УДК 681.32
DOI: 10.15587/1729-4061.2019.157299

Результати дослідження коректувальної здатності ітеративних завадостійких кодів

В. П. Семеренко

Проведено дослідження впливу теорії інформації на розвиток теорії завадостійкого кодування. Показані основні відмінності між ймовірнісним та детермінованим підходами при аналізі коректувальної здатності різних класів лінійних кодів.

Розроблені автоматні ієрархічні моделі для аналізу перестановочного декодування циклічних кодів і запропоновано генератор циклічних перестановок на основі двох автоматів Мура.

На основі автоматного представлення циклічних кодів проведено дослідження регулярних і нерегулярних станів лінійних послідовнісних схем (ЛПС). Показана можливість суттєвого спрощення декодування циклічних кодів на основі переведення нерегулярних синдромів ЛПС в регулярні за допомогою перестановок.

Розроблено формалізовані методи визначення коректувальної здатності циклічних кодів, що ітеративно декодуються (ІДЦК). Традиційний повний перебір всіх можливих варіантів порівняння кодових слів замінено направленим пошуком розв'язання поставленої задачі, що призводить до значної економії часу обчислень. Наведено алгоритм визначення коректувальної здатності ІДЦК відносно подвійних помилок.

Показано, що всі ітеративні коди підвищують свою коректувальну здатність зі збільшенням числа ітерацій і її можна задавати у відсотках для помилок різної кратності. Синдроми помилок розподіляються по окремим ітераціям, що дозволяє зменшити розрядність перевіряльного слова коду. В кінцевому результаті це призводить до збільшення швидкості ітеративних кодів в порівнянні з традиційними коректувальними кодами.

Проведено порівняльний аналіз ІДЦК і LDPC-кодів для визначення сфери їх оптимального застосування

Ключові слова: циклічні коди, низькогустинні коди, коректувальна здатність, ітеративне декодування, лінійна послідовнісна схема, перестановки

1. Введение

Теория помехоустойчивого кодирования прошла в своем развитии сложный и противоречивый путь. Весь этот 70-летний период времени прошел под знаком поиска ответов на главные вопросы: какой код является наилучшим и как его построить?

Не является первоисточником

Эти вопросы интересны не только с позиций математики. Помехоустойчивые коды нашли широчайшее применение в различных технических отраслях: спутниковой и мобильной связи, в системах хранения и архивации данных и т. д. Даже небольшое продвижение в теории может дать огромный экономический выигрыш на практике.

Еще в [1] были обоснованы основные принципы построения помехоустойчивых кодов и предложен критерий наилучшего кода: максимальное приближение к пропускной способности канала передачи данных. Но уже с самого начала стали появляться коды, неплохие с практической стороны, но не соответствующие критерию Шеннона. Возникли несколько направлений в помехоустойчивом кодировании, которые развиваются обособлено и редко взаимодействуют между собой. Поэтому затрудняется совместный анализ и сравнение характеристик различных классов кодов. Особое значение, как теоретическое, так и практическое, имеет сравнительное исследование корректирующей способности помехоустойчивых кодов.

Главной тенденцией в развитии современных систем связи является постоянное увеличение скоростей передачи, практическое освоение терабайтного диапазона. Как и в предыдущие годы, основным резервом в повышении качества и скорости передачи является использование помехоустойчивого кодирования. Современные технологии требуют новых идей, новых способов преобразования данных. Для решения этих задач очень перспективны итеративные помехоустойчивые коды.

2. Анализ литературных источников и постановка проблемы

Теория кодирования основана на теории информации. Кодирование (декодирование) информации можно определить как такое ее преобразование, когда остается неизменным количество информации, но изменяется качественная природа носителей информации [2]. Для выполнения любых преобразований информации ее необходимо представить математически и задать способ ее измерения.

В [3] предложен комбинаторный метод измерения информации на основе выбора из некоторого набора возможностей.

С 1948 года стал широко использоваться термин “бит” для обозначения единицы информации [1]. По Шеннону количество информации равно снятию неопределенности (энтропии) до и после эксперимента. Такой метод требует наличия статистических характеристик отдельных символов и сообщений.

В [4] предложен алгоритмический метод измерения количества информации: относительной сложностью объекта y при заданном объекте x можно рассматривать минимальную длину $l(p)$ “программы” p, которая необходима для преобразования x в y.

Поскольку теория помехоустойчивого кодирования появилась как ответ на потребности систем связи (в первую очередь, космической связи), теория кодирования и стала развиваться в основном на основе вероятностного, т. е. шенноновского подхода в теории информации.

ТОЛЬКО ДЛЯ ЧТЕНИЯ

С позиций практики это был оптимальный выбор, но с позиций теории возникло много вопросов, часть из которых не решены до сих пор. Недостатки вероятностного подхода были отмечены еще в 60-х и 70-х годах: “теория информации должна предшествовать теории вероятностей, а не опираться на неё” [5]. Дальнейшие исследования подтвердили, что ”количество информации не обязательно связано со случайными событиями” [6]. Пробелы в теории скоро проявились на практике.

Основатели теории помехоустойчивого кодирования под термином “наилучшего кода” понимали код, который в максимальной степени соответствовал различным теоретическим границам и соотношениям между параметрами кода и его корректирующей способностью. Однако известные к тому времени коды имели характеристики значительно хуже, чем предсказанные теорией. Поэтому была поставлена цель построения таких кодов, для которых “теоретический интерес зависел от того, насколько реально создание оборудования для их практического использования” [7].

Улучшение одних характеристик кодов, ведет, как правило, к ухудшению других из них [8]. С позиций практики предпочтение следует отдать проблеме повышения способности кодов обнаруживать и исправлять ошибки.

Среди всех помехоустойчивых кодов можно ограничиться только классом линейных кодов, среди которых следует различать вероятностные коды (например, *low-density parity-check* (LDPC)-коды) и детерминированные коды (например, циклические коды). Именно эти коды наиболее распространены сейчас в различных технических сферах.

Основная идея кодирования линейных кодов состоит в добавлении в исходному информационному слову в явном или неявном виде дополнительных проверочных разрядов и получении кодового слова Z. В результате получаем “выигрыш от кодирования”, т. е. возможность выявлять и исправлять ошибки в Z [9]. Оценить аналитически “выигрыш от кодирования” довольно сложно.

Обычно анализ различных кодов осуществляется в сравнении графиков (рис. 1), показывающих зависимость средней вероятности ρ_b ошибочного разряда от соотношения E_b/N_o для этих кодов (E_b – энергия бита, N_0 – спектральная плотность мощности шума). Все кривые должны находиться правее вертикальной ординаты со значением –1,6 дБ, которая обозначает границу Шеннона. С помощью такого графика (кратко его можно назвать графиком BER – *bit error rate*) можно определить расстояние от кодовой кривой до границы Шеннона, и чем оно ближе, тем лучшим считается код. Один код будет лучше другого пропорционально величины γ_e.

