В.В. Золотарев, Г.В. Овечкин

# Основы современных технологий помехоустойчивого кодирования. История. Возможности. Перспективы.

**Аннотация**

В популярной форме описано современное состояние теории и техники помехоустойчивого кодирования. Рассмотрен алгоритм многопорогового декодирования. Дается сопоставление методов повышения достоверности цифровых данных в каналах с большим уровнем шума.

Для широкого круга читателей, может быть полезна студентам и аспирантам институтов связи и инженерам, занимающимся вопросами повышения достоверности передачи данных по каналам спутниковой, космической и иных видов связи с большим уровнем шума.

# 1. Кругом помехи и шумы

Вы включили приемник, чтобы скоротать непогоду наедине с музыкой. Но вместе с мелодией вы слышите треск, свист и завывания эфира. Они мешают звукам симфонии дойти до вас во всем своем великолепии. Это помехи, шум разной природы. Но музыку вы слышите, мелодию улавливаете, потому что музыка содержательна, прогнозируема и узнаваема, хотя отдельные ноты вы можете за помехами и не расслышать. Таким образом, радиопередача дошла о вас.

А что делать, если при наличии больших помех нужно с высокой достоверностью принять именно ту информацию, которую хотел послать вам отправитель, да еще и в цифровом виде? А ошибиться можно в среднем только в одном бите из многих тысяч или даже миллионов. При этом переспрашивать сообщения нельзя или можно, но крайне редко. Этими задачами занимается теория помехоустойчивого кодирования, в обязанности которой, среди прочих сложных проблем, входит разработка методов декодирования (обработки) корректирующих кодов, обеспечивающих по возможности быстрое и точное решение о переданном сообщении по искаженному шумами бледному подобию оригинала, достигшему получателя.

Устройства, реализующие в приемнике эту важную миссию, называются декодерами. Над тем, как их сделать как можно проще, трудятся большие коллективы специалистов по всему миру. История этих разработок полна совершенно необычных историй. О части из них мы расскажем тоже.

В конце нашего путешествия по огромному полю поистине спортивных достижений теории и техники кодирования мы предложим вам несколько простых программных образцов многопорогового декодера и других наиболее эффективных алгоритмов декодирования. Но сначала предлагаем внимательно прочесть эту книгу, которую мы старались сделать интересной и полезной. После ознакомления с основными теоретическими и прикладными вопросами, которые рассмотрены далее, ваше

понимание возможностей программных версий многопорогового декодера (МПД) будет более глубоким.

Хотя базовые идеи алгоритмов типа МПД оказываются на удивление простыми, возможно, что некоторые разделы книги для их более полного понимания потребуется перечитать еще один–два раза. Разумеется, мы надеемся, что интерес некоторых наших читателей к очень простым по реализации и необычным по своей очень высокой корректирующей способности методам позволит расширить сферу применения этих алгоритмов в системах связи.

## 2. Что делает теория кодирования?

Сорокалетний путь теории кодирования изобилует удивительными и неожиданными событиями. Впрочем, и само-то ее рождение вместе с первыми успехами было довольно нетрадиционно. Сначала К. Шеннон указал, что для повышения качества передачи, ее точности не нужно увеличивать энергию передачи, т.е. мощность системы связи. Достаточно только хорошо выбрать сигналы и затем правильно или с допустимой точностью воспринять их и обработать. Инженеры весьма удивились и захотели узнать все о правильном выборе сигналов. Но оказалось, что теория сразу вышла на так называемую теорему существования, сказав, что может быть, а чего не бывает ни при каких условиях. О том же, как выбрать систему сигналов, т.е. наилучшую последовательность битов при передаче по двоичному каналу, например, с независимыми ошибками, сначала было не очень ясно. Кстати, именно этот канал мы и будем далее рассматривать. Поскольку все передаваемые двоичные символы (биты) искажаются в этой модели канала одинаково независимо с вероятностью $p_0$, будем называть его двоичным симметричным каналом (ДСК) без памяти. Он обладает многими достоинствами. Во-первых, простота модели позволяет широко использовать ДСК в различных аналитических и численных оценках. Во-вторых, особенно важно, что эта модель канала хорошо соответствует существующим спутниковым, космическим и ряду других реальных и обычно весьма дорогих каналов связи. Это

помогает весьма точно определять потенциальные возможности таких каналов и характеристики конкретных реализаций систем цифровой связи.

Коды, обеспечивающие коррекцию ошибок, характеризуются избыточностью, которая вводится в передаваемое сообщение. Если для передачи 100 битов информации в сообщение добавляется такое же количество проверочных символов, то можно говорить о 100%-ной избыточности выбранного кода. Однако чаще используется понятие кодовой скорости $R$, равной отношению числа информационных символов кода $k$ к его полной длине $n=k+r$, где $r$ – число избыточных символов.

Пусть далее для только что введенного канала типа ДСК задана вероятность $p_0$ искажения произвольно выбранного символа сообщения $p_0<0{,}5$. Какова допустимая избыточность кода, его кодовая скорость, чтобы передача с высокой достоверностью была в принципе возможна?

Фундаментальное понятие пропускной способности канала $C$, введенное Шенноном, как оказалось, очень просто соотносится со скоростью $R$: всегда должно быть $R<C$! Если это условие выполняется, то канал не перегружается информацией и может доставить получателю не слишком испорченное ошибками сообщение.

Если сообщение достаточно длинное, то доля искаженных символов в нем как раз и будет близка к $p_0$. Для ДСК $p_0$ однозначно определяет $C$:

$$C=1-H(p_0),$$

где $H(x) = -x\log_2 x-(1-x)\log_2(1-x)$ – двоичная энтропия.

На рис. 1 представлена зависимость $C=C(p_0)$, из которой видно, что с ростом $p_0$ $C$ быстро убывает. Например, для $p_0=0{,}11$ $C=0{,}5$, что и определяет возможности кодов с $R=1/2$: они могут работать только в каналах типа ДСК с $p_0<0{,}11$. Но если это условие выполняется, то возможна передача с последующим восстановлением истинного двоичного сообщения со сколь угодно большой достоверностью. Правда, эта радость, гарантированная теорией, не без горького привкуса: чтобы реально обеспечить очень малую вероятность ошибки после сколь угодно сложной обработки, нужно иметь достаточно длинный кодовый блок, про

