

УДК 519.725.2

ДЕКОДИРОВАНИЕ ОБОБЩЕННЫХ КАСКАДНЫХ КОДОВ С ВНУТРЕННИМИ ПОЛЯРНЫМИ КОДАМИ

П. К. Семенов¹,

ассистент

Санкт-Петербургский государственный политехнический университет

Приводится алгоритм декодирования для обобщенных каскадных кодов с внутренними полярными кодами. Предлагаемый декодер является списочным и основан на описанных в работе структурах данных — деревьях решений для внутреннего и для всего каскадного кодов. Для этих структур данных в работе рассматривается способ эффективной реализации. Предлагаемый в работе алгоритм декодирования позволяет достичь лучшей корректирующей способности по сравнению с традиционным декодером обобщенных каскадных кодов и декодером по максимуму правдоподобия для полярных кодов.

Ключевые слова — обобщенные каскадные коды, декодирование, полярные коды.

Введение

В работе [1] были предложены полярные коды и доказано, что они достигают пропускной способности симметричных каналов без памяти с двоичным входом. Однако на практике полярные коды при использовании последовательного декодера [1] обеспечивают большую вероятность ошибки декодирования, чем классические коды. В статье [2] была предложена конструкция обобщенных каскадных кодов (ОКК) с внутренними полярными кодами, позволяющая обеспечить меньшую вероятность ошибки декодирования по сравнению с низкоплотными кодами. При этом использовался многошаговый декодер, имеющий критическую зависимость от оценок информационных символов с предшествующих шагов: ошибки на предыдущих шагах не могут быть исправлены на последующих, что приводит к их катастрофическому распространению.

В данной работе предлагается алгоритм декодирования для ОКК с внутренними полярными кодами, основанный на списочном декодировании внешних кодов. Главным отличием от стандартного декодера ОКК является введение дерева решений, связывающего списочное декодирование внешних кодов со списочным декодированием внутренних кодов. Таким образом, пред-

лагаемый алгоритм можно рассматривать как обобщение списочного декодера Тала — Варди [3] полярных кодов, который позволяет достичь корректирующей способности декодера по максимуму правдоподобия при достаточно небольшой длине списка. С другой стороны, метод деревьев решений может быть использован и для случая других внутренних кодов. Основным требованием является лишь существование для них декодера, последовательно оценивающего значения информационных символов. Для полярных кодов таким декодером выступает метод последовательного исключения [1] с некоторыми модификациями [4].

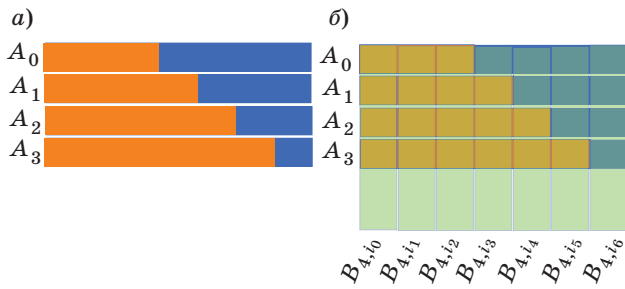
Обобщенные каскадные коды

Обобщенные каскадные коды [5, 6] основаны на семействе внешних (ограничимся случаем линейных внешних кодов) (N, K_i, D_i) -кодов A_i над $GF(2^{m_i})$, $0 \leq i < l$, и внутреннем (n, k_0, d_0) -коде B_0 , который рекурсивно разбивается на семейства непересекающихся равномошных подкодов в соответствии с правилами

$$B_0 = \bigcup_{j=0}^{2^{m_0}-1} B_{1,j}, B_{i,j} = \bigcup_{t=0}^{2^{m_i}-1} B_{i+1,j \cdot 2^{m_i} + t}, 0 \leq j < 2^{\sum_{t=0}^i m_t},$$

где $B_{i,j}$ — некоторые его (n, k_i, d_i) -подкоды, а индекс j обозначает естественную нумерацию «слева направо» ветвей в i -м уровне дерева разбиения (см. примеры на рис. 1, а, б).

