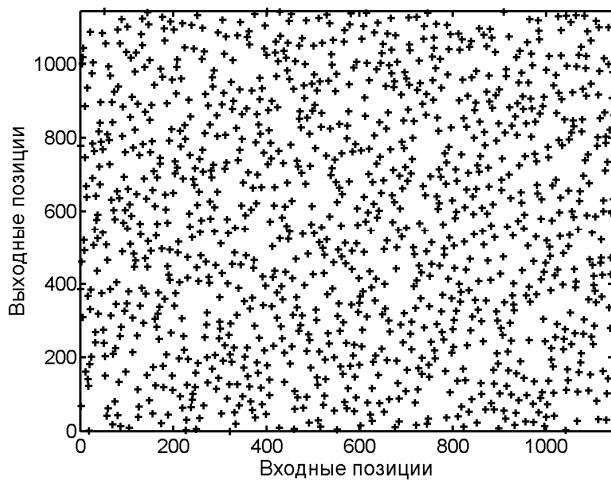


Рис. 4.14. Зависимость выходной позиции бита в блоке от его входной позиции для псевдослучайного перемежителя

образующие полиномы составляющих кодеров  $g_0=13$  и  $g_1=15$ , псевдослучайный перемежитель длиной  $L=1146$  бит, Max-Log-MAP алгоритм декодирования составляющих кодов, 8 итераций декодирования.

Влияние количества итераций декодирования на эффективность турбо кода показано на рис. 4.15. Как видно из рисунка, характеристики турбо кода уже после одной итерации декодирования сопоставимы с характеристиками составляющего сверточного кода с  $K=4$ , и с увеличением количества вероятность ошибки постепенно уменьшается. Отметим, что уже после восьми итераций это уменьшение становится незначительным. Поэтому для всех далее приводимых результатов по умолчанию предполагается использование восьми итераций декодирования.

На рис. 4.16 представлены результаты моделирования турбо кода со стандартными параметрами при использовании различных алгоритмов декодирования. Заметим, что MAP и Log-MAP алгоритмы обладают одинаковыми характеристиками, а Max-

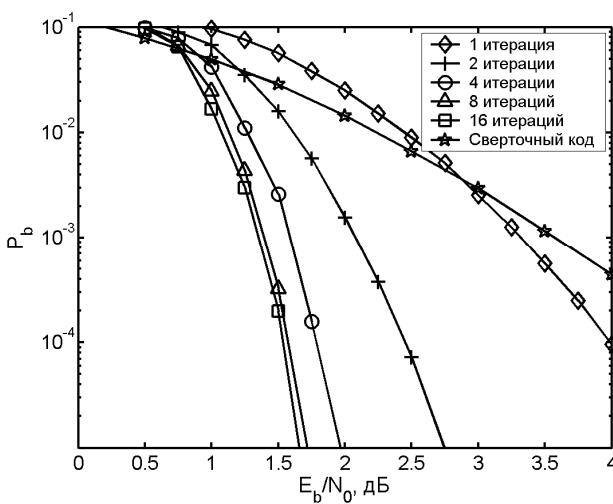


Рис. 4.15. Характеристики турбо кода при использовании различного числа итераций декодирования

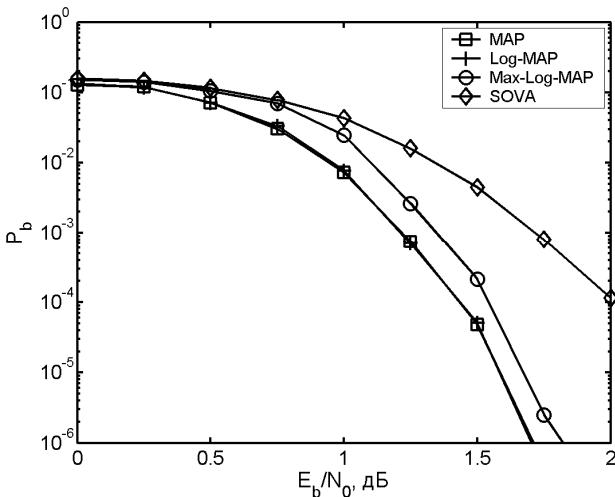


Рис. 4.16. Характеристики турбо кода при использовании различных алгоритмов декодирования составляющих кодов

Log-MAP и SOVA алгоритмы уступают при вероятности ошибки  $P_b=10^{-4}$  им примерно 0.1 и 0.6 дБ соответственно.