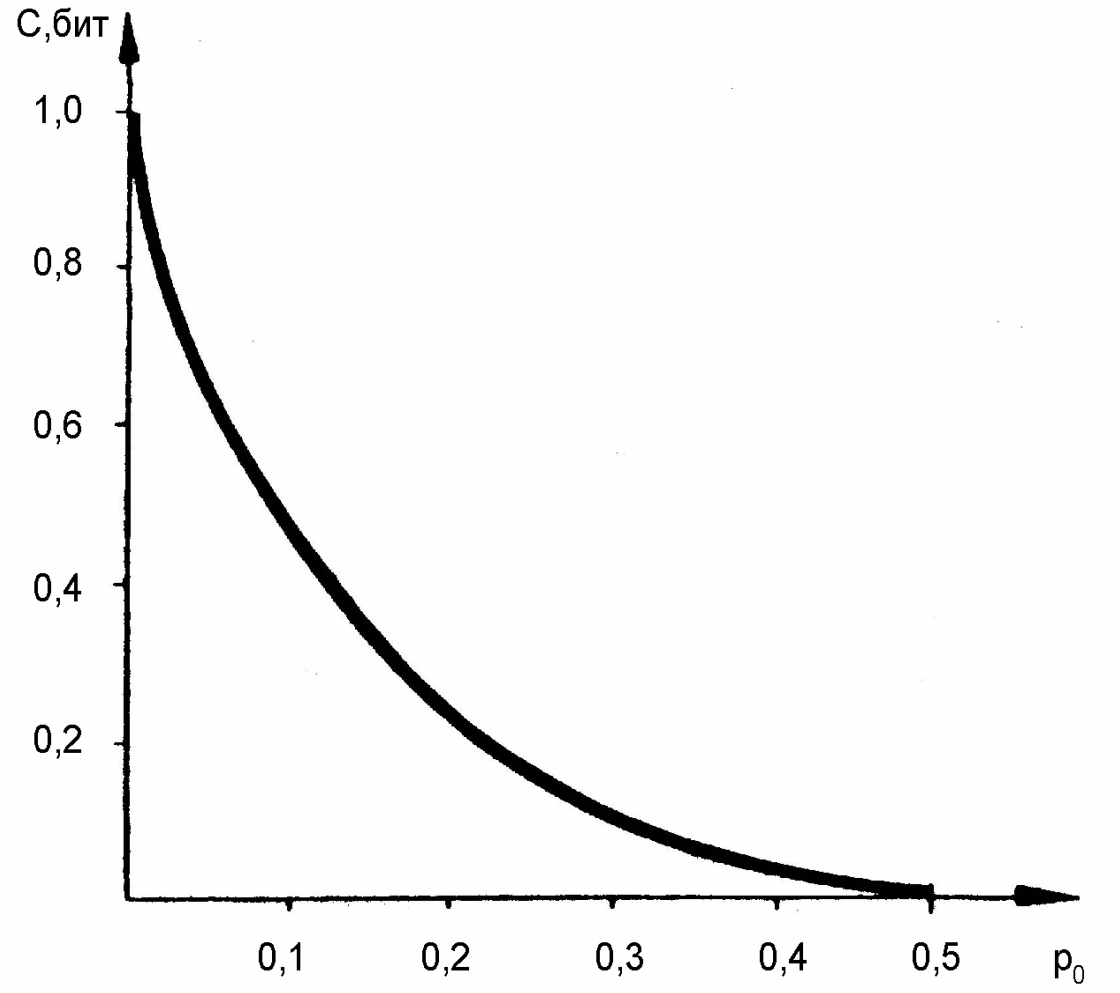


Рис. 1. Зависимость пропускной способности ДСК
от вероятности ошибки канала.

метод выбора которого теория, снабдившая инженера сначала лишь теоремой существования, не очень спешила подсказать что-либо конструктивное и полезное. Одновременно с этим и метод обработки такого блока, как догадывались специалисты, не может быть слишком простым.

Кстати, а для каких длин кодов достижимы заданные достоверности? И, как во многих других науках, сразу приходится вместо использования точных выражений переходить к оценкам, хотя во многих случаях они оказываются довольно простыми и удобными, а также, что уж совсем кстати, достаточно точными для многих приложений.

Давайте попробуем получить такие оценки, лучше которых нельзя сделать ничего.

## 3. Если самый-самый хороший

Для определения соотношений между длиной кодов и достоверностью декодирования введем важнейшее понятие кодового расстояния *d*. Если выбрать двоичный вектор (последовательность) длины *n*, то всего возможно $2^n$ таких векторов, причем для каждого из них есть другой вектор, отличающийся от него только в одном символе.

Конечно, число таких «соседей» у любой последовательности равно *n*. Количество позиций, в которых два двоичных вектора равной длины отличаются, будем называть расстоянием Хемминга между ними. А для некоторого множества таких векторов назовем расстоянием *d* минимум по всем попарным расстояниям по Хеммингу между векторами этого множества. Это и есть кодовое расстояние, если множество – код. Значит, для полного набора двоичных векторов длины *n* в приведенном примере расстояние равно 1.

Хорошо известны последовательности, расстояние в которых равно 2. Для их выбора подсчитаем в последовательностях длины (*n*–1) количество единичек. Если их будет четное число, то *n*-й символ в каждом таком векторе выберем равным 0, а если нечетное, то последний символ пусть будет единицей. Тогда получим код контроля по четности.

Так что же такое код? Множество допустимых сообщений! Если векторы с нечетным весом недопустимы, то вы можете проверить расстояние между словами кода проверки на четность. При нашем методе отбора слов для этого кода (и любом другом) расстояние между ними всегда будет равно двум или больше. Это его кодовое расстояние. И такой код уже может обнаруживать одну ошибку, так как при единственном искажении любого из символов вектора длины *n* сумма по модулю 2 становится равной 1, что и обнаруживает ошибку. Но не исправляет ее!

Для исправления хотя бы одного искажения расстояние должно быть равным 3. Таковы коды Хемминга, широко используемые при защите машинной памяти. Но число дополнительных избыточных символов в коде будет при этом уже суще-

ственно больше за счет дополнительных контрольных разрядов. В коде Хемминга длины $n$, число избыточных символов равно $r=\log_2 n$. Эти коды можно назвать совершенными. Они составляют весьма ничтожную долю от общего числа всех кодов, и обладают замечательным свойством, которое, вообще говоря, абсолютно не типично для произвольно взятого кода: любая точка в $n$-мерном пространстве (а каждому вектору длины $n$, конечно, можно сопоставить точку этого пространства) находится на расстоянии не более $d/2$ хотя бы относительно одного кодового слова.

Иначе говоря, если $d$ нечетно, то любая точка попадает в $n$-мерный шар радиусом $(d–1)/2$ около единственного кодового слова, и в канале типа ДСК наилучшие (т.е. оптимальные!) решения о переданном сообщении должны приниматься именно по этому правилу: считается, что передано ближайшее к принятому вектору кодовое слово. Вы легко можете проверить, что, например, для кода Хемминга с $n=15$ и $r=n–k=4$ проверочными символами при $2^{11}$ кодовых словах и $n$ точках, удаленных от каждого из них на 1, общая сумма всех возможных сообщений длины 15 будет равной как раз $2^{15}$, что и иллюстрирует приведенные рассуждения.

При желании исправлять все ошибки веса $t_0$ кодовое расстояние должно быть хотя бы $d=2t_0+1$. Конечно, лучше выбирать более высокие значения $d$ при заданной избыточности, т.е. величине $R$. Но тогда и обработка вектора, в котором есть ошибки, будет усложняться. Но самое главное, хотя и вполне ожидаемое, обстоятельство состоит в том, что при заданном $R$ существуют коды лишь с ограниченными значениями $d$.

Совершенные коды можно также называть сферически плотно упакованными, так как они буквально все многомерное пространство без остатка делят на «сферы влияния». Эта их особенность и позволила находить полезные оценки для потенциальной помехоустойчивости кодов через оценки для совершенных кодов, которых очень мало, а среди длинных фактически и нет.

Эти чрезвычайно важные для теории кодирования оценки можно получить на основе очень простых рассуждений. Пусть

есть код длины $n$ с некоторым значением кодовой скорости $R=k/n$. Тогда для $2^k$ кодовых слов в сферу каждого из них попадает $2^{n-k}$ точек пространства. А поскольку сфера радиуса $m$ должна содержать $C_n^m$, точек, то все $2^r$ точек будут перечислены, если окажется, что $t_0$ такое максимальное число, что сумма всех $C_n^m$, $0<m<t_0$, меньше чем $2^r$. Это значит, что все $2^r$ выделенные для каждого кодового вектора последовательности уже использованы для формирования сфер радиуса $t_0$ и, значит, нет способа дальнейшего увеличения расстояния между $2^k$ кодовыми словами. Величина $t_0$ и определяла бы корректирующую способность совершенного кода. Для него $d=2t_0+1$, и найти лучшие коды уже нельзя. Тогда вероятность ошибки декодирования при его использовании определяется вероятностями появления ошибок веса более $d/2$ через обычное биномиальное распределение. И хотя совершенных кодов почти нет, оценки, полученные из приведенных соображений, оказываются вполне приемлемыми для расчета помехоустойчивости в больших шумах.