¹ Научный руководитель — кандидат технических наук, доцент Санкт-Петербургского государственного политехнического университета П. В. Трифионов.



■ Рис. 1. Кодирование ОКК: а — внешними кодами; б — с помощью внутреннего кода

На размерность подкодов на j -м уровне разбиения накладывается ограничение $k_j = \sum_{t=j}^{l-1} m_t$, $0 \leq j \leq l$; таким образом, все коды $B_{l,j}$ последнего, l -го уровня разбиения имеют нулевую размерность и состоят из единственного фиксированного кодового слова внутреннего кода B_0 . Кодирование данных проводится следующим образом.

1. Данные кодируются внешними кодами: пусть $(c_{0,0}, \dots, c_{0,N-1}), \dots, (c_{l-1,0}, \dots, c_{l-1,N-1})$ — полученные при этом кодовые слова.

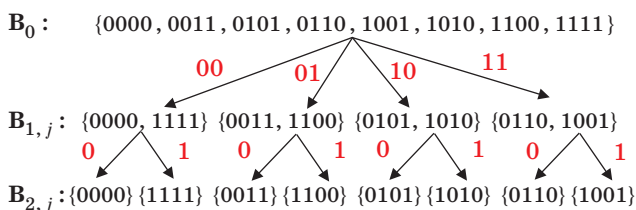
2. Для всех $j = 0 \dots N-1$ символы $c_{i,j} \in GF(2^{m_i})$, $0 \leq i < l$ преобразуются в кортежи длин m_i с использованием некоторых фиксированных базисов $GF(2^{m_i})$. Эти кортежи объединяются в вектор длины k_0 , который однозначно задает путь в иерархии разбиения внутреннего кода B_0 (см. рис. 1).

В примере разбиения внутреннего (4, 3, 2)-кода с одной проверкой на четность для обобщенного каскадного кода с двумя внешними кодами (рис. 2): A_0 — (8, 1, 8)-код с повторениями над $GF(4)$ и A_1 — расширенный (8, 4, 4)-код Хэмминга над $GF(2)$.

Если все используемые коды A_i и $B_{i,j}$ являются линейными, то множество получаемых таким образом векторов образует линейный блочный

$$\left(nN, \sum_{i=0}^{l-1} K_i m_i, \min(D_0 d_0, \dots, D_{l-1} d_{l-1}) \right)\text{-код.}$$

Стоит отметить, что при разбиении внутреннего кода элементы одного семейства являются различными смежными классами одного и того же подкода в силу выбора внутреннего кода линейным.



■ Рис. 2. Разбиение внутреннего кода

И, кроме того, для разбиения на рис. 2 подкоды $B_{2,j}$ имеют бесконечное минимальное расстояние: $B_{1,j} - 4$, а $B_0 - 2$.

Многие коды с наилучшими параметрами были получены как обобщенные каскадные [5].

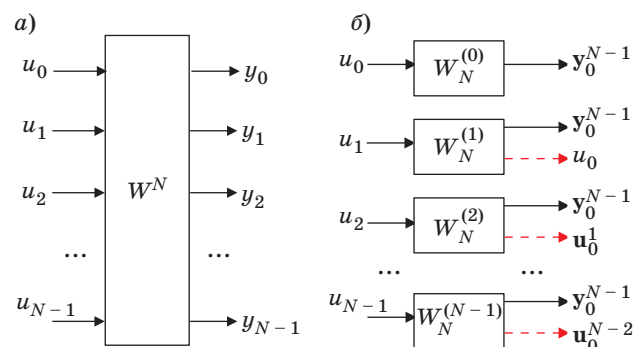
Полярные коды на основе ядра Арикана

Назовем ядром Арикана матрицу $G = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$,