На рис. 4.17 показаны характеристики турбо кода для нескольких типов перемежителей различной длины. Видно, что при небольших длинах кодов эффективность перемежителей рассмотренных типов примерно одинакова, а с увеличением длины значительно лучшие результаты показывают псевдослучайные перемежители.

Влияние длины перемежителя на эффективность турбо кода показано на рис. 4.18. Для сравнения на данном рисунке также представлены характеристики сверточного кода с  $K=11$ , декодируемого с помощью алгоритма Витерби. Данный код был выбран потому, что  $K=11$  является наименьшей конструктивной длиной, при которой сложность декодирования сверточного кода, исходя из табл. 4.1, становится больше сложности декодирования рассматриваемого турбо кода. Как видно из рисунка, турбо код с длиной блока всего 190 бит по эффективности соответ-

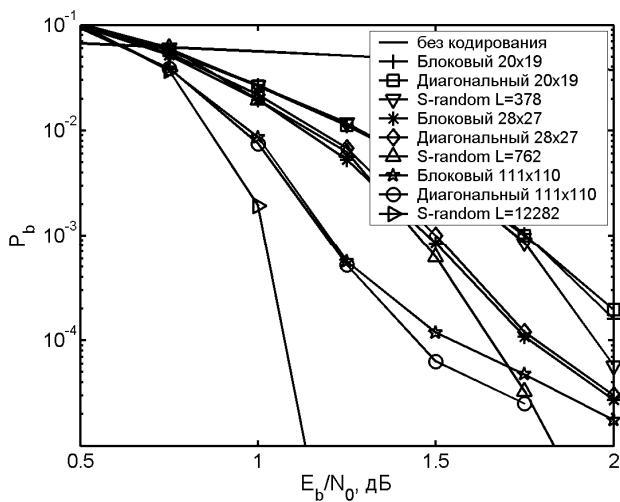


Рис. 4.17. Характеристики турбо кода при использовании различных типов перемежителей

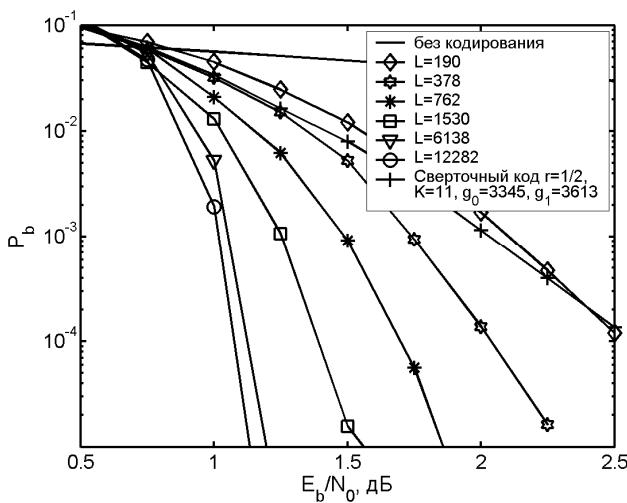


Рис. 4.18. Характеристики турбо кода при использовании различных длин S-random перемежителей

ствует сверточному коду с  $K=11$ , а с ростом длины перемежителя характеристики турбо кода значительно улучшаются.

Из рис. 4.19, отражающего влияние конструктивной длины составляющих кодов на эффективность турбо кода при использовании Log-MAP алгоритма декодирования, следует, что эффективность турбо кодов с  $K=4$  и 5 несколько больше, чем эффективность кодов с  $K=3$ . Также видно, что коды с  $K=4$  и 5 обладают примерно одинаковой эффективностью, но так как сложность декодера турбо кода экспоненциально зависит от  $K$ , для построения турбо кода в реальных системах связи часто используют турбо коды, построенные на базе RSC кодов с  $K=4$  [40, 57].