Весьма интересно, что описанный метод получения оценки для $t_0$ фактически является способом вывода границы Хемминга для кодового расстояния, но более реалистично смотрится граница Варшамова-Гилберта: $H(d/n)=1-R$, функцию энтропии для которой мы уже определили выше. При больших $n$ она гарантирует наличие только кодов с кодовым расстоянием фактически вдвое меньшим, чем граница Хемминга для сферической упаковки. Она же определяет и вероятностные корректирующие возможности реально существующих кодов при малом шуме.

Однако мы хотим успешно работать при значительных уровнях шума в канале. Именно поэтому сейчас мы и воспользуемся оценками вероятностей ошибки декодирования через найденные выше оценки для $t_0$.

Для кодов с $R=1/2$ зависимости вероятности ошибки декодирования $P_B(e)$ для блока длины $n$ в ДСК от вероятности $p_0$ представлены для разных $n$ на рис. 2 сплошными линиями, а оценки вероятностей ошибки на бит $P_b(e)$ – пунктирными. Как видим, если выбирать длинные коды, то можно достичь хороших показателей помехоустойчивости при $p_0<0{,}11$. Так в чем же пробле-

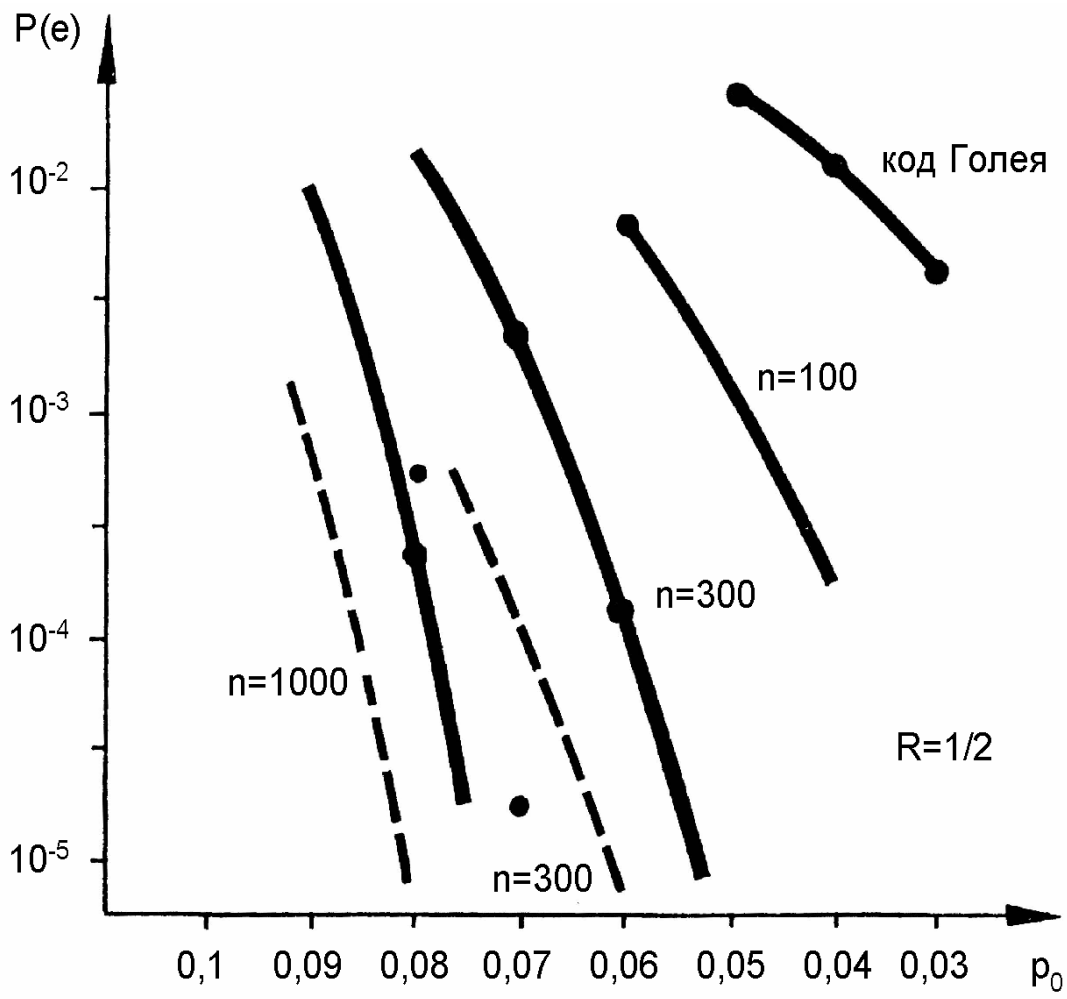


Рис. 2. Нижние оценки вероятностей ошибки блокового кода с $R$=1/2 в ДСК в зависимости от уровня шума канала

ма? Да в том, что хороших методов декодирования для кодов с большим $n$ просто нет. Нужно перебирать $2^r$ или $2^k$ вариантов решений декодера, чтобы выяснить, какое кодовое слово наиболее правдоподобно для данного вектора шума, и если код выбран случайно, то при количестве вариантов всего $2^{150}$ процедура декодирования на самом деле может затянуться весьма надолго. Но из графиков мы видим, что и при $n$~1000 характеристики кодов еще очень далеки от желаемых. Разница с границей $C$ для $R$=1/2, равной $p_0$=0,11 оказывается все еще очень значительной. Число допустимых кодовых комбинаций в этом коде составляет уже $2^{500}$, что равно приблизительно $3\cdot10^{150}$. Это число, кажется, превышает даже количество атомов во Вселенной. Таковы возможности оптимальных декодеров (ОД) с полным перебором. Слабо, да еще с такими сложностями!

А что же можно сделать проще? Вот эта задача и является основной проблемой теории кодирования: построение по воз-

можности более простых методов декодирования, которые были бы не намного хуже переборных ОД. Этим в течение многих уже десятилетий занимаются специалисты, работающие в области теории и прикладных методов помехоустойчивого кодирования.

## 4. Так много способов!

Гораздо проще можно декодировать другими, неперебор-ными способами. Одними из первых в теории кодирования появились коды Боуза-Чоудхури-Хоквингема (БЧХ) и методы их декодирования. Разработанные для них достаточно простые версии алгоритмов декодирования не могут исправить те сообщения, в которых число искаженных символов превышает $d/2$, тогда как другим методам это под силу. Но сложность декодирования этих кодов как число необходимых операций была по порядку величины даже существенно меньшей, чем $n^2$ (сравните с ОД и сложностью $2^k$), что в свое время было очень серьезным достижением.

Во многих случаях можно считать, что число ошибок в информационных символах неправильно декодированного блока близко к $dR/2$, что приводит к вероятности ошибки на бит $P_b(e) \sim dRP_B(e)/(2n)$. На рис. 3 представлены характеристики эффективности кодов БЧХ. Сплошными линиями указаны вероятности $P_B(e)$ для этих кодов. А примеры зависимостей $P_b(e)$ от $p_0$ для этих кодов также показаны пунктирными линиями.

Переход к $P_b(e)$ очень полезен, так как далее будут рассмотрены не только блоковые, но и сверточные коды, которые чаще сравнивают именно по этому параметру. Хотя при анализе и использовании блоковых и сверточных кодов сразу видны некоторые различия, покажем на примере, что между ними есть и значительная общность. На рис. 4,а приведен вариант кодера двоичного блокового кода, а рис. 4,б показывает его преобразование в сверточный кодер.