Кроме такой вероятностной оценки, имеется и целочисленная характеристика способности обнаружения и исправления ошибок – минимальное кодовое расстояние $d_{\min}$. По определению, $d_{\min}$ равна наименьшему из всех расстояний по Хэммингу

Не является переизданием

между различными парами кодовых слов. Параметр $d_{\min}$ позволяет точно определить число обнаруживаемых τ_d и исправляемых τ_c ошибок заданным кодом:

$$d_{\min} \geq \tau_d + \tau_c + 1 = 2\tau_c + 1 = \tau_d + 1.$$

Универсальный способ точного вычисления параметра $d_{\min}$ для произвольного блокового линейного кода основан на анализе спектра весов кода. Для некоторых подклассов циклических кодов известно распределения весов кода, даже в аналитическом виде. Однако для всех кодов эта задача до сих пор не решена, поскольку она принадлежит к NP-сложным задачам [10].

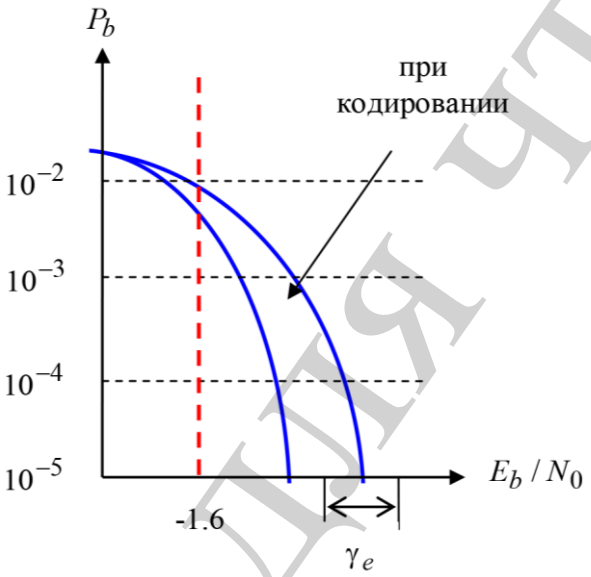

Рис. 1. График зависимости ρ_b от отношения E_b/N_0

Сложность вычисления точных значений $d_{\min}$ способствовало появления различных асимптотических границ, устанавливающих взаимосвязь между $d_{\min}$ и избыточностью кода. Границы Хэмминга, Плоткина и Элайеса устанавливает верхнюю границу для $d_{\min}$, а граница Варшамова-Гильберта – нижнюю границу для $d_{\min}$ [7].

Если для вероятностных кодов более приемлемой является использование графика BER, то для детерминированных кодов более понятным является минимальное кодовое расстояние и соответствующие ему асимптотические границы.

И здесь появляется серьезная проблема: как сравнить корректирующую способность столь разных кодов. Можно попытаться выполнить эту задачу, используя оба критерия для всех кодов.

Например, на одном графике BER можно изобразить вместе как LDPC-коды, так и циклические коды. Однако такой подход будет корректным только для случайных ошибок произвольной кратности. Циклические коды могут также испра-

ТОЛЬКО ДЛЯ ЧТЕНИЯ

лять и пакеты ошибок различной длины. Как было показано в [11], для всех кодов Файра расстояние до границы Шеннона будет одинаковым, т. е. не получим полной оценки по исправлению ошибок циклическими кодами.

Рассмотрим противоположный подход: оценка корректирующей способности вероятностных кодов с помощью параметра $d_{\min}$. За последнее десятилетие появилось огромное количество исследований по этому вопросу. Обобщая полученные результаты относительно LDPC-кодов можно сделать такие выводы:

1. Минимальное кодовое расстояние $d_{\min}$ по-прежнему рассчитывается на основе распределения весов [12].

2. Поэтому ничего удивительного в том, что вычисление $d_{\min}$ для LDPC-кодов является NP-сложной задачей [13].

3. Значения параметра $d_{\min}$ для конкретных LDPC-кодов получены в результате численных экспериментов на суперкомпьютере [14]. Отсутствие математически обоснованных методов оценки корректирующей способности кодов компенсируется переборными алгоритмами, требующих огромных вычислительных ресурсов.

4. Минимальное расстояние алгебраических кодов LDPC обычно выше чем случайных LDPC-кодов [15]. Минимальное расстояние регулярных LDPC-кодов приближается к расстоянию лучших случайных линейных кодов [16]. Следовательно, минимальное расстояние классических (т. е. нерегулярных) LDPC-кодов является низким. С другой стороны, никто не сомневается в высокой корректирующей способности LDPC-кодов, иначе эти коды не применялись бы на практике.

Разрешением этого парадокса может быть лишь признание того факта, что традиционный параметр $d_{\min}$ не является адекватной оценкой корректирующей способности для вероятностных итеративных кодов. Соответственно, для оценки аналогичных свойств детерминированных кодов не самым лучшим критерием будет анализ расстояния до границы Шеннона на графиках BER.

Следовательно, в настоящее время отсутствует общая методика математически обоснованного сравнения корректирующей способности различных классов кодов.

3. Цель и задачи исследований

Целью данной работы является получение результатов сравнительного анализа корректирующей способности линейных помехоустойчивых кодов и получение детерминированных и точных оценок корректирующей способности итеративно декодируемых циклических кодов (ИДЦК) на основе математического аппарата линейных последовательностных схем (ЛПС).

Для достижения поставленной цели необходимо решить следующие задачи:

– исследовать способности обнаружения и исправления ошибок итеративными помехоустойчивыми кодами с позиций теории информации и теории кодирования;

– дать математическое обоснование перестановочного декодирования циклических кодов;

Не является переизданием

– исследовать влияние циклических перестановок на сложность декодирования и на корректирующую способность циклических кодов;

– предложить эффективные для практической реализации средства оценки корректирующей способности ИДЦК.

4. Математическое обоснование перестановочного декодирования циклических кодов

Будем использовать следующие циклические перестановки разрядов кодового слова Z [9, 11]:

$$i \to (i + \upsilon) \bmod n, \quad GF(2) \;\; \upsilon = 2, 3, 4, \ldots \tag{1}$$

Формирование перестановок можно рассматривать как результат работы некоторого генератора перестановок и представить его функционирование конечным автоматом Мура Λ с конечным множеством состояний S, конечным множеством выходов Y, функцией переходов

$$S(t+1) = P \times S(t), \quad GF(n)$$

и функцией выходов

$$Y(t) = S(t), \quad GF(n),$$

где P – оператор перестановок.

Функционирование автомата Λ происходит в дискретные такты t времени. Будем именовать автомат Λ автоматом высокого уровня.

На входы автомата поступают только нулевые значения, а выходами автомата является его состояние. На такте t состояние

$$S(t)=\{s(1), s(2), s(3),\ldots, s(n)\}$$

представляет собой значение позиций всех разрядов кодового слова Z (n, k)-кода.

Для циклического (n, k)-кода элементы множеств S и Y выбираются из целочисленного алфавита $\{0, 1,\ldots , n–1\}$, поэтому можно считать, что вычисления происходят в поле Галуа $GF(n)$.