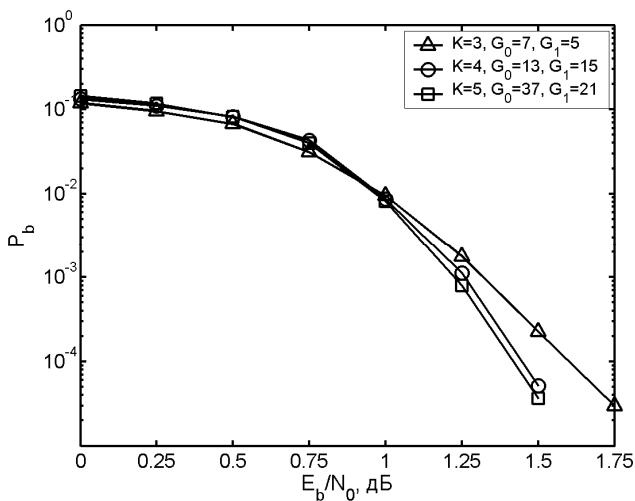


Рис. 4.19. Характеристики турбо кода при использовании различных составляющих кодов

## 4.5. Сравнение сложности реализации эффективных методов декодирования помехоустойчивых кодов

В данном параграфе проведем сравнение сложности практической реализации [13] нескольких наиболее эффективных методов декодирования, таких как алгоритм декодирования Витерби [4], методы декодирования турбо кодов [30] и многопороговый алгоритм декодирования (МПД) [12, 22]. Под сложностью реализации здесь будем понимать количество арифметических операций, требуемых для декодирования одного информационного бита.

Из табл. 4.1 видно, что сложность оптимального алгоритма Витерби растет экспоненциально с ростом конструктивной длины кода  $K$ , в связи с чем на практике данный метод обычно применяется для декодирования кодов с  $K \leq 9$ , обладающих низкой эффективностью.

Сложность декодирования турбо кодов определяется сложностью методов декодирования составляющих кодов и количеством итераций декодирования. Вычислительная сложность Log-MAP, Max-Log-MAP и SOVA алгоритмов [51], используемых для декодирования составляющих кодов, представлена в табл. 4.1. Очевидно, что относительная сложность данных алгоритмов также экспоненциально растет с ростом конструктивной длины  $K$  составляющих кодов. Однако для построения турбо кода обычно применяются коды с небольшим  $K=3 \div 5$ , так как на его характеристики основное влияние оказывает длина используемого перемежителя. Поэтому сложность данного декодера оказывается значительно меньше сложности декодера Витерби при равной эффективности.

Для МПД, как было показано в параграфах 2.8 и 3.4, количество операций, выполняемых при декодировании одного информационного бита, в основном зависит от кодового расстояния  $d$  используемого кода и количества итераций декодирования  $I$ , т.е.  $N_{\text{МПД}} \approx (I+1)(d+2)$ . Поскольку МПД обычно применяется для декодирования кодов с  $d=7 \div 15$  при  $I < 20$ , его сложность оказывается примерно на порядок меньше сложности сопоставимого по эффективности декодера турбо кода. Кроме того, небольшая модификация МПД при пренебрежимо малых

потерях в энергетике ( $\sim 0.1$  дБ) позволяет снизить суммарное количество операций, выполняемых на всех итерациях декодирования, до величины порядка  $N_{\text{МПД}} \approx 4d + 3I$ . В результате данных изменений сложность МПД при высоком уровне шума становится в десятки раз меньшей сложности всех других методов коррекции ошибок.

Для примера в табл. 4.2 и табл. 4.3 приведены результаты исследования скорости работы декодера турбо кода и «быстро-го» многопорогового декодера для кодовых скоростей  $1/2$  и  $3/4$ . В последней строке табл. 4.2 представлены результаты для каскадного кода, состоящего из СОК, декодируемого с помощью МПД и кода с контролем четности. Данные результаты были получены на компьютере с процессором Athlon 850 при декодировании  $10^9$  информационных бит при работе в канале с АБГШ и использовании двоичной фазовой модуляции. Отметим, что скорость декодирования получилась в несколько раз ниже, чем можно было бы ожидать исходя из представленных выше формул, поскольку процессорное время также расходовалось на кодирование информации, ее искажение в канале связи и некоторые другие вспомогательные операции (например, определение вероятности ошибки  $P_b$ ).