В блоковом коде 13 информационных битов записываются, например, в регистр, а далее начинается собственно процесс кодирования. Замыкается ключ К1, и регистр сдвига кодера стано-

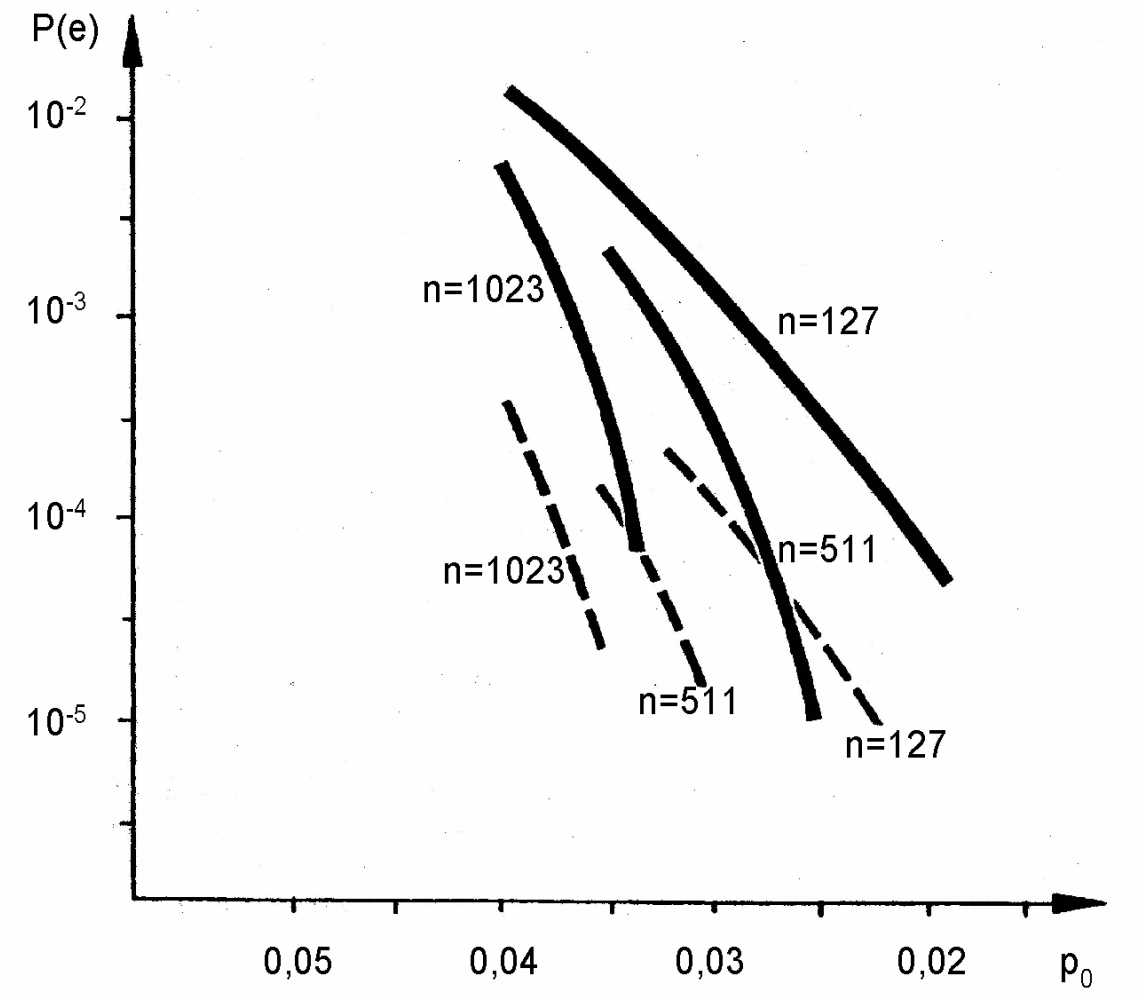


Рис. 3. Вероятности ошибки декодирования блока кода БЧХ в ДСК при $R$=1/2 (сплошные линии) и вероятности ошибки на бит (пунктирные линии)

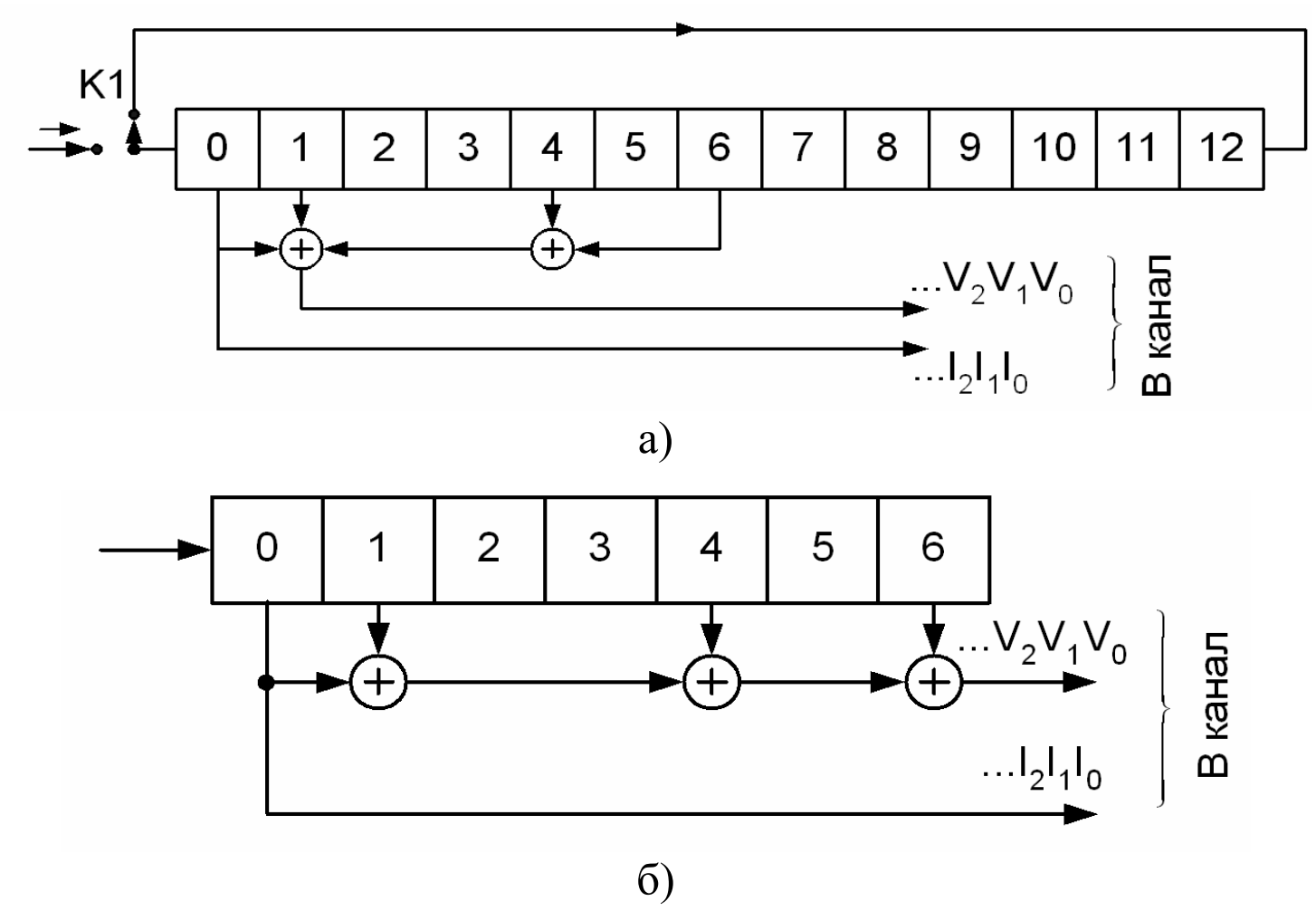


Рис. 4. Кодеры блокового (а) и сверточного (б) кодов с $R$=1/2

вится циклическим. Связи ячеек с многовходовым полусумматором (сумматором по модулю 2) указывают конкретную процедуру формирования проверочных разрядов и, следовательно, сам код.