а через $G^{\otimes s}$ обозначим ее s -ю кронекеровскую степень — матрицу размерности $2^s \times 2^s$. Пусть B_{2^s} — это $(2^s \times 2^s)$ -матрица такой перестановки для вектора длины 2^s , что номер новой позиции i -го элемента получается как обратная запись представленного в виде s -битной последовательности числа i . Например, $4 \equiv (1\ 0\ 0) \rightarrow (0\ 0\ 1) \equiv 1$, $3 \equiv (0\ 1\ 1) \rightarrow (1\ 1\ 0) \equiv 6$. Будем обозначать скалярный канал со входом u и выходом y как $W(y|u)$, отождествляя его с функцией плотности условной вероятности выходного символа. В случае передачи последовательности данных $\mathbf{u}_0^{N-1} = (u_0 \ u_1 \ \dots \ u_{N-1})$ размера N через канал $W(y|u)$ будем рассматривать эквивалентный векторный канал

$$W^N(\mathbf{y}_0^{N-1} | \mathbf{u}_0^{N-1}) = \prod_{i=0}^{N-1} W(y_i | u_i)$$

с выходом $\mathbf{y}_0^{N-1} = (y_0 \ y_1 \ \dots \ y_{N-1})$ (рис. 3, а). В статье [1] был предложен метод поляризации симметричного канала $W^N(\mathbf{y}_0^{N-1} | \mathbf{u}_0^{N-1})$ без памяти с двоичным входом: предполагается, что $N = 2^s$ и через канал передается преобразованный блок данных $\mathbf{u}_0^{N-1} B_N G^{\otimes s}$. Метод заключается в представлении $W^N(\mathbf{y}_0^{N-1} | \mathbf{u}_0^{N-1})$ как совокупности $\bar{W} \equiv \{W_N^{(i)}(\mathbf{y}_0^{N-1}, \mathbf{u}_0^{i-1} | u_i), 0 \leq i < N\}$ двоичных каналов (рис. 3, б), которая обладает следующими свойствами.



■ Рис. 3. Эквивалентные представления канала: а — векторный канал W^N ; б — эквивалентные подканалы $W_N^{(i)}$ (пунктирная линия означает достоверную передачу через канал)

1. Эквивалентность W и \bar{W} : суммарная пропускная способность $\sum_{i=0}^{N-1} C(W_N^{(i)})$ равна $C(W)$ при условии, что для каждого $W_N^{(i)}(y_0^{N-1}, u_0^{i-1} | u_i)$ известны точные значения информационных символов u_0^{i-1} (см. рис. 3, б).

2. Асимптотическая поляризация каналов из \bar{W} : при $N \rightarrow \infty$ относительная доля $W_N^{(i)}$ в \bar{W} таких, что $C(W_N^{(i)}) \rightarrow 1$, стремится к $C(W)$, а пропускная способность остальных $N(1 - C(W))$ каналов стремится к нулю.

Поляризационный эффект достигается применением невырожденного преобразования $\mathbf{B}_N \mathbf{G}^{\otimes s}$ к данным. И, таким образом, поляризующее преобразование канала $W^{2N}(y_0^{2N-1} | u_0^{2N-1})$ основано на поляризации векторных каналов меньшей размерности $W^i(y_0^{i-1} | u_0^{i-1})$, $i = N, \frac{N}{2}, \frac{N}{4}, \dots, 1$.

Эта декомпозиция для плотностей $W_N^{(i)}(y_0^{N-1}, u_0^{i-1} | u_i)$ вероятностей выражается как [1]

$$W_{2N}^{(2i)}(y_0^{2N-1}, u_0^{2i-1} | u_{2i}) = \frac{1}{2} \sum_{u_{2i+1}} W_N^{(i)}(y_0^{N-1}, u_{0,e}^{2i-1} \oplus u_{0,o}^{2i-1} | u_{2i} \oplus u_{2i+1}) \times W_N^{(i)}(y_N^{2N-1}, u_{0,o}^{2i-1} | u_{2i+1}); \quad (1)$$

$$W_{2N}^{(2i+1)}(y_0^{2N-1}, u_0^{2i} | u_{2i+1}) = \frac{1}{2} W_N^{(i)}(y_0^{N-1}, u_{0,e}^{2i-1} \oplus u_{0,o}^{2i-1} | u_{2i} \oplus u_{2i+1}) \times W_N^{(i)}(y_N^{2N-1}, u_{0,o}^{2i-1} | u_{2i+1}) \quad (2)$$