В начале функционирования автомата Λ состояние $S(1)$ совпадает с исходным значением кодового слова Z. Задачей генератора на основе автомата Λ является вычисление на каждом такте новых (переставленных) позиций разрядов кодового слова Z, что эквивалентно получению следующего состояния $S(t+1)$. Получение очередного состояния автомата Λ будем именовать итерацией.

ТОЛЬКО ДЛЯ ЧТЕНИЯ

Значение позиции первого разряда состояния $S(t+1)$ всегда неизменно: $s(1)=1$. Значения позиций последующих разрядов вычисляются согласно (1). Поскольку эти вычисления осуществляются рекурсивно, поэтому их удобно представить с помощью еще одного автомата Мура (будем именовать его автоматом π низкого уровня) с функциями переходов и выходов:

$$s(i+1) = s(i) + \upsilon, \quad GF(n),$$

$$y(i) = s(i), \quad GF(n).$$

В этом автомате значение позиции $s(i+1)$ зависит как от значения предыдущей позиции $s(i)$, так и от целочисленной константы υ, которая не изменяется на протяжении всего сеанса работы автомата Λ.

На основе автомата π удобно исследовать свойства последовательностей позиций $s(i)$, т. е. циклов перестановок низкого уровня. Рассмотрим граф переходов автомата π для $n=15$ и разных значений константы υ (рис. 2).

При $i \rightarrow (i+1) \bmod n$ задается исходная циклическая последовательность позиций $s(i)$, которая совпадает с исходным значением кодового слова Z. При $i \rightarrow (i+2) \bmod n$ получается новая циклическая последовательность позиций $s(i)$ длины $n=15$. При $i \rightarrow (i+3) \bmod n$ получаются три отдельных циклических последовательностей позиций $s(i)$ длины $n=5$ каждая.

Структура циклов перестановок низкого уровня зависит от соотношений параметров n и υ. Можно отметить главные закономерности:

– при отсутствии общих кратных параметров n и υ получаются циклы максимальной длины, т. е. длины $L=n$.

– при наличии общего кратного m параметров n и υ формируется m циклов длины $L = n/m$.

В итоге на основе одного или нескольких циклов перестановок низкого уровня будет сформировано очередное состояние $S(t)$ автомата Λ. Получение w различных состояний $S(t)$ будет означать возможность w итераций для генераторов перестановок на основе автомата Λ.

Не является переизданием

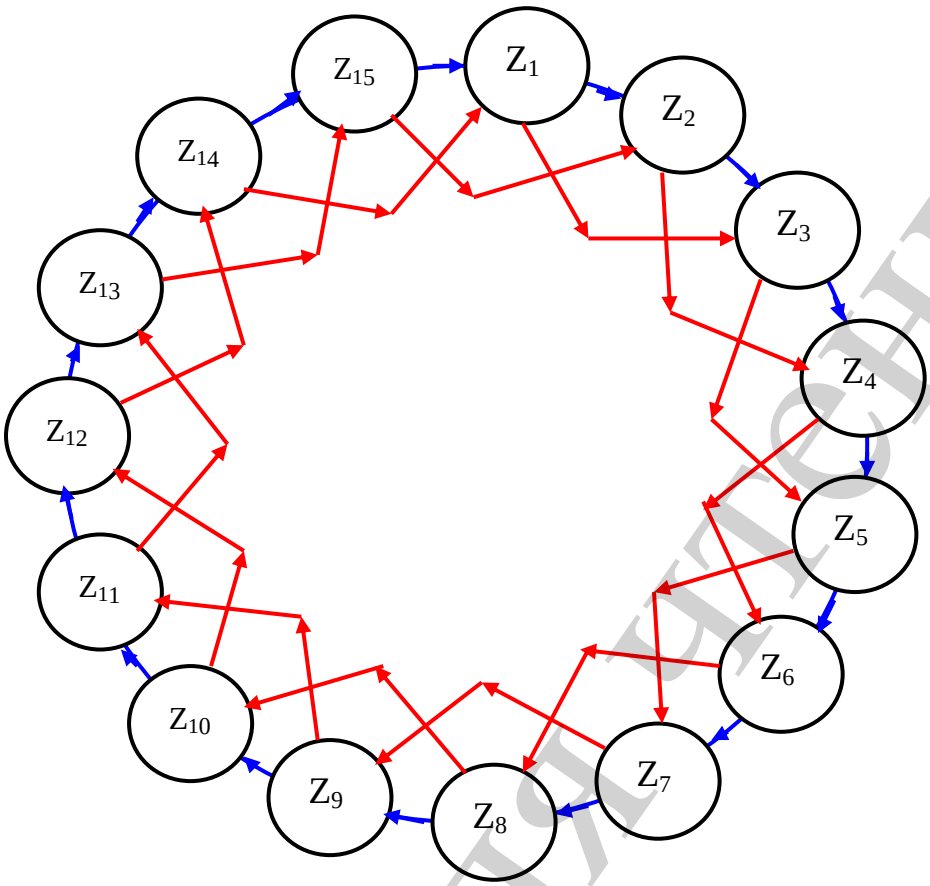

Рис. 2. Циклические перестановки вида $i \rightarrow (i+\upsilon) \bmod 15$ для $\upsilon=1$ (синий цвет) и $\upsilon=2$ (красный цвет)

В табл. 1 приведены параметры генератора перестановок для $n=15$ и возможных значений константы υ.

Как показывает компьютерное моделирование, это свойство генератора возможно только при $L=n$, в большинстве случаем это обеспечивает перестановка вида $i \rightarrow (i+2) \bmod n$.

Иногда встречаются пары разрядов (именуемые маятниковыми), которые только взаимно меняются местами, либо постоянно находятся на одном месте. После перестановок всех остальных разрядов продолжать итерации специально для маятниковых разрядов нет смысла.

Такой эффект, когда при декодировании часть ошибок не поддается коррекции, можно назвать порогом коррекции. Такая ситуация характерна для всех итеративных кодов (для LDPC-кодов и турбо-кодов она известна под термином "*error floor*" [17]). В отличии от вероятностных кодов, в ИДЦК известно точное условие появления порога коррекции.

При перестановках с параметром $\upsilon=2$ маятниковые разряды существуют только для (n, k)-кодов с нечетной длиной n, кратной трем. Их позиции в кодовом слове следующие:

ТОЛЬКО ДЛЯ ЧТЕНИЯ

$n/3+1$ и $2n/3+1$.

Для маятниковых разрядов необходимо дополнительно генерировать проверяющие наборы.

Таблица 1
Возможные параметры генератора перестановок

n	υ=2	w	L
15	2,7,8,13	4	15
15	4,11,14	2	15
15	3,6,9,12	3	5
15	5,10	5	3

Определить количество итераций для заданного циклического кода очень просто: если из начального состояния $S(t)$ на $(w+1)$-м такте работы рассмотренный ранее автомат Λ перешел снова в состояние $S(t)$, следовательно, такой генератор имеет w итераций перестановок. Лучшим будет генератор с наибольшим количеством итераций, поскольку это позволяет исправить больше ошибок.