После замыкания ключа с выхода полусумматора первый проверочный символ поступает в канал, например, с первым же информационным символом. Потом при циклическом сдвиге информационных символов в регистре последовательно получаются все 13 проверочных символов. Таким образом, мы рассмотрели процедуру кодирования ($n$, $k$, $d$)-кодом (26, 13, 5).

Данный код относится к линейным. Это значит, что сумма двух любых кодовых слов $a$ и $b$ также будет кодовым словом $c=a+b$, полагая, что сложение векторов (кодовых слов) производится покомпонентно. Линейность обеспечивает очень малую сложность технической реализации кодера: для многих классов таких кодов для генерации $2^k$ кодовых слов нужен лишь регистр сдвига длины $k$ битов и один или несколько полусумматоров.

Важнейшим свойством линейности является то, что нулевая последовательность всегда будет кодовым словом. Данное очень полезное обстоятельство значительно упрощает многие рассуждения при изучении линейных кодов. Поэтому одним из соображений (пока еще не доказанным) в пользу того, что в рассмотренном коде $d=5$, оказывается тот факт, что вес (число единиц) $w$ любого кодового слова с одной информационной единицей в рассматриваемом примере кода равен $w=5$. Это очевидно, так как в процессе генерации кода помещенная первоначально в любое место информационного регистра единица (при нулях на всех прочих позициях) обязательно проходит за 13 тактов сдвига через все четыре ячейки, с которых подаются сигналы на полусумматор, что приводит к генерации еще четырех единичек в проверочных символах кода. К оценкам кодовых расстояний мы еще будем возвращаться не раз. А пока перейдем к сверточным кодам.

На рис. 4,б представлен кодер сверточного кода также с $R=1/2$ и $d=5$. В кодер, первоначально содержащий только нули, поступает произвольно длинная информационная последовательность по одному биту за каждый такт. Самые правые сим-

волы регистра при сдвиге просто теряются. Поскольку общая длина кода, с учетом избыточности, удваивается, то она становится равной $n_A$=14 и называется длиной кодового ограничения. Показанный пример иллюстрирует связь между блоковыми и сверточными кодами.

Не вдаваясь пока в детали описания алгоритмов, рассмотрим вероятности ошибки декодирования весьма эффективным алгоритмом Витерби (АВ) для широко используемого в технике связи сверточного кода с длиной кодирующего регистра $K$=7 при $R$=l/2 в рассматриваемом нами канале типа ДСК.

Кодовое расстояние указанного кода равно 10 и количество кодовых слов такого веса, а также близких нему, т.е. 12 и более, составляет несколько десятков, что и определяет его помехоустойчивость при малых вероятностях ошибки в канале. Поскольку АВ относится к оптимальным алгоритмам, обеспечивающим минимальную вероятность ошибки решения для используемых кодов, то фактически мы говорим о потенциальной помехоустойчивости этого кода, которая может быть реализована при наилучших способах обработки, в частности, на основе АВ. Однако сложность реализации АВ имеет порядок $2^{K-1}$, что и ограничивает длины тех кодов, для которых он может быть создан.

График зависимости вероятности $P_b(e)$ от $p_0$ представлен для АВ на рис. 5. Существуют также последовательные алгоритмы Фано и стекового типа (ПА), теоретические возможности которых характеризуются тем, что они могут обычно работать только при $R<R_1$, где $R_1$ – вычислительная скорость канала, соответствующая $R_1$=l/2 при $p_0$=0,045. Ясно, что это существенно хуже, чем $p_0$=0,11 для $C$=1/2. Возможности весьма сложного ПА также можно увидеть из рис. 5, который построен в предположении, что память декодера составляет порядка $10^5$ битов для оперативных вычислений. У этих алгоритмов много специфических свойств и недостатков определенного вида, которые несколько ограничивают их возможности.

К ним относятся, например, частые стирания декодируемых блоков данных, которые невозможно декодировать из-за большого числа ошибок, оказавшихся в них, и превышении вследст-

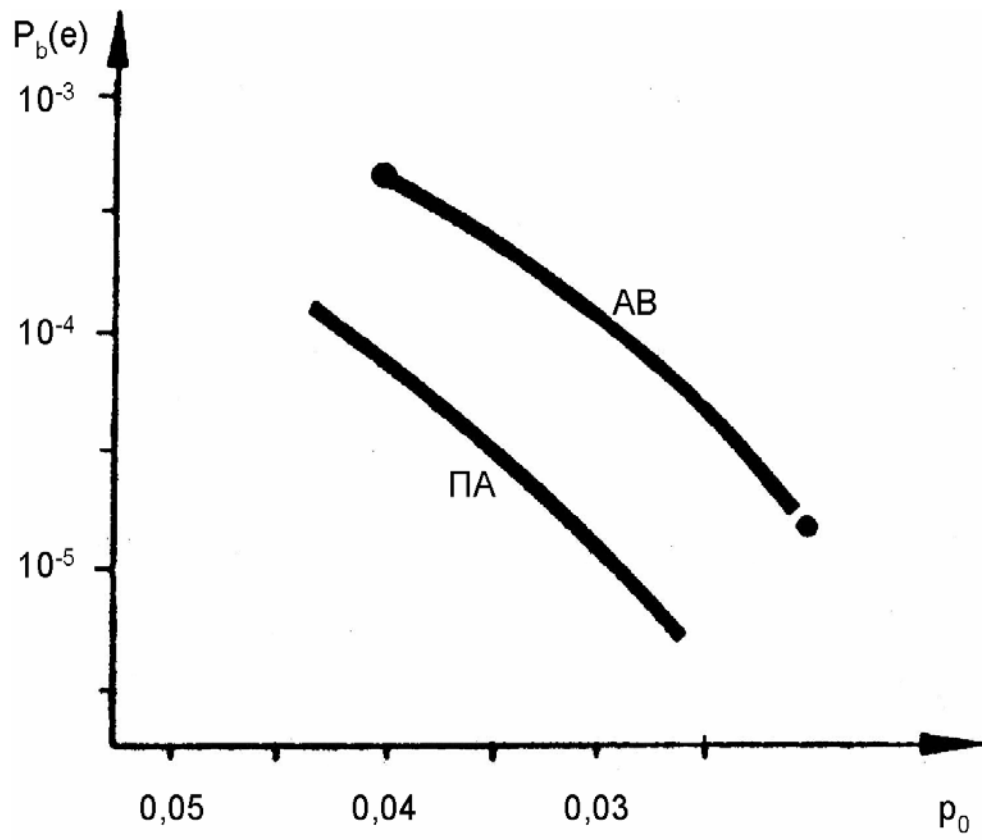


Рис. 5. Характеристики алгоритма Витерби
и последовательных процедур в ДСК

вие этого времени, отпускаемого на декодирование. Кроме того, декодер выдает внешнему потребителю поток декодированных символов в весьма неравномерном темпе, что также связано с существенно неодинаковыми по объему вычислениями при вынесении решений о различных символах сообщения.