где $u_{0,e}^{2i-1}$ и $u_{0,o}^{2i-1}$ обозначают компоненты вектора u_0^{2i-1} только с четными и нечетными номерами соответственно. Таким образом, вероятности передаваемых битов в каналах $W_N^{(i)}(y_0^{N-1}, u_0^{i-1} | u_i)$ оцениваются через рекуррентные формулы (1), (2) с условием $W_1^{(1)}(y | u) = W(y | u)$ для выхода из рекурсии.

При $N \rightarrow \infty$ каналы $W_N^{(i)}$ будут либо абсолютно без шума, либо абсолютно ненадежными. Поэтому информационные символы u_i , передаваемые по каналам плохого качества, можно считать всегда фиксированными («замороженными»). Если сопоставить i -ю строку матрицы $\mathbf{B}_N \mathbf{G}^{\otimes s}$ каналу $W_N^{(i)}(y_0^{N-1}, u_0^{i-1} | u_i)$, то r наиболее надежным каналам из \bar{W} соответствует линейный код длины 2^s и размерности r , называемый полярным кодом. В итоге, чтобы задать полярный код порядка s , достаточно указать множество A индексов неиспользуемых строк матрицы $\mathbf{G}^{\otimes s}$.

SCDecoder(s, A):

1. for $i \leftarrow 0$ to $2^s - 1$ do
2. if $i \in A$ then $u_i \leftarrow 0$
3. else $u_i = \begin{cases} 0, & \frac{W_N^{(i)}(y_0^{N-1}, u_0^{i-1} | 0)}{W_N^{(i)}(y_0^{N-1}, u_0^{i-1} | 1)} \geq 1 \\ 1, & \text{в противном случае} \end{cases}$
4. return u_0^{N-1}

■ Рис. 4. Последовательный декодер Арикана полярных кодов

Был предложен [1] последовательный метод декодирования полярных кодов — алгоритм последовательного исключения (рис. 4). Этот алгоритм обладает существенным недостатком: при неверном оценивании информационного символа u_i все последующие компоненты u_j , $j = i + 1, \dots, (2^s - 1)$ вектора u_0^{N-1} , вероятно, также будут оценены ошибочно.

Обобщенные каскадные коды с внутренними полярными кодами

Несмотря на хорошие асимптотические свойства, на практически значимых длинах (~1000) полярные коды проигрывают классическим кодам, так как вероятности ошибок в эквивалентных подканалах $W_N^{(i)}$ остаются достаточно большими. Использование внешних кодов позволяет серьезно улучшить корректирующую способность полярных кодов [2].

Рассмотрим ОКК C с 2^l внешними кодами A_i над $GF(2)$ длины m (возможно $(m, 0)$ -кодов с бесконечным минимальным расстоянием) и внутренним полярным $(2^l, 2^l)$ -кодом B_0 со следующим разбиением: $B_0 = B_{1,0} \cup B_{1,1}$,

$$B_{i,j} = \bigcup_{t=0}^{2^i-1} B_{i+1,j \cdot 2^i + t}, \quad 0 \leq j < 2^i,$$

где $B_{i,j}$ состоит из таких кодовых слов B_0 , для которых первые i информационных символов совпадают с маркерами пути до вершины $B_{i,j}$ (для $l = 1$ пример приведен на рис. 5). Таким образом, разбиение является идеально сбалансированным бинарным деревом и совпадает со схемой кодирования данных ядром Арикана порядка l . Стоит отметить, что после процедуры кодирования таким кодом получается m кодовых слов полярного $(2^l, 2^l)$ -кода. В работе [2] был предложен метод, позволяющий синтезировать коды из представленного семейства исходя из правила равенства вероятностей ошибки декодирования внешних кодов в им соответствующих каналах из \bar{W} при заданном отношении сигнал/шум.