5. Влияние перестановок на сложность декодирования традиционных циклических кодов

Рассмотрим более подробно влияние перестановок на сложность декодирования циклических кодов.

Задачу декодирования циклических кодов стали считать решенной после появления в 60-х годах различных алгебраических методов декодирования, в частности, метода Берлекэмпа-Месси [18]. Здесь следует уточнить: задача действительно решена, но лишь теоретически. С увеличением числа ошибок и длины кодового слова сложность вычислений стремительно возрастает. Поэтому и появились различные вероятностные методы декодирования линейных кодов [9]. Однако, в итоге сложность вычислений еще более возросла (главным образом из-за использования нецелочисленной арифметики), и при этом исчезли гарантии получения точных решений.

Поэтому задача нахождения точных решений при приемлемой сложности вычислений по-прежнему остается важнейшей нерешенной проблемой в теории помехоустойчивого кодирования.

Перспективным направлением в этом плане является использование автоматного представления циклических кодов. Будем использовать еще одну автоматную модель [19], которая базируется на специальном типе линейных конечных автоматов – линейных последовательностных схемах (ЛПС). ЛПС с r элементами памяти, l входами и m выходами над полем Галуа *GF(2)* описывается функцией состояний (переходов)

Не является переизданием

$$S(t+1) = A \times S(t) + B \times U(t), \quad GF(2) \qquad (2)$$

и функцией выходов

$$Y(t) = C \times S(t) + D \times U(t), \quad GF(2),$$

где $A = \left|a_{ij}\right|_{r \times r}$, $B = \left|b_{ij}\right|_{r \times l}$, $C = \left|c_{ij}\right|_{m \times r}$ и $D = \left|d_{ij}\right|_{m \times l}$ – характеристические матрицы ЛПС; $S(t) = \left|s_i\right|_r$ – слово состояния; $U(t) = \left|u_i\right|_l$ – входное слово; $Y(t) = \left|y_i\right|_m$ – выходное слово.

Процесс декодирования циклических (n, k)-кодов на основе автоматных моделей состоит из двух этапов:

– определения факта наличия или отсутствия ошибки;

– определение параметров ошибки при ее наличии.

Первый этап состоит в вычислении состояния $S(n)$, в которое перейдет ЛПС после подачи на ее вход n-разрядного кодового слова Z по рекурсивной формуле, следующей из (2):

$$S(j+1) = A \times S(j) + B \times z_j, \quad GF(2), \; z_j \in Z, \; j = 1 \div n.$$

Состояние $S(n)$ принято именовать синдромом: нулевое значение этого состояния свидетельствует об отсутствии ошибок в переданном кодовом слове в пределах обнаруживающей способности выбранного циклического кода. При наличии ошибки кратности τ в кодовом слове, которое обозначим как $Z_{err}^{(\tau)}$, будет получен ненулевой синдром ошибки $S_{err}^{(\tau)}()$.

Особенностью циклических кодов является то, что после n циклических сдвигов слова $Z_{err}^{(\tau)}$ получаем n сдвигов j-й конфигурации ошибки $E_j^{(\tau)}$ кратности τ. Каждому сдвигу $E_j^{(\tau)}$ соответствует j-й синдром ошибки $S_{err}^{(\tau)}()$. Всем n сдвигам $E_j^{(\tau)}$ соответствуют n синдромов $S_{err}^{(\tau)}()$, которые образуют нулевой цикл (НЦ) для ошибок кратности τ [20].

Для исправления ошибок в принятом кодовом слове $Z_{err}^{(\tau)}$ достаточно выбрать в каждом НЦ лишь по одному синдрому. Предпочтение следует отдавать регулярному синдрому $S_{err}^{(\tau)}()$, который представляет собой r-разрядное циклическое слово, содержащее τ единиц и (r–τ) нулей, причем с единицей в младшем (левом) разряде (r=n–k) [21].

Регулярный синдром (содержащий его НЦ будем также именовать регулярным) соответствует такой конфигурации ошибочных разрядов в $Z_{err}^{(\tau)}$, когда все

ТОЛЬКО ДЛЯ ЧТЕНИЯ

они попадают в *r*-разрядное проверочное окно $X_{err}^{(\tau)}$ ($X_{err}^{(\tau)} \subset Z_{err}^{(\tau)}$). Для получения разрядов проверочного окна $X^{(\tau)}$ без ошибок достаточно выполнить операцию

$$X^{(\tau)} = X_{err}^{(\tau)} + S_{err}^{\tau}(n), \quad GF(2).$$

Нерегулярному синдрому $S_{err}^{(\tau)}()$ (содержащий его НЦ будем также именовать нерегулярным) соответствует такая конфигурация ошибочных разрядов, когда они расположены по всей длине слова $Z_{err}^{(\tau)}$ и вследствие сдвига никогда не попадают в *r*-разрядное проверочное окно. Для таких ошибок необходимо либо хранить громоздкую таблицу соответствия между $S_{err}^{(\tau)}()$ и $Z_{err}^{(\tau)}$, либо выполнять сложные алгебраические преобразования.

Для циклического (*n*, *k*)-кода при наличии τ ошибок количество регулярных НЦ определяется по формуле:

$$N_{reg}^{(\tau)} = \binom{n-k-1}{\tau-1}, \tag{3}$$

а нерегулярных НЦ:

$$N_{not}^{(\tau)} = \left\lfloor \binom{n}{\tau}\frac{1}{n} \right\rfloor - \binom{n-k-1}{\tau-1}, \tag{4}$$

где $\binom{j}{i}$ – число комбинаций с *j* по *i*.

Из (3) и (4) следует, что в пределах корректирующей способности (*n*,*k*)-кода будут получены только регулярные синдромы ошибок, если выполняется неравенство

$$\binom{n-k-1}{\tau-1} \geq \left\lfloor \binom{n}{\tau}\frac{1}{n} \right\rfloor \text{ для } \tau = 1...\tau_{\min}.$$

Например, (15,7)-код БЧХ, исправляющий две ошибки, отвечает этому требованию и имеет все регулярные НЦ для τ=2 и, поэтому, легко декодируется. А вот уже квадратично-вычетный (17,9)-код с такой же корректирующей способностью ($d_{\min}$=5) имеет один нерегулярный НЦ для τ=2.

В общем случае, с увеличением длины *n* кода и числа исправляемых ошибок τ увеличивается число нерегулярных НЦ и, соответственно, возрастает сложность исправления таких ошибок.

Не является переизданием

И здесь на помощь приходит перестановочное декодирование. Под влиянием циклических перестановок происходит преобразование нерегулярных синдромов в регулярные. Вследствие перемещения разрядов слова $Z_{err}^{(\tau)}$ на каждой итерации большее число конфигураций ошибок попадают в проверочное окно и исправляются.

На рис. 3–5 показаны графики постепенного увеличения регулярных НЦ при увеличении итераций для нескольких кодов.