Как следует из табл. 4.2 и табл. 4.3, при кодовой скорости  $1/2$  более мощный турбо за счет использования меньшего количества итераций и составляющих кодов с небольшой конструктивной длиной  $K$  оказывается медленнее МПД примерно в 13 раз. При переходе с кодовой скорости  $3/4$  для получения сопоставимой с МПД эффективности в декодере турбо кода необходимо применять коды с большей конструктивной длиной и использовать большее количество итераций, что приводит к существенному замедлению процесса декодирования, в то время как скорость работы МПД из-за использования кодов с меньшей длиной кодового расстояния  $d$  при том же количестве итераций несколько возрастает. В результате МПД оказывается быстрее декодера турбо кода в 40 и более раз.

Табл. 4.2. Результаты моделирования МПД и каскадного кода с кодом с контролем четности на его основе

$E_b/N_0$ , дБ	$r$	$d$	$I$	Использовался ККЧ	Скорость декодирования, кбит/с	Вероятность ошибки $P_b$
3.0	1/2	9	8	нет	2420	$1.6 \cdot 10^{-5}$
2.5	1/2	9	14	нет	870	$4.2 \cdot 10^{-5}$
3.4	3/4	7	17	нет	780	$5.1 \cdot 10^{-6}$
3.3	3/4	7	16	да	760	$4.0 \cdot 10^{-8}$

Табл. 4.3. Результаты моделирования декодера турбо кода

$E_b/N_0$ , дБ	$r$	Длина перемежителя $L$	Конструктивная длина кодов $K$	$I$	Скорость декодирования, кбит/с	Вероятность ошибки $P_b$
3.0	1/2	1000	3	4	187	$6.0 \cdot 10^{-7}$
2.5	1/2	1000	3	8	68	$2.1 \cdot 10^{-6}$
3.4	3/4	1000	5	10	9	$1.2 \cdot 10^{-5}$
3.4	3/4	1000	4	10	18	$2.0 \cdot 10^{-5}$
3.4	3/4	2500	4	10	18	$1.3 \cdot 10^{-6}$

## Заключение

Представленный вниманию специалистов справочник является фактически первой попыткой в нашей стране систематически и кратко описать обширные возможности современных методов помехоустойчивого кодирования для систем связи самого широкого назначения.

Он выходит в свет в то время, когда динамичный конкурс между несколькими наиболее известными алгоритмами декодирования по критериям эффективности и сложности реализации становится особенно напряженным. В этой ситуации становится особенно важным, насколько реально полезными для создателей систем современной связи окажутся те алгоритмы, которые в настоящее время демонстрируют наиболее обнадеживающие характеристики.

Хотелось бы надеяться, что настоящий справочник поможет специалистам, создающим аппаратуру и программное обеспечение для систем и сетей связи, выбрать наиболее соответствующие их потребностям алгоритмы декодирования для будущих высококачественных и очень надежных систем телекоммуникаций и информатики. В этом случае авторы справочника будут считать, что свою задачу ускорения продвижения самых последних идей теории и достижений технологии кодирования в системы связи они выполнили.

## Список литературы

1. Банкет В.Л., Дорофеев В.М. Цифровые методы в спутниковой связи. – М.: Радио и связь, 1988. 240 с.
2. Берлекэмп Э.Р. Техника кодирования с исправлением ошибок // ТИИЭР, Т. 68, №5, 1980. – С. 24-58.
3. Боуз Р.К., Рой-Чоудхури Д.К. Об одном классе двоичных групповых кодов с исправлением ошибок // Кибернетический сборник. – 1961. – Вып. 2. – С. 83-94.
4. Витерби А. Границы ошибок для сверточных кодов и асимптотически оптимальный алгоритм декодирования // Некоторые вопросы теории кодирования. – М., 1970. С. 142-165.
5. Золотарев В.В. Алгоритмы кодирования символьных данных в вычислительных сетях // В сб.: «Вопросы кибернетики», ВК-106. – М., 1985.
6. Золотарев В.В. Использование помехоустойчивого кодирования в технике связи // Электросвязь, №7. – М., 1990, С.7-10.
7. Золотарев В.В. Коды и кодирование. - М.: Знания, 1990. 64 с.
8. Золотарев В.В. Многопороговое декодирование в недвоичных каналах // В сб.: «Вопросы радиоэлектроники», Серия ЭВТ, вып. 12, – М., 1984.
9. Золотарев В.В. Параллельное кодирование в каналах СПД. – В сб.: «Вопросы кибернетики», ВК-120, М., 1986.
10. Золотарев В.В. Реальный энергетический выигрыш кодирования для спутниковых каналов // В кн.: 4-я Международная Конференция. Спутниковая связь. ICSC-2000., Т.2, – М.: МЦНТИ, 2000, С. 20-25.
11. Золотарев В.В., Овчинин Г.В. Использование многопорогового декодера в каскадных схемах // Вестник РГРТА. 2003. Вып. 11. С. 112-115.
12. Золотарев В.В., Овчинин Г.В. Эффективные алгоритмы помехоустойчивого кодирования для цифровых систем связи // Электросвязь. 2003, №9. С.34-37.
13. Золотарев В.В., Овчинин Г.В. Сложность реализации эффективных методов помехоустойчивого кодирования. – 6-я