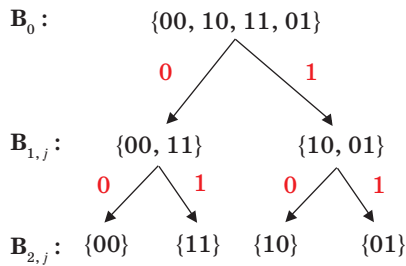


Рис. 5. Разбиение ядра Арикана порядка 1

И хотя рассмотренная каскадная конструкция позволяет уменьшить негативный эффект от слабой поляризации эквивалентных подканалов для коротких полярных кодов, она не лишена главного недостатка алгоритмов последовательного декодирования: результат каждого шага декодирования полностью зависит от достоверности оценок информационных битов с предшествующих шагов.

Списочный декодер ОКК с внутренними полярными кодами

Одним из способов преодоления проблемы критической зависимости от оценок информационных символов с предшествующих шагов является списочное декодирование, позволяющее одновременно отслеживать несколько вариантов таких оценок. Дерево всевозможных решений для полярного (4, 3)-кода представлено на рис. 6, а. Будем называть его деревом решений внутреннего кода (ДВК). Для того чтобы ограничить рост числа его листьев, необходимо на каждом шаге ранжировать текущие ветви по степени надежности размещенных в их листьях оценок и отсекал

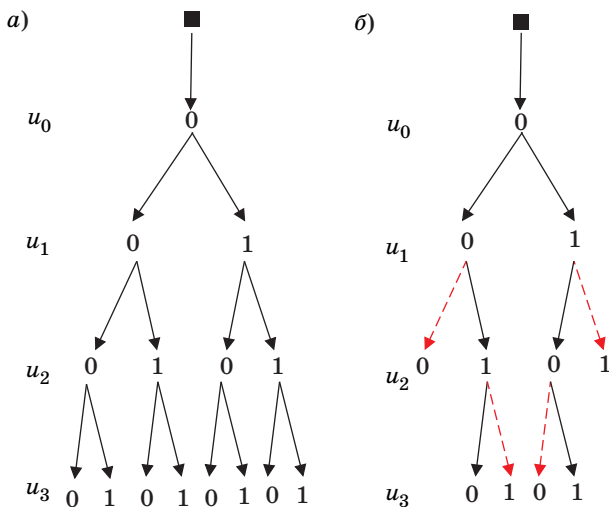


Рис. 6. Дерево решений для полярного (4, 3)-кода: а — идеальное дерево решений; б — выбор только двух ветвей на каждом шаге

те, что соответствуют наименее надежным вариантам. В качестве меры надежности оценок u_i выступают величины $W_N^{(i)}(\mathbf{y}_0^{N-1}, \mathbf{u}_0^{i-1} | u_i)$. Таким образом, все вычисления в списочном декодере необходимо проводить напрямую с вероятностями, что приводит к быстрому убыванию вычисляемых характеристик к нулю. Предлагается [3] на каждом шаге нормировать вероятности на их максимум по всем ветвям. Такой подход позволяет их дальнейшее использование при декодировании внешних кодов в метрике Евклида в каскадной конструкции с внутренними полярными кодами. Но его существенными недостатками являются использование вычислений с плавающей запятой и операций деления, невозможность получить истинные вероятности значений битов, что накладывает ограничение на используемые внешние коды. В связи с этим было показано [4], каким образом необходимо модифицировать алгоритм последовательного исключения, чтобы он вычислял величины $\log W_N^{(i)}(\mathbf{y}_0^{N-1}, \mathbf{u}_0^{i-1} | 0)$ и $\log W_N^{(i)}(\mathbf{y}_0^{N-1}, \mathbf{u}_0^{i-1} | 1)$ для i -го информационного символа. Назовем этот алгоритм как *LogSCDecoder*(s, A, i).