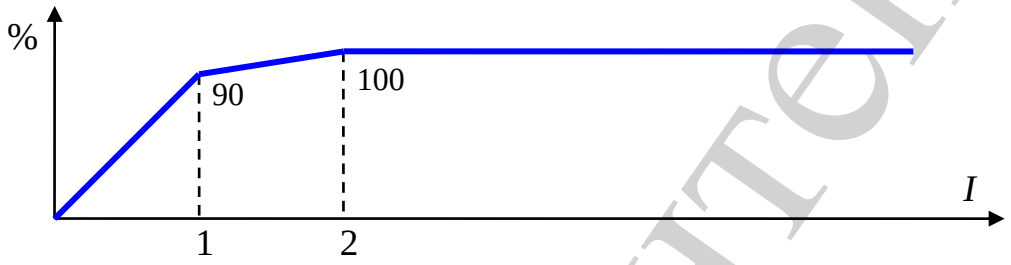

Рис. 3. Доля (в %) регулярных синдромов ошибок по итерациям для квадратично-вычетного (17, 9)-кода, кратность ошибок τ=2

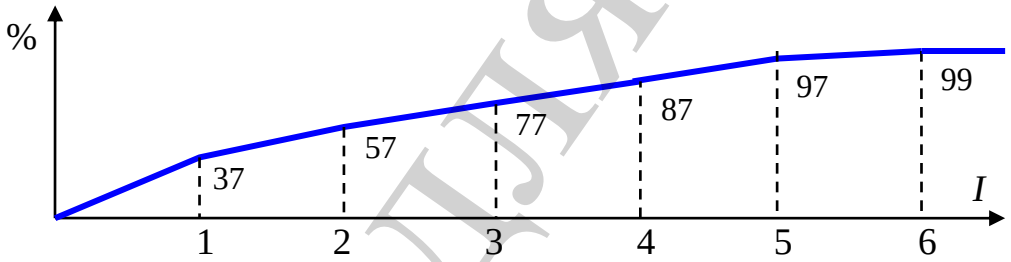

Рис. 4. Доля (в %) регулярных синдромов ошибок по итерациям для (63, 51)-кода БЧХ, кратность ошибок τ=2

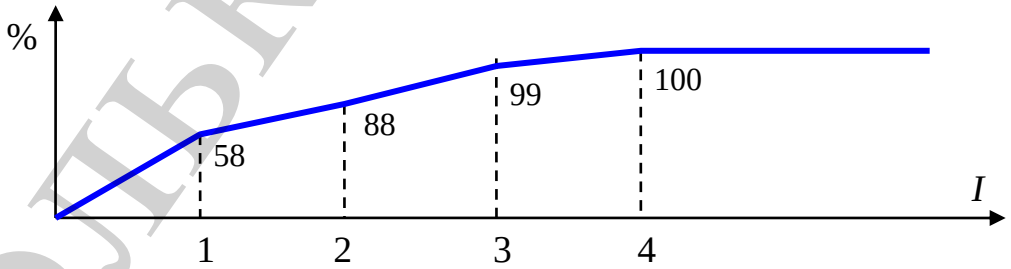

Рис. 5. Доля (в %) регулярных синдромов ошибок по итерациям для (23,12)-кода Голея, кратность ошибок τ=3

Только для чтения

Пример 1. В известном (23,12)-м коде Голея с порождающим полиномом

$$g(x)=1+x^2+x^4+x^5+x^6+x^{10}+x^{11}$$

всем одиночным , 91 % двойным и 58 % тройным ошибкам соответствуют регулярные НЦ. Не рассматривая процедуру декодирования, которая подробно изложена в [21], проанализируем лишь влияние перестановок на сложность выявление ошибок. Пусть по каналу связи было получено кодовое слово с тремя ошибками (отмечены красным цветом):

$$Z_{err}^{(3)}=1\,0\,0\,0\,0\,0\,0\,0\,\mathbf{1}\,0\,0\,0\,1\,1\,0\,0\,0\,\mathbf{0}\,1\,1\,\mathbf{1}\,1\,0. \qquad (5)$$

Конфигурация ошибочных разрядов в (5) такова, что они не попадают в 11-разрядное проверочное окно этого кода, что соответствует нерегулярному синдрому ошибки.После первой перестановки вида $i\rightarrow(i+2)\bmod 23$ получаем кодовое слово

$$Z_{err}^{(3)(1:2)}=1\,0\,0\,0\,\mathbf{1}\,0\,1\,0\,0\,1\,\mathbf{1}\,0\,0\,0\,0\,0\,0\,0\,1\,0\,\mathbf{0}\,1\,1.$$

После второй перестановки $i\rightarrow(i+2)\bmod 23$ получаем кодовое слово

$$Z_{err}^{(3)(1:4)}=1\,0\,\mathbf{1}\,1\,0\,\mathbf{1}\,0\,0\,0\,1\,\mathbf{0}\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1. \qquad (6)$$

Слово (6) уже соответствует регулярному синдрому ошибки, попадает в 11-разрядное проверочное окно (выделено цветом) и легко исправляется. В завершение остается лишь сделать две обратные перестановки и получить исправленное кодовое слово Z.

Таким образом, для декодирования кода Голея не требуется ни сложных алгебраических преобразований, ни хранения дополнительной информации.

В упрощении процесса декодирования проявляется феномен ИДЦК: при наличии итераций количество исправляемых ошибок (т. е. параметр $d_{\min}$) не изменяется, а сложность вычислений резко уменьшается. В итоге значительно сокращается время декодирования ошибок высокой кратности.

6. Определение корректирующей способности циклических кодов с помощью перестановок

Решить задачу определения корректирующей способности помехо-устойчивых кодов в общем виде пока не представляется возможным. Сегодня можно лишь говорить об отдельных классах кодов. Для вероятностных кодов, в частно-

Не является переизданием

сти LDPC-кодов, эта задача решена лишь на уровне различных видов аппроксимаций экспериментальных данных аналитическими выражениями [14].

Задача определения корректирующей способности кодов основана на переборных алгоритмах, что и обуславливает ее принадлежность к NP-сложным задачам. Необходимо в максимальной степени упростить перебор кодовых слов (синдромов ошибок), используя различные свойства конкретных классов кодов.

Наиболее подходят к достижению этой цели циклические коды.

Во-первых, благодаря свойству цикличности можно заменить n конфигураций ошибок n-разрядного кодового слова одной конфигурацией ошибок. Кроме того, для каждой ошибки кратности τ имеется τ эквивалентных конфигураций ошибок. Достаточно использовать только одну (базовую) конфигурацию ошибок. В итоге можно уменьшить объем анализируемых данных в $n\tau$ раз.

Во-вторых, сократить перебор и сделать его более формализованным можно с помощью циклических перестановок разрядов кодовых слов.

Рассмотрим подробнее метод нахождения корректирующей способности ИДЦК. Этот метод основан на обобщенном алгоритме точного определению степени исправления ошибок различной кратности. Для каждой кратности τ ошибки имеются свои варианты алгоритма, которые отличаются по сложности вычислений: с увеличением кратности τ ошибки возрастает сложность и время выполнения алгоритма. При этом для каждой кратности ошибки указанные варианты алгоритма можно выполнять параллельно.