Международная конференция по цифровой обработке сигналов ЦОС-2004, тезисы докладов, в печати.

14. Зяблов В.В., Коробков Д.Л., Портной С.Л. Высокоскоростная передача сообщений в реальных каналах. – М.: Радио и связь, 1991. 288 с.
15. Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи // Пер. с англ. С.И. Гельфанда под ред. Б.С. Цыбакова – М.: Радио и связь, 1987. 392 с.
16. Красносельский И.Н. Турбокоды: принципы и перспективы // Электросвязь. 2001. №1. С. 17-20.
17. Месси Дж. Пороговое декодирование // Пер. с англ. Ю.Л. Сагаловича под ред. Э.Л. Блоха – М.: Мир, 1966. 208 с.
18. Нейфах А.Э. Сверточные коды для передачи дискретной информации. – М.: Наука, 1979. 222 с.
19. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки // Пер. с англ. под ред. Р.П. Добрушина и С.И. Самойленко. – М.: Мир, 1976. 594 с.
20. Прокис Дж. Цифровая связь /Пер с англ. под ред. Кловского Д.Д. – М.: Радио и связь, 2000. 797 с.
21. Рид И.С., Соломон Г. Полиномиальные коды над некоторыми конечными полями // Кибернетический сборник. – 1983– Вып. 7 – С. 74-79.
22. Самойленко С.И., Давыдов А.А., Золотарев В.В., Третьякова Е.И. Вычислительные сети. – М.: Наука, 1981, 277 с.
23. Склар Б. Цифровая связь. Теоретические основы и практическое применение. – М. 2003.
24. Форни Д. Каскадные коды // Пер. с англ. В.В. Зяблова и О.В. Попова под ред. С.И. Самойленко – М.: Мир, 1970. 208 с.
25. Шенон К.Е. Математическая теория связи // Работы по теории информации и кибернетике. – М., 1963. С. 243-332.
26. Andrews K., Berner J., Chen V., Dolinar S., Pollara F., Stanton V. Turbo-decoder implementation for the deep space network. *IPN Progress Report 42-148*, Feb. 15, 2002.
27. Bahl L.R. and Cocke J. and Jelinek F. and Raviv J., Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate

*IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.

28. Barbulescu S.A., Pietrobon S.S., Interleaver design for turbo codes. *Electronic Letters*, vol. 30, pp. 2108, Dec. 1994.
29. Benedetto S., Divsalar D., Montorsi D., and Pollara F., Serial Concatenation of Interleaved Codes. *IEEE Transactions on Information Theory* 44, May 1998, pp. 909–26.
30. Berrou C. and Glavieux A. and Thitimajshima P., Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes. *in Proceedings of the International Conference on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 1993.
31. Berrou C., Adde P., Angui E., and Faudeil S., A low complexity soft-output viterbi decoder architecture. *in Proceedings of the International Conference on Communications*, pp. 737–740, May 1993.
32. Chase. D. A class of algorithms for decoding block codes with channel measurement information. *IEEE Trans. Inf. Theory*, IT-18, 170-182, 1972.
33. Chi D.T., A new block helical interleaver. *in Proceeding of MILCOM 1992*, (San Diego, CA), pp. 799-804, Oct. 1992.
34. Consultative Committee for Deep Space Data Systems, Telemetry Channel Coding (101.0-B-5 Blue Book), CCSDS: Newport Beach, California, June 2001.
35. Croizer S., Lodge J., Guinand P., Hunt A., Performance of turbo-codes with relative prime and golden interleaving strategies. *in Proceeding Sixth International Mobile Satellite Conference*, (Ottawa), pp. 268-275, June 1999.
36. Divsalar D., Jin H., McEliece R. J., Coding theorems for «turbo-like» codes. *in Proc. 36th Allerton Conf. on Communication, Control, and Computing*. Allerton, Illinois, pp. 201-210, Sept. 1998.
37. Divsalar D., Pollara F., Hybrid Concatenated Codes and Iterative Decoding. *JPL TDA Report PR 42-130*, April–June 1997. pp. 1–23.
38. Dolinar S., Divsalar D., Weight distribution for turbo codes using random and nonrandom permutations. *JPL TDA Progress Report*, pp. 56-65, Aug. 1995.