## 5. Самый незатейливый

Если для описания декодера кода БЧХ следует изучить некоторые разделы алгебры конечных полей, то наипростейший из известных пороговый декодер (ПД) имеет и самое наглядное описание своей работы. Этот декодер при аппаратной реализации представлен на рис. 6. Мы будем пока говорить о блоковом декодере, но переход к сверточному будет столь же простым, как и при описании кодеров. На полусумматоре А результаты работы кодера из пунктирной рамки (а это проверочные символы) складываются с другими проверочными символами, принятыми из канала связи. Конечно, их сумма по mod 2 всегда даст 0, если в канале не было внесено в сообщение никаких ис-

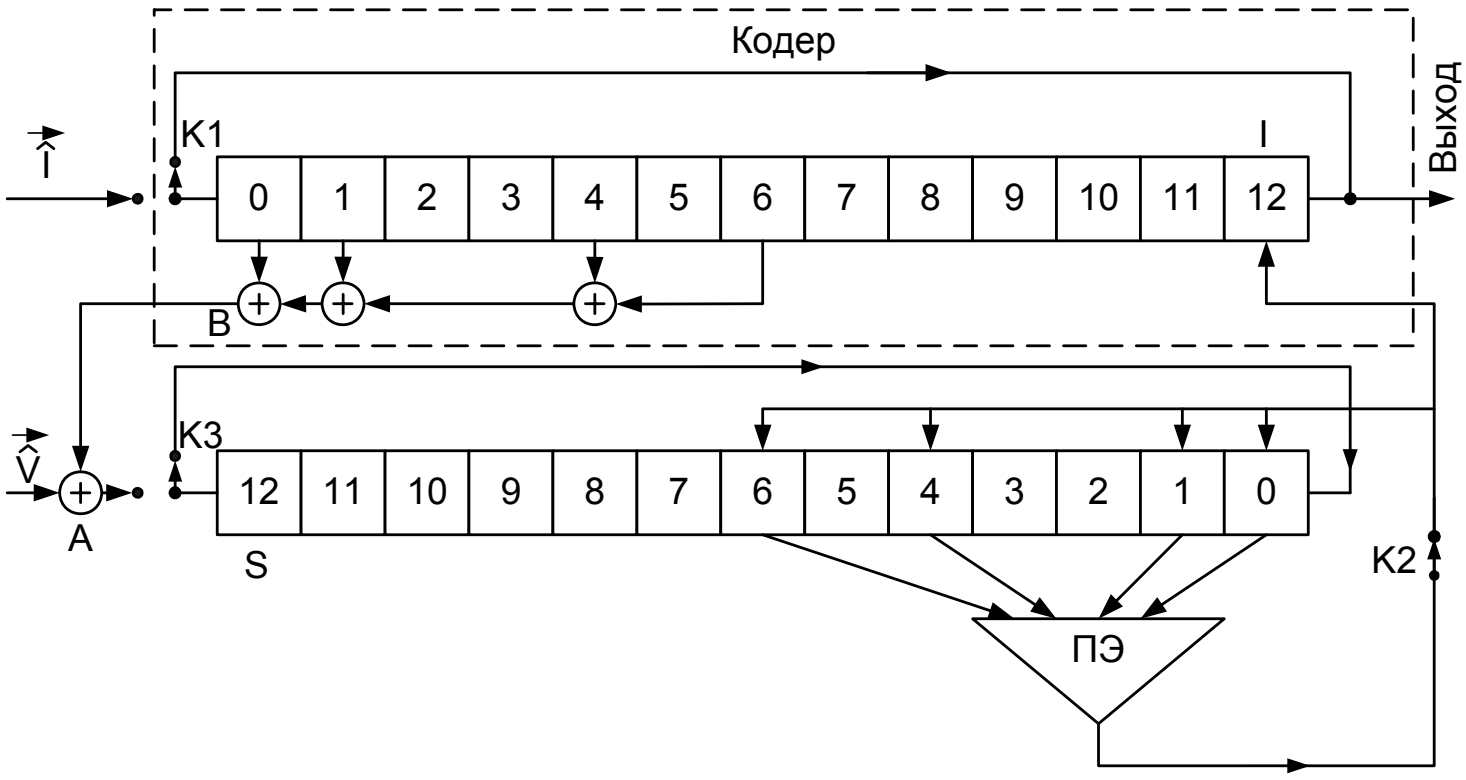


Рис. 6. Схема порогового декодера блокового (26, 13, 5)-кода

кажений. Это обстоятельство хорошо иллюстрирует главное свойство вектора синдрома: его вид полностью определяется вектором ошибки, который исказил при передаче принятое сообщение, и совершенно не связан с информационной частью кодового вектора.

Но если из канала в декодер попали ошибки, то в регистр синдрома *S*, с которым соединен полусумматор А, начнут поступать единички. Причем, единственная ошибка в проверочных символах приведет к появлению одной единички в регистре *S* декодера на месте, соответствующем ошибочному проверочному символу. Будем считать, что при вычислении синдрома пороговый элемент (ПЭ) декодера отключен ключом *K*2 и не мешает выполнению этой процедуры. При выполнении главного второго шага декодирования этот ключ замкнут и ПЭ вырабатывает на своем выходе 1, если число единичек на входе больше двух, а иначе на его выходе будет 0. Ясно, что единственная ошибка в проверочном символе кода приведет к тому, что на любом из 13 сдвигов синдромного (и одновременно, конечно, информационного) регистра сумма единичек на пороге не может быть более одной и пороговый элемент не срабатывает, так как на его выходе сигнал остается равным 0.

Теперь разберем важнейший случай исправления единственной ошибки, происшедшей в информационном символе кода. Пусть при заполнении информационного регистра *I* декодера ошибка канала попала в ячейку 9. Воспользуемся свойством линейности кода, которое мы уже обсудили, и будем считать, что посылалось нулевое кодовое слово. Это облегчает описание декодирования, так как ошибки и единички будут означать при дальнейших рассуждениях одно и то же. А так как вид синдромного регистра (подумайте еще раз, почему это следует из линейности) зависит только от принятого вектора ошибок, дальнейший анализ будет сохранять требуемую общность.

Поскольку процедура кодирования уже подробно разбиралась ранее, внимательный читатель может проверить, что в результате вычисления синдрома, совпадающего просто с кодированием, так как на второй вход полусумматора при этом приходят одни нули, получается заполнение синдромного регистра *S*, соответствующее первой строке табл. 1, где указаны результаты и последующих сдвигов синдрома с учетом решения ПЭ.

Если ПЭ принимает решение «1», то через цепь обратной связи при сдвиге инвертируются именно все те ячейки регистра *S*, с которых поступали сигналы на входе ПЭ. Это обстоятельство и нужно учесть при анализе 10-й и 11-й строк таблицы. Ясно, что только когда ошибка в регистре *I* станет в крайней правой позиции, сумма на ПЭ превысит 2, и поэтому ошибка исправится.

Но очень важно здесь и то, что при попытке декодирования предыдущих символов сумма на ПЭ не превышает 1 ни разу. Более того, если бы мы разомкнули ключ 2, не исправив при этом ошибки, то сумма на ПЭ и относительно последующих символов также не превысила бы 1. А в этом случае оказывается возможным исправление еще одного любого другого информационного символа в принятом блоке, поскольку каждая ошибка канала в данном коде искажает лишь одну проверку.

Значит, сумма на ПЭ не будет более двух на правильных символах и окажется не менее трех на ошибочных, тех, которые нужно корректировать. Это гарантированное исправление $t_0=2$ ошибок и доказывает, наконец, что для рассматриваемых нами в качестве примера кодов $d=2t_0+1=5$.