На основе этого метода можно не только определять точное количество исправляемых ошибок различной кратности, но также и их зависимость от параметров кода.

7. Алгоритм определения корректирующей способности ИДЦК относительно двойных ошибок

Пусть задан циклический (n, k)-код с кодовым словом

$$Z=z_1\, z_2\, \ldots\, z_{n-1}\, z_n, \qquad (7)$$

1. Установить номер итерации w=1. Сформировать текущее кодовое слово Z_c, совпадающее с исходным словом (7).

2. Из n разрядов кодового слова Z_c сформировать три проверочных окна:

левое проверочное окно X_{lt} (младшие r разрядов текущего слова Z_c)

$$z_1, z_2, \ldots, z_{r-1}, z_r,$$

правое проверочное окно X_{rg} (старшие r–1 разрядов текущего слова Z_c)

$$z_m, z_{m+1}, \ldots, z_n, \quad (m=n-2r-2)$$

ТОЛЬКО ДЛЯ ЧТЕНИЯ

и основное проверочное окно $X^{(w)}$ (все разряды исходного слова (7)).

3. В основном проверочном окне $X^{(w)}$ отметить те разряды слова Z_c, которые на итерации w расположены в окнах X_{lt} и X_{rg}.

Если все разряды в окне $X^{(w)}$ будут отмечены, тогда все разряды слова Z проверяются в течении w итераций, идти к п. 5.

4. Увеличить номер итерации $w=w+1$. Выполнить перестановку (например, вида $i \rightarrow (i+2) mod n$) разрядов слова Z_c и из переставленных разрядов сформировать следующее текущее кодовое слово Z_c.

Если переставленное слово Z_c совпадает с исходным словом (7), тогда идти к п. 5, иначе к п. 2.

5. Конец.

$\perp$

Основная идея этого алгоритма состоит в том, что в левое и правое проверочные окна в течение нескольких итераций (может быть и одновременно) попадают разряды конфигураций ошибок кратности τ. Наличие таких конфигураций гарантирует проверку соответствующих ошибок кратности τ.

Этот алгоритм удобно выполнять в таблице.

Пример 2. Определим степень корректирующей способности относительно двойных ошибок циклического (31,26)-кода Хэмминга с примитивным порождающим полиномом $g(x)$ степени 5 (табл. 2). Таким ошибкам соответствуют 30 конфигураций двойных ошибок:

$z_1, z_2,$
$z_1, z_3,$
. . .
$z_1, z_{31}.$

После формирования трех проверочных окон видно, что на первой итерации проверяются разряды z_1, z_2, z_3, z_4, z_5 и $z_{28}, z_{29}, z_{30}, z_{31}$ (выделены красным цветом). На второй итерации с перестановкой вида $i \rightarrow (i+2) \bmod 31$ в левое проверочное окно попадают новые разряды z_7 и z_9, а в правое проверочное окно – разряды z_{24} и z_{26}. В основном проверочном окне $X^{(w)}$ отмечаем эти новые разряды.

На третьей итерации в левое проверочное окно попадают новые разряды z_{13} и z_{17}, а в правое проверочное окно – разряды z_{16} и z_{20}. На четвертой итерации в обеих крайних окнах добавляются по одному новому разряду: z_{25} и z_8. На пятой итерации в обеих крайних окнах добавляются еще по одному новому разряду: z_{18} и z_{15}. На этом перестановки приводят к исходному кодовому слову.

Не является переизданием

Таблица 2

Проверка разрядов кодового слова Z по итерациям

Итерации	Левое окно X_{lt}	Основное проверочное окно $X^{(w)}$	Правое окно X_{rg}
1	$z_1z_2z_3z_4z_5$	$z_1z_2\,z_3\,z_4z_5z_6\,z_7z_8z_9z_{10}z_{11}z_{12}z_{13}z_{14}z_{15}z_{16}$ $z_{17}z_{18}z_{19}z_{20}z_{21}z_{22}z_{23}z_{24}z_{25}z_{26}z_{27}z_{28}z_{29}z_{30}z_{31}$	$z_{28}z_{29}z_{30}z_{31}$
2	$z_1z_3z_5z_7z_9$	$z_1z_2\,z_3\,z_4z_5z_6\,z_7z_8z_9z_{10}z_{11}z_{12}z_{13}z_{14}z_{15}z_{16}$ $z_{17}z_{18}z_{19}z_{20}z_{21}z_{22}z_{23}z_{24}z_{25}z_{26}z_{27}z_{28}z_{29}z_{30}z_{31}$	$z_{24}z_{26}z_{28}z_{30}$
3	$z_1z_5z_9z_{13}z_{17}$	$z_1z_2\,z_3\,z_4z_5z_6\,z_7z_8z_9z_{10}z_{11}z_{12}z_{13}z_{14}z_{15}z_{16}$ $z_{17}z_{18}z_{19}z_{20}z_{21}z_{22}z_{23}z_{24}z_{25}z_{26}z_{27}z_{28}z_{29}z_{30}z_{31}$	$z_{16}z_{20}z_{24}z_{28}$
4	$z_1z_9z_{17}z_{25}z_2$	$z_1z_2\,z_3\,z_4z_5z_6\,z_7z_8z_9z_{10}z_{11}z_{12}z_{13}z_{14}z_{15}z_{16}$ $z_{17}z_{18}z_{19}z_{20}z_{21}z_{22}z_{23}z_{24}z_{25}z_{26}z_{27}z_{28}z_{29}z_{30}z_{31}$	$z_{31}z_8z_{16}z_{24}$
5	$z_1z_{17}z_2z_{18}z_3$	$z_1z_2\,z_3\,z_4z_5z_6\,z_7z_8z_9z_{10}z_{11}z_{12}z_{13}z_{14}z_{15}z_{16}$ $z_{17}z_{18}z_{19}z_{20}z_{21}z_{22}z_{23}z_{24}z_{25}z_{26}z_{27}z_{28}z_{29}z_{30}z_{31}$	$z_{30}z_{15}z_{31}z_{16}$

Таким образом, для заданного кода можно после 5 итераций выявить и исправить 21 конфигурацию двойных ошибок, что составляет 68 % от всех возможных двойных ошибок. Таким образом, традиционный полный перебор всех возможных вариантов сравнения кодовых слов заменен алгоритмом направленного поиска решения поставленной задачи, что приводит к значительной экономии времени вычислений.

8. Взаимосвязь между корректирующей способностью и параметрами ИДЦК

Итеративные методы декодирования детерминированных кодов позволяют с новых позиций подойти к решению ключевой задачи кодирования: нахождения минимальной избыточности кода для обеспечения заданной корректирующей способности кода.

Для (n, k)-кода можно вначале можно найти точную степень (выраженную, например, в %) исправления ошибок заданной кратности τ. Постепенно увеличивая значение $(n–k)$, что эквивалентно увеличению длины проверочного окна X, можно постепенно увеличить степень исправления заданных ошибок.