39. Erfanian J., Pasupathy S., and Gulak G., Reduced complexity symbol detectors with parallel structures for ISI channels. *IEEE Transactions on Communications*, vol. 42, pp. 1661–1671, 1994.
40. European Telecommunications Standards Institute (ETSI), Universal Mobile Telecommunications System (UMTS): Multiplexing and Channel Coding (FDD). *3 GPP TS 125.212 Version 3.4.0* (23 September, 2000).
41. Frenger P., Orten P., Ottosson T., Convolutional Codes with Optimum Distance Spectrum. *Submitted to IEEE Communications Letters*, July 1997.
42. Hagenauer J. and Hoeher P., A viterbi algorithm with soft-decision outputs and its applications. *in IEEE Globecom*, pp. 1680–1686, 1989.
43. Hagenauer J., Offer E., and Papke L., Iterative decoding of binary block and convolutional codes. *IEEE Transactions on Information Theory*, vol. 42, pp. 429–445, March 1996.
44. Hamming R.W. Error detecting and error correcting codes. *Bell System Tech. J.*, 29, 147-160, April 1950.
45. Ho M.S.C., Pietrobon S.S., Giles T., Interleavers for punctured turbo codes. *in Proceedings APCC/ICCS '98*, vol. 2, (Singapore), pp. 520-524, Nov. 1998.
46. <http://www.mtdbest.iki.rssi.ru>.
47. Jin H., Khandekar A., McEliece R., Irregular repeat-accumulate codes. *Proc. 2nd Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 1-8, Sept. 2000.
48. Koch W. and Baier A., Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference. *IEEE Globecom*, pp. 1679–1684, December 1990.
49. Press Release, AHA announces Turbo Product Code Forward Error Correction Technology. Nov. 2, 1998
50. Robertson P., Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes. *IEEE Globecom*, pp. 1298–1303, 1994.
51. Robertson P., Villebrun E. and Höher P., A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain. *in Proceedings of the International Conference on*

*Communications*, (Seattle, United States), pp. 1009–1013, June 1995.

52. Robinson J.P., Bernstein A.J. A Class of Binary Recurrent Codes with Limited Error Propagation. *IEEE Trans.*, vol. IT-13, №1, 1967, pp.106-113.

53. Seghers J., On the Free Distance of TURBO Codes and Related Product Codes. *Final Report*, Diploma Project SS 1995, Number 6613, Swiss Federal Institute of Technology Zurich, Switzerland, August 1995.

54. Small World Communications, Iterative Decoding of Parallel Concatenated Convolutional Codes. Jan. 1999. Application Note Version 1.4.

55. Takeshita O.Y., Collins O.M., Massey P.C., and Costello D.J., A Note on Asymmetric Turbo-Codes. *IEEE Communication Letters* 3, March 1999, pp. 69–71.

56. Takeshita O.Y., Costello D.J., Jr., New classes of algebraic interleavers for turbo-codes. in *Proceeding of IEEE International Symposium on Information Theory*, (MIT, Cambridge, MA USA), p. 419, Aug. 1998.

57. Third Generation Partnership Project 2 (3GPP2), Physical Layer Standard for cdma2000 Spread Spectrum Systems, Release C. *3GPP2 C.S0002-C*, Version 1.0 (May 28, 2002).