**Т а б л и ц а 1. Состояния декодера при исправлении одной ошибки в ячейке 9 информационного регистра**

| № такта | Позиция ошибки в информационном регистре | Содержимое регистра синдрома (12 11 10 9 8 7 **6** 5 **4** 3 2 **1** **0**) на ПЭ | Сумма на ПЭ | Решение ПЭ |
|---|---|---|---|---|
| 0 | 9 | 0 0 0 1 0 1 **0** **0** **1** 1 0 **0** **0** | | |
| 1 | 8 | 0 0 1 0 1 0 **0** **1** **1** 0 0 **0** **0** | 1 | 0 |
| 2 | 7 | 0 1 0 1 0 0 **1** **1** **0** 0 0 **0** **0** | 1 | 0 |
| 3 | 6 | 1 0 1 0 0 1 **1** **0** **0** 0 0 **0** **0** | 1 | 0 |
| 4 | 5 | 0 1 0 0 1 1 **0** **0** **0** 0 0 **0** **1** | 1 | 0 |
| 5 | 4 | 1 0 0 1 1 0 **0** **0** **0** 0 0 **1** **0** | 1 | 0 |
| 6 | 3 | 0 0 1 1 0 0 **0** **0** **0** 0 1 **0** **1** | 1 | 0 |
| 7 | 2 | 0 1 1 0 0 0 **0** **0** **0** 1 0 **1** **0** | 1 | 0 |
| 8 | 1 | 1 1 0 0 0 0 **0** **0** **1** 0 1 **0** **0** | 1 | 0 |
| 9 | 0 | 1 0 0 0 0 0 **0** **1** **0** 1 0 **0** **1** | 1 | 0 |
| 10 | 12 | 0 0 0 0 0 0 **1** **0** **1** 0 0 **1** **1** | 4 | 1 |
| 11 | (11) | 0 0 0 0 0 0 **0** **0** **0** 0 0 **0** **0** | 0 | 0 |

Разумеется, все изложенное справедливо лишь для конкретных связей в кодере, т.е. именно для этого кода. Но существуют развитые методы построения кодов с разными $R$ и $d$.

Сверточный ПД для кодера с рис. 4,б представлен на рис. 7. Он также имеет $R$=1/2 и $d$=5. Теми же методами можно показать, что и почти вдвое более короткий сверточный ПД с длиной кода, точнее, длиной кодового ограничения $n_A$=14 исправляет любые $t_0$=2 ошибки. Хороши ли ПД? Не очень. Это расплата за простоту.

Вероятность ошибки на блок $P_B(e)$ для самоортогональных кодов (СОК), к каким относились оба рассмотренных примера,

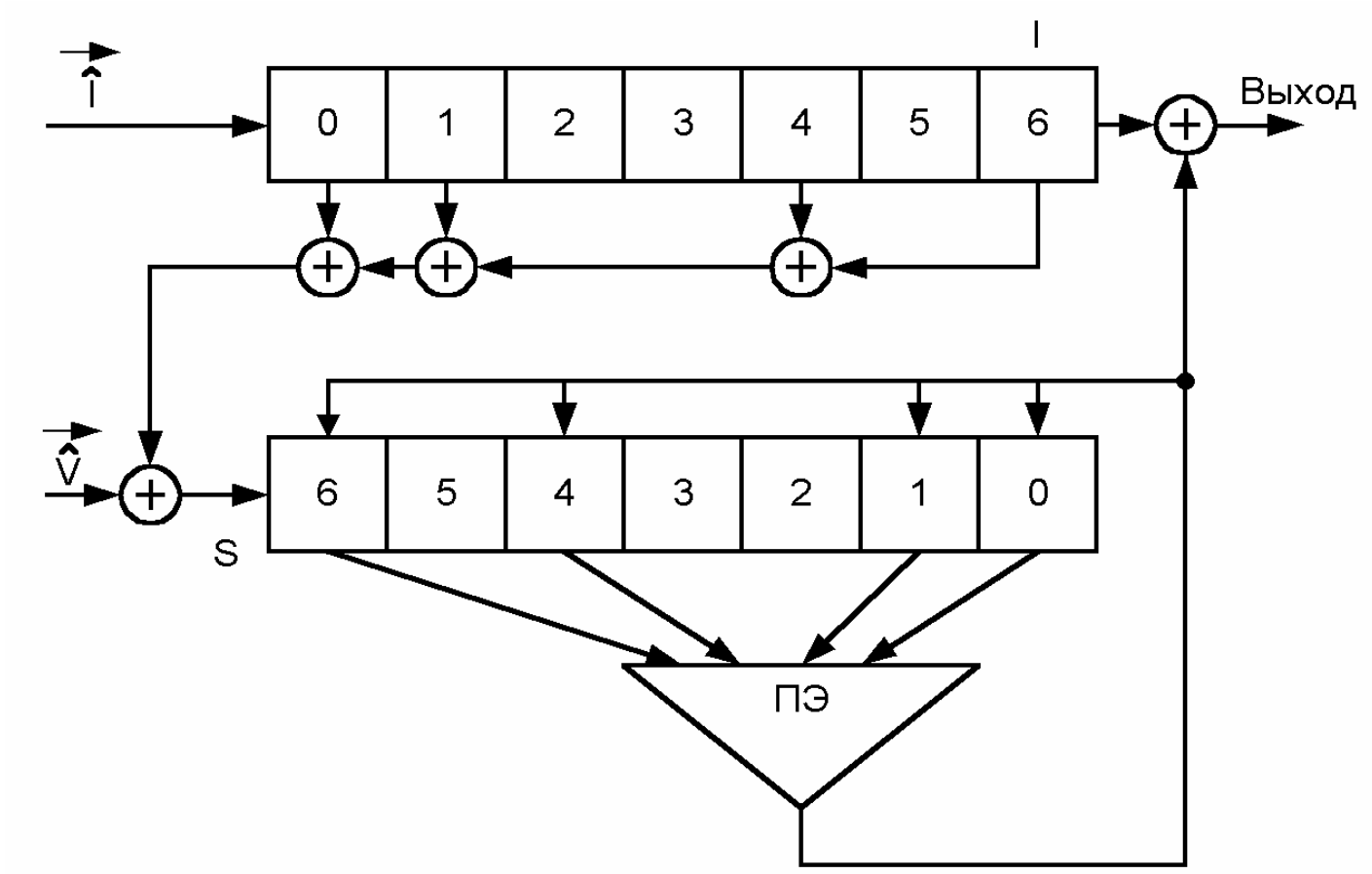


Рис. 7. Декодер сверточного кода с $R$=1/2 и $d$=5

представлена при $R$=l/2 для $d$=11, $n$=182 и $d$=15, $n$=366 кривыми на рис. 8.

Заметим, что результаты весьма скромные. Сопоставление ПД с декодерами кода БЧХ на рис. 3 показывает, что они приблизительно одинаковы, поскольку обеспечивают вероятности $P_b(e)\approx10^{-5}$ при $p_0$=0,01–0,02. Мелкие детали здесь неважны. Для нашего обсуждения пока что существенно только то, что эти вероятности $p_0$, при которых ПД эффективен, много меньше, чем $p_0$=0,11, когда $C$=1/2, и даже меньше, чем $p_0$=0,045, когда $R_1$=1/2. Таким образом, теоретические границы значительно лучше, чем те значения $p_0$, при которых могут результативно работать ПД. Но хорошо, что все же сложность, т.е. число операций в ПД, около $d$ в пересчете на каждый информационный символ. Это, конечно, уже не экспонента.