Например, для полного исправления всех двойных ошибок из Примера 2 достаточно увеличить на единицу разрядность проверочного слова кода. С этой целью можно использовать порождающий полином $g(x)(x+1)$, что эквивалентно переходу к (31,25)-коду Абрамсона.

Приведенный в [11] итеративно декодируемый циклический (15,11)-код позволяет исправить 40 % тройных ошибок. Для повышения корректирующей способности до 100 % необходимо увеличить проверочное слово до 7 разрядов.

При необходимости можно уменьшить корректирующую способность кода.

ТОЛЬКО ДЛЯ ЧТЕНИЯ

Разумеется, остается еще задача различения между собой ошибок различной кратности (либо с помощью вспомогательного циклического кода [11] либо традиционным способом с помощью дополнительной информации от демодулятора [9]).

В традиционных циклических кодах разрядность проверочного слова Ψ (длину проверочного окна *X*) выбирается из того требования, чтобы в нем можно было представить все синдромы ошибок заданной кратности τ. При итеративном декодировании синдромы ошибок распределяются по отдельным итерациям, что позволяет уменьшить разрядность слова Ψ (проверочного окна *X*), а также поместить в нем синдромы ошибок более высокой кратности [6]. А уменьшение разрядности слова Ψ позволяет соответственно увеличить разрядность *k* информационного слова *I*, что в конечном итоге приводит к увеличению скорости кода.

В известных кодах (например, кодах БЧХ), для соседних значений τ и τ+1 параметры *n*, *k* и *r* кода изменяются с большими интервалами. При итеративном декодировании указанные интервалы значительно меньше, что удобно для практической реализации.

9. Исследование корректирующей способности вероятностных кодов

Рассмотрим подробнее LDPC-коды. Эти коды являются наиболее полным воплощением на практике теоремы Шеннона о передаче информации по каналу связи с помехами. При бесконечно большой длине кодового слова (что эквивалентно бесконечно большом времени декодирования) можно исправить практически все ошибки. Если бы при этом была поставлена задача исправления всех ошибок в течение одного такта времени, тогда потребовался бы декодер бесконечной сложности. Такой помехоустойчивый код был бы нереализуем на практике.

Частично обойти проблему сложности удалось в LDPC-кодах благодаря многотактному (итеративному) способу декодирования: сложность декодирования уменьшается до приемлемых значений за счет увеличения тактов (итераций) декодирования [22].

Аналогичным способом решается и проблема корректирующей способности: за счет увеличения итераций декодирования увеличивается число обнаруживаемых и исправляемых ошибок.

Характерной особенностью вероятностных итеративно декодируемых кодов является то, что степень их корректирующей способности невозможно выразить точными аналитическими оценками.

Можно предположить, что вероятностным кодам будут наиболее соответствовать вероятностные оценки для каждой кратности ошибки. Следовательно, корректирующую способность LDPC-кодов для ошибок кратности τ_i и менее можно представить в виде последовательности нецелых чисел

$$p_1(), p_2(), \ldots p_{i-1}(), p_i(),$$

Не является переизданием

где $p_i()$ – вероятность исправления ошибок кратности τ_i,

$p_1()=1$, $p_i() \leq p_{i-1}()$.

Величина $p_1()$ для одиночной ошибки всегда равна 1, а остальные величины $p_i()$ будут постепенно возрастать на каждой итерации от 0 до 1, не достигая единицы для больших кратностей ошибок.

Похожая ситуация с изменением корректирующей способности имеет место при итеративном декодировании циклических кодов на основе перестановок разрядов кодового слова [11]. При этом степень исправления ошибок также может быть выражена последовательностью нецелых чисел (или в процентах к общему числу ошибок) [23].

Факторы корректирующей способности (изменяемость и представление нецелыми числами) итеративных кодов различных классов делает некорректным ее сравнение с корректирующей способностью традиционных детерминированных кодов на основе единственного, константного и целого параметра d_{min}.

10. Обсуждение критериев корректирующей способности кодов

В современной теории помехоустойчивого кодирования имеется много нерешенных пока проблем и важнейшей из них является проблема эффективного (не NP-сложного) нахождения корректирующей способности кода.

С одной стороны, имеется большое разнообразие помехоустойчивых кодов. С другой стороны, известны два основных критерия оценки кодов:

– по расстоянию к границе Шеннона на графиках BER;

– на основе минимального кодового расстояния и асимптотических границ.

Родоначальник вероятностной ветви теории информации К. Шеннон также рассматривал коды только с вероятностной стороны и, поэтому, первый критерий до сегодняшнего дня является основным для большинства помехоустойчивых кодов.

Однако для детерминированных кодов давно известны другие критерии (кодовое расстояние d_{min}) и другие границы – асимптотические [7]. Эти границы позволяют оценить свойства кодов по выявлению ошибок в зависимости от введенной избыточности. Предложенные ИДЦК наиболее близки к асимптотической границе Хэмминга, и с этих позиций они имеют существенное преимущество относительно известных кодов, в частности, кодов БЧХ [20]. На практике это преимущество проявляется в увеличении скорости кодов и передаче большего количества полезной информации за единицу времени (табл. 3).

Относительно параметра $d_{\min}$ можно отметить, что он дает очень приблизительную оценку корректирующей способности даже для детерминированных кодов, поскольку для большинства кодов можно исправлять ошибки и за пределами $d_{\min}$ [23]. Наиболее точные оценки можно получить путем непосредственного определения количества исправляемых ошибок, что относительно легко сделать для кодов ИДЦК.

ТОЛЬКО ДЛЯ ЧТЕНИЯ

LDPC-коды имеют заслуженное преимущество среди вероятностных кодов благодаря максимальной близости к границе Шеннона, а попытки их оценки с помощью кодового расстояния $d_{\min}$ является некорректными и лишь уменьшают их достоинства.

Таблица 3
Параметры итеративных и традиционных кодов

Код	Скорость кода, k/n	Разрядность проверочного слова, r	Количество исправляемых ошибок	Степень исправления ошибок	Число итераций
(15,11) Хэмминга	73 %	4	1	100 %	1
(15,7) БЧХ	47 %	8	2	100 %	1
(15,5) БЧХ	33 %	10	3	100 %	1
(31,16) БЧХ	52 %	15	3	100 %	1
(31,21) БЧХ	68 %	10	2	100 %	1
(15,11) ИДЦК	73 %	4	2	87 %	3
(15,11) ИДЦК	73 %	4	3	40 %	4
(17,10) ИДЦК	59 %	7	3	100 %	4
(31,25) ИДЦК	81 %	6	2	100 %	5

Если для каждого типа кода использовать соответствующие им критерии, тогда можно увидеть интересные закономерности между рассмотренными вероятностными и детерминированными итеративными кодами.

Во-первых, все итеративные коды увеличивают свою корректирующую способность постепенно, с увеличением числа итераций.

Во-вторых, для указанных итеративных кодов корректирующую способность можно задавать в процентах для ошибок различной кратности.

В-третьих, итеративные коды – высокоскоростные коды.