58. Townsend R.L., Weldon E.J. Self-Orthogonal Quasi-cyclic Codes. *IEEE Trans.*, IT-13, 1967, pp.183-195.

59. Valenti M.C., Iterative detection and decoding for wireless communications. *Ph.D. Dissertation*, Bradley Dept. of Elect. & Comp. Eng., Virginia Tech, July 1999.

60. Williams D., Turbo Product Code FEC Contribution. *IEEE 802.16.1pc-00/35*, June 19, 2000.

61. Woodard J.P., Hanzo L., Comparative study of turbo decoding techniques: an overview. *IEEE Trans. Veh. Technol.*, vol. 49, pp. 2208-2233, Nov. 2000.

62. Zolotarev V.V., The Multithreshold Decoder Performance in Gaussian Channels. In *Proc.: 7-th International Symposium on Communication Theory and Applications*, held on 13-18 July 2003, St. Martin's College, Ambleside, UK, 7ISCTA'03, pp.18-22.

## Список сокращений

АБГШ	аддитивный белый гауссовский шум
БЧХ	код Боуза-Чоудхури-Хоквингема
ДКБП	дискретный канал без памяти
ДСК	двоичный симметричный канал
ККЧ	код с контролем по четности
МПД	многопороговый декодер (алгоритм декодирования)
ОД	оптимальный декодер
ПД	пороговый декодер
ПЭ	пороговый элемент
РО	размножение ошибок
РС	код Рида-Соломона
СК	сверточный код
СОК	самоортогональный код
ЭВК	энергетический выигрыш кодирования
MAP	maximum a posteriori
RSC	recursive systematic convolutional (рекурсивный систематический сверточный)
qМПД	q-ичный многопороговый декодер
qПЭ	q-ичный пороговый элемент
qСК	q-ичный симметричный канал
SISO	soft in soft out
SOVA	soft-output Viterby algorithm

# Оглавление

Введение .....	2
1. ОБЩИЕ СВЕДЕНИЯ .....	5
1.1. Элементы систем цифровой связи .....	5
1.2. Модели каналов связи .....	6
1.3. Пропускная способность канала связи .....	9
1.4. Помехоустойчивые коды .....	15
1.5. Основные характеристики методов коррекции ошибок .....	25
2. БЛОКОВЫЕ КОДЫ И МЕТОДЫ ИХ ДЕКОДИРОВАНИЯ .....	28
2.1. Коды Хэмминга.....	28
2.2. Коды Боуза-Чоудхури-Хоквингема .....	31
2.3. Коды Рида-Соломона .....	35
2.4. Мажоритарно декодируемые коды .....	37
2.5. Декодер Меггита.....	40
2.6. Мягкое декодирование блоковых кодов.....	43
2.7. Пороговое декодирование.....	45
2.8. Многопороговый декодер .....	52
2.9. Недвоичные многопороговые декодеры.....	58
3. СВЕРТОЧНЫЕ КОДЫ И МЕТОДЫ ИХ ДЕКОДИРОВАНИЯ .....	64
3.1. Алгоритм декодирования Виттерби .....	64
3.2. Последовательные алгоритмы декодирования .....	74
3.3. Пороговый декодер.....	78
3.4. Многопороговый декодер .....	83
4. КАСКАДНЫЕ СХЕМЫ КОДИРОВАНИЯ.....	87
4.1. Каскадные коды, построенные с использованием кода Рида-Соломона .....	88
4.2. Каскадные коды, декодируемые с использованием многопорогового декодера .....	91
4.3. Применение многопорогового декодера в схемах с параллельным кодированием .....	96
4.4. Турбо коды .....	98
4.5. Сравнение сложности реализации эффективных методов декодирования помехоустойчивых кодов .....	113
Заключение.....	116
Список литературы.....	117
Список сокращений .....	122
Оглавление .....	123

## Многопороговые декодеры – современное решение проблемы помехоустойчивого кодирования

ФГУП НИИР предлагает эффективные декодеры пятого поколения на базе самых последних теоретических и экспериментальных исследований метода **многопорогового декодирования (МПД)** блоковых и свёрточных кодов. Впервые на новейших программных и аппаратных средствах достигнуты энергетические выигрыши кодирования 7 дБ и более при линейной, т.е. минимально возможной сложности реализации самого декодера и практически неограниченном быстродействии.