Давайте все же более внимательно посмотрим, почему так слабы результаты у обычного порогового декодера? Может, мы что-то не доглядели? Ведь при элементарно простом алгоритме ПД имеет еще очень важное свойство – возможность исправлять

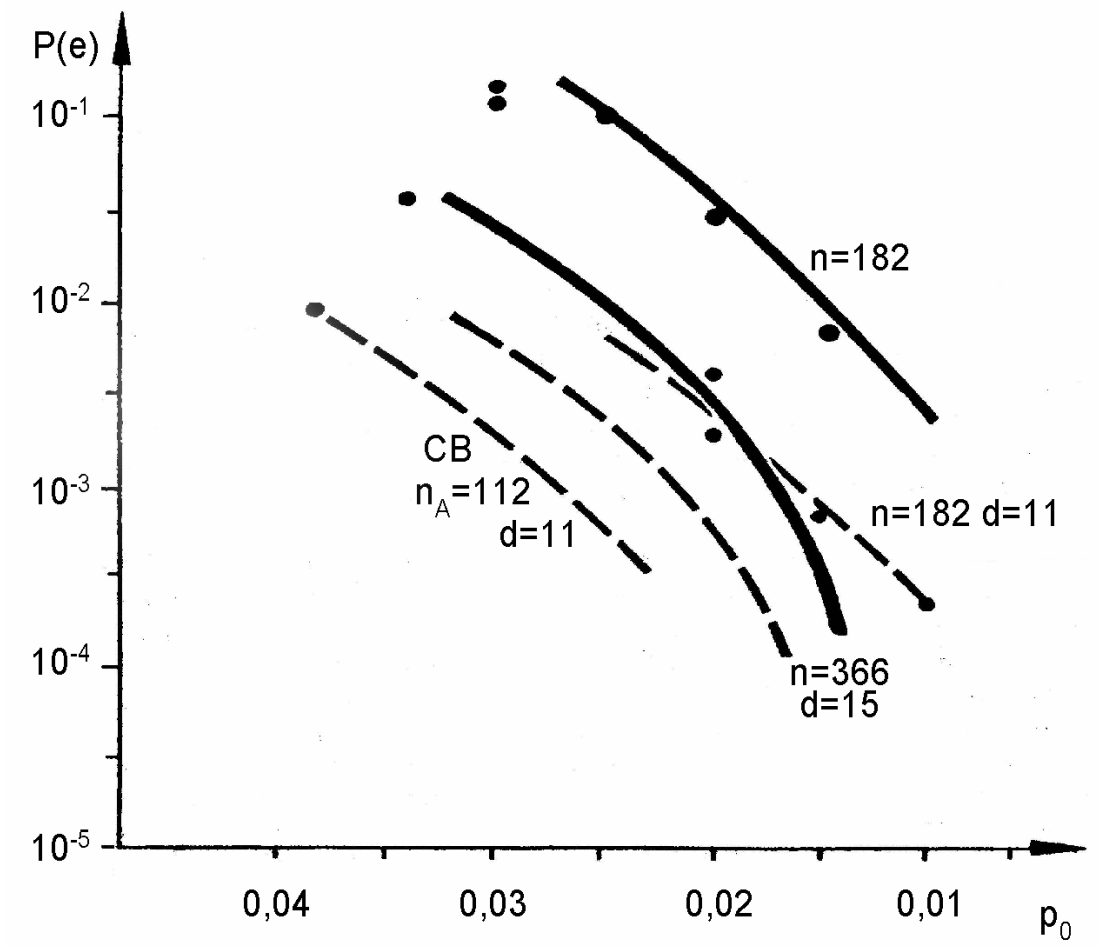


Рис. 8. Характеристики ПД блоковых и сверточных кодов

некоторые (не все!) ошибки большего веса, чем $d/2$. Для большинства других методов это невозможно, а вот ПД делает это, но, действительно, как-то незаметно. Перебирая разные сочетания ошибок даже для простого декодера кода, представленного на рис. 6, можно легко найти такие конфигурации из трех ошибок, которые на самом деле будут исправлены с помощью ПД. (Проверьте: ошибки в информационных позициях 0, 9 и 10 исправляются!) Алгебраические декодеры обычно не могут исправлять такие ошибки, а здесь можно.

И вот здесь возникает этот важный в науке вопрос: почему? А вдруг именно здесь надо хорошо копнуть, и потом найдется повод воскликнуть: «Эврика!». Кстати, рассмотрите, пожалуйста, сочетания ошибок веса 4, конечно, помещая их в синдромный регистр. Кое-что ПД исправит и здесь! А это почему? Может что-то совсем не так просто в теории кодирования?

Так почему же ПД исправляет некоторые ошибки веса более $t_0$, гарантируемого минимальным кодовым расстоянием $d$ используемого кода?

Остальные разделы присутствуют в полной версии книги «Основы современных технологий помехоустойчивого кодирования. История. Возможности. Перспективы».

Информацию по приобретению данной книги можно найти на веб-сайте www.mtdbest.iki.rssi.ru.

## Список литературы


1. Золотарев В.В. Использование помехоустойчивого кодирования в технике связи // Электросвязь. – 1990. – № 7. – С. 7–10.
2. Берлекэмп Э.Р. Техника кодирования с исправлением ошибок // ТИИЭР. – 1980. – Т. 68, № 5. – С. 24–58.
3. Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи: Пер. с англ. под ред. Б.С. Цыбакова. – М.: Радио и связь, 1987. – 392 с.
4. Золотарев В.В. Реальный энергетический выигрыш кодирования для спутниковых каналов // 4-я Междунар. конф. «Спутниковая связь – ICSC-2000». – М.: МЦНТИ, 2000. – Т. 2. – С. 20–25.
5. Месси Дж. Пороговое декодирование: Пер. с англ. под ред. Ю.Л. Сагаловича. – М.: Мир, 1966. – 208 с.
6. Самойленко С.И., Давыдов А.А., Золотарев В.В., Третьякова Е.И. Вычислительные сети. – М.: Наука, 1981.
7. Форни Д. Каскадные коды: Пер. с англ. под ред. С.И. Самойленко. – М.: Мир, 1970. – 208 с.
8. Berrou C., Glavieux A., Thitimajshima P. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes // Proceeding of ICC'93, Geneva, Switzerland, May 1993. – P. 1064–1070.
9. Barbulescu S.A. Iterative decoding of turbo codes and other concatenated codes. Ph.D. dissertation. – Feb. 1996.
10. Andrews K., Berner J., Chen V. at all. Turbo-decoder implementation for the deep space network // IPN Progress Report 42–148. – Feb. 15, 2002.
11. Benedetto S., Montorsi G., Divsalar D., and Pollara F. Serial concatenation of interleaved codes: Performance analysis, design and iterative decoding // JPL TDA Progress Report. – August 1996. – V. 42–126,
12. Höst S., Johannesson R., Zyablov V. A first encounter with binary woven convolutional codes // In Proc. International Symposium on Communication Theory and Applications, Lake District, UK. – July 1997. – P. 13–18.

13. Freudenberger J. Untersuchung von woven-codes. Ph.D. dissertation. – Jan. 1999.
14. Зяблов В.В., Йоханнессон Р., Скопинцев О.Д., Хест С. Асимптотические дистанционные свойства двоичных плетеных сверточных кодов // Проблемы передачи информации. – 1999. – Т. 35, вып. 4. – С. 29–46.
15. Freudenberger J, Shavgulidze S., Zyablov V., Bossert M. Woven codes with outer warp: variations, design and distance properties // Journal on Selected Areas in Communications issue on The Turbo Principle: From Theory to Practice, 2001.

## Содержание

Остальные разделы присутствуют в полной версии книги «Основы современных технологий помехоустойчивого кодирования. История. Возможности. Перспективы».

Информацию по приобретению данной книги можно найти на веб-сайте www.mtdbest.iki.rssi.ru.