В ИДЦК можно выбрать минимально возможную разрядность проверочного слова для обеспечения заданной корректирующей способности кода. Этим итеративные коды отличаются от других помехоустойчивых кодов, которые сразу формируют свою корректирующую способность в виде целочисленного параметра d_{min}.

Каждый тип итеративного кода имеет свою сферу применения, где полностью раскрываются его достоинства и где он является лучшим. С этих позиций LDPC-коды являются лучшими для больших длин кодов, для которых допустимы приближенные оценки их корректирующей способности с достоверностью менее

Не является переизданием

100 %, а ИДЦК – для небольших длин кодов с точными оценками их корректирующей способности.

Таким образом, проведенные исследования показали следующее:

– необходимо различать вероятностные и детерминированные помехоустойчивые коды;

– итеративный подход к декодированию возможен не только для вероятностных кодов, но и для детерминированных кодов;

– имеется много общего между итеративными вероятностными и детерминированными кодами, однако критерии оценки их корректирующей способности различны;

– достоинствами предложенного перестановочного декодирования являются уменьшения сложности вычислений и полная формализация действий в алгоритмах определения корректирующей способности детерминированных кодов;

– требуются дальнейшие исследования по упрощению алгоритмов определения корректирующей способности кодов.

Безусловно, эти рассуждения являются предварительными, развитие теории помехоустойчивых кодов продолжается.

11. Выводы

1. Выполнено математическое обоснование перестановочного декодирования циклических кодов. Предложен генератор циклических перестановок на основе двух автоматов Мура. С помощью автомата низкого уровня происходит формирование одного состояния автомата высокого уровня. С помощью состояний автомата высокого уровня формируются циклы перестановок разрядов кодового слова.

2. На основе автоматного представления циклических кодов проведено исследование регулярных и нерегулярных состояний автомата, приведены количественные оценки. Доказано, что под влиянием циклических перестановок происходит преобразование нерегулярных синдромов в регулярные, а с увеличением числа регулярных синдромов уменьшается сложность декодирования циклических кодов.

3. Предложены методы точного определения корректирующей способности ИДЦК на основе циклических перестановок. Для этих кодов, как и для других классов кодов, сложность решения этой задачи будет существенно возрастать с увеличением кратности ошибок, но медленнее в $n\tau$ раз благодаря более простой кодовой структуре.

4. Обоснована взаимосвязь между корректирующей способностью и параметрами ИДЦК. Показано значительное повышение скорости ИДЦК по сравнению с известными кодами БЧХ (от 1.2 до 2.2 раза для приведенных примеров).

Только для чтения

Литература

1. Шеннон К. Работы по теории информации и кибернетике. М., 1963. 829 с.
2. Урсул А. Д. Природа информации. Философский очерк. М.: Политиздат, 1968. 288 с.
3. Hartley R. V. L. Transmission of Information // Bell System Technical Journal. 1928. Vol. 7, Issue 3. P. 535–563. doi: https://doi.org/10.1002/j.1538-7305.1928.tb01236.x
4. Колмогоров А. Н. Три подхода к определению понятия “количество информации” // Проблемы передачи информации. 1965. Т. 1, Вып. 1. С. 3–11.
5. Колмогоров А. Н. Теория информации и теория алгоритмов. М.: Наука, 1987. 304 с.
6. Булычёв И. И., Сорока М. Ю. О природе и сущности информации // Ноосферные исследования. 2016. Вып. 1-2 (13-14). С. 191–207.
7. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976. 596 с.
8. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. М.: Изд. дом «Вильямс», 2004. 1104 с.
9. Кларк Дж. мл., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи. М.: Радио и связь, 1987. 392 с.
10. Dumer I., Micciancio D., Sudan M. Hardness of approximating the minimum distance of a linear code // IEEE Transactions on Information Theory. 2003. Vol. 49, Issue 1. P. 22–37. doi: https://doi.org/10.1109/tit.2002.806118
11. Semerenko V. Iterative hard-decision decoding of combined cyclic codes // Eastern-European Journal of Enterprise Technologies. 2018. Vol. 1, Issue 9 (91). P. 61–72. doi: https://doi.org/10.15587/1729-4061.2018.123207
12. Garrammone G., Declercq D., Fossorier M. P. C. Weight Distributions of Non-Binary Multi-Edge Type LDPC Code Ensembles: Analysis and Efficient Evaluation // IEEE Transactions on Information Theory. 2017. Vol. 63, Issue 3. P. 1463–1475. doi: https://doi.org/10.1109/tit.2016.2647724
13. Computing the Minimum Distance of Nonbinary LDPC Codes / Liu L., Huang J., Zhou W., Zhou S. // IEEE Transactions on Communications. 2012. Vol. 60, Issue 7. P. 1753–1758. doi: https://doi.org/10.1109/tcomm.2012.050812.110073a
14. Урывский Л. А., Осипчук С. А. Исследование свойств помехоустойчивых кодов класса LDPC. Наукоемкие технологии в инфокоммуникациях: обработка информации, кибербезопасность, информационная борьба: колект. моногр. / ред. В. М. Безрук, В. В. Баранник. Харьков, 2017. С. 137–139.
15. Error-Correction Coding and Decoding. Bounds, Codes, Decoders, Analysis and Applications / Tomlinson M., Tjhai C. J., Ambroze M. A., Ahmed M., Jibril M. Springer, 2017. doi: https://doi.org/10.1007/978-3-319-51103-0
16. Distance Properties of Short LDPC Codes and Their Impact on the BP, ML and Near-ML Decoding Performance / Bocharova I. E., Kudryashov B. D., Skachek V., Yakimenka Y. // Lecture Notes in Computer Science. 2017. P. 48–61. doi: https://doi.org/10.1007/978-3-319-66278-7_5

Не для продажи

17. Butler B. K., Siegel P. H. Error Floor Approximation for LDPC Codes in the AWGN Channel // IEEE Transactions on Information Theory. 2014. Vol. 60, Issue 12. P. 7416–7441. doi: https://doi.org/10.1109/tit.2014.2363832

18. Берлекэмп Э. Алгебраическая теория кодирования. М.: Мир, 1971. 477 с.

19. Semerenko V. P. Burst-Error Correction for Cyclic Codes // IEEE EUROCON 2009. 2009. doi: https://doi.org/10.1109/eurcon.2009.5167864

20. Семеренко В. П. Параллельное декодирование кодов Боуза-Чоудхури-Хоквингема // Электронное моделирование. 1998. № 1. С. 82–87.

21. Семеренко В. П. Теорія циклічних кодів на основі автоматних моделей: монографія. Вінниця: ВНТУ, 2015. 444 с.

22. Галлагер Р. Коды с малой плотностью проверок на четность. М.: Мир, 1966. 144 с.

23. Семеренко В. П. Оценка корректирующей способности циклических кодов на основе их автоматных моделей // Восточно-Европейский журнал передовых технологий. 2015. Т. 2, № 9 (74). С. 16–24. doi: https://doi.org/10.15587/1729-4061.2015.39947

Только для чтения