При сопоставимой эффективности кодирования программные версии МПД проще других методов в каналах с большим уровнем шума на 2 порядка.

Реальный выигрыш по скорости декодирования при аппаратной реализации ещё более значителен. Для некоторых параметров систем кодирования МПД может быть почти в 1000 раз быстрее, чем аналогичные по эффективности турбо декодеры.

**НИИР** предлагает **программные декодеры** на базе МПД для компьютеров общего назначения со скоростями более 2 Мбит/с при отношении сигнал/шум канала менее -0,25 дБ.

МПД обеспечивает дополнительный энергетический выигрыш в 3-5 дБ и более по сравнению со стандартным алгоритмом Витерби.

Принимаем заказы на высокоскоростные аппаратные версии МПД.

Дополнительную постоянно обновляемую информацию об МПД алгоритмах можно также найти на специализированном сайте [www.mtdbest.iki.rssi.ru](http://www.mtdbest.iki.rssi.ru).

---

Приглашаем специалистов в области связи на цикл лекций о методах МПД «**Современные методы помехоустойчивого кодирования с характеристиками оптимального декодирования**» ведущего специалиста НИИР д.т.н. В.В. Золотарёва

Предлагаем организовать учёбу специалистов по методам помехоустойчивого кодирования на вашем предприятии.

\*\*\*\*\*

Для контакта:

**НИИРадио: (095)-261-03-27, (095)-261-54-44;**

**e-mail: niir.189@mail.ru, zolotasd@yandex.ru, g\_ovechkin@mail.ru,**

**моб.: 8-916-518-86-28**

## Multithreshold decoders – are up-to-date solution for a noiseproof coding problem

FGUE NIIRadio suggests fifth generation of effective **multithreshold decoders (MTD)** at the latest theoretical and experimental research basis for block and convolutional codes. For the first time a code gain 7 dB and more is reached with linear complexity on the most modern program and hardware means. MTD has minimally possible hardware implementation of the decoder itself and practically unlimited throughout.

At comparable coding efficiency soft MTD versions are simpler than other methods in channels with the big noise level on 2 decimal orders.

The real MTD advantage on decoding rate at hardware realization is even more significant. For some parameters of coding systems MTD can be more fast almost in 1000 times, than similar by efficiency turbo decoders.

**NIIRadio suggests the software decoding systems** on the MTD basis for general purpose computers **with speeds more than 2 Mbit/s** at a signal-to-noise ratio in a channel less than -0,25 dB.

MTD provides additional code gain 3-5 dB and even more with respect to standard Viterbi algorithm.

We are ready for hardware versions of MTD orders.

We are ready to international cooperation.

The full additional permanently upgraded information on MTD algorithms can be also found on our specialized site [www.mtdbest.iki.rssi.ru](http://www.mtdbest.iki.rssi.ru).

---

We invite the specialists in the field of communication to a lecture cycle about MTD methods «**Modern methods of noiseproof coding with performance of total searching algorithms**», which one will be readed by the leading NIIRadio specialist in the noiseproof coding Doctor of Sciences V.V.Zolotarev.

We suggest to organize study of experts on methods of noiseproof coding at your enterprise.

\*\*\*\*\*

**For contact:** In FGUE NIIRadio, Russia, Moscow

tel.: +7 095 261 03 27 and +7 095 261 54 44;

e-mail: [zolotasd@yandex.ru](mailto:zolotasd@yandex.ru), mob.: +7 916 518 86 28, V.V. Zolotarev or  
e-mail: [g\\_ovechkin@mail.ru](mailto:g_ovechkin@mail.ru) G.V. Ovechkin

## **Summary**

In the reference book the methods are submitted systematically and the characteristics of different noiseproof coding methods are described. The capabilities of many coding methods are reviewed, which ones until recently were insufficiently widely submitted in the literature on error correcting codes. The development of technologies in communication engineering lately purely has changed this branch. Therefore many decoding algorithms, unapproachable earlier for implementation, now quite enable creation of new methods of error correction on their fundamentals, which one can be effective enough and low-budget. It allows to consider, that described in the reference book the new decoding methods can be useful in the most different areas of communication, providing thus the so high characteristics of decoding, about which one there was no also speech only 10-15 years ago.