Основная идея алгоритма *LogSCDecoder*(s, A, i) заключается в решении следующей задачи: имея

$$\log(a_t) = \log W_N^{(i)}(\mathbf{y}_0^{N-1}, \mathbf{u}_0^{2i-1} \oplus \mathbf{u}_0^{2i-1} | t)$$

и

$$\log(b_t) = \log W_N^{(i)}(\mathbf{y}_0^{2N-1}, \mathbf{u}_0^{2i-1} | t)$$

для $t = 0, 1$, необходимо вычислить

$$\begin{aligned} \log W_{2N}^{(2i)}(\mathbf{y}_0^{2N-1}, \mathbf{u}_0^{2i-1} | u_{2i}) &= \log \frac{1}{2} (a_0 b_0 + a_1 b_1) = \\ &= \log \left(\frac{1}{2} e^{\log(a_0) + \log(b_0)} + \frac{1}{2} e^{\log(a_1) + \log(b_1)} \right). \end{aligned}$$

Эта проблема может быть решена приближенно. Так, не ограничивая общности, будем считать, что $\log(a_0) + \log(b_0) > \log(a_1) + \log(b_1)$. Тогда искомое выражение можно представить как

$$\begin{aligned} \log W_{2N}^{(2i)}(\mathbf{y}_0^{2N-1}, \mathbf{u}_0^{2i-1} | u_{2i}) &= \\ &= \log \frac{1}{2} + \log(a_0) + \log(b_0) + \\ &+ \log \left(1 + e^{\log(a_1) + \log(b_1) - \log(a_0) - \log(b_0)} \right). \end{aligned}$$

В случае, когда $\log(a_1) + \log(b_1) - \log(a_0) - \log(b_0)$ мало (например, меньше $q = -70$),

$$\log \left(1 + e^{\log(a_1) + \log(b_1) - \log(a_0) - \log(b_0)} \right) \approx 0.$$

В итоге аппроксимация для формулы (1) представляется в следующем виде:

$$\log W_{2N}^{(2i)}(\mathbf{y}_0^{2N-1}, \mathbf{u}_0^{2i-1} | u_{2i}) \approx \begin{cases} \log \frac{1}{2} + \log(a_0) + \log(b_0), c < -q \\ \log \frac{1}{2} + \log(a_0) + \log(b_0) + \log(1 + e^c), -q \leq c \leq q, \\ \log \frac{1}{2} + \log(a_1) + \log(b_1), c > q \end{cases}$$

где $c = \log(a_1) + \log(b_1) - \log(a_0) - \log(b_0)$.

Пусть теперь L — параметр длины списка декодера обобщенных каскадных кодов с внутренними полярными кодами: на каждом шаге декодирования в ДВК оставляются только L наиболее надежных ветвей (рис. 6, b — отсекаемые ветви обозначены пунктирной линией). В данной работе надежность ветвей предлагается оценивать с помощью списочных декодеров внешних кодов A_i . Пусть $\mathbf{x}_i = (x_{i,j})_{j=0}^{m-1}$ — это кодовое слово кода A_i . Тогда $(z_{j,0}, z_{j,1}, \dots, z_{j,l-1}) = (x_{0,j}, x_{1,j}, \dots, x_{l-1,j}) \times \mathbf{B}_N \mathbf{G}^{\otimes l}$ является кодовым словом внутреннего полярного тривиального $(2^l, 2^l)$ -кода B_0 . Таким образом, чтобы получить оценку кодового слова внешнего кода A_i , необходимо знать оценки i -х информационных символов всех m внутренних кодов. Пусть известны логарифмические отношения правдоподобия $LLR_j, 0 \leq j \leq m$ оценок i -х информационных символов всех внутренних кодов. Тогда с помощью некоторой процедуры $ListDecode(A_i, (LLR_0, LLR_1, \dots, LLR_{m-1}))$ списочного декодирования внешнего кода A_i может быть получен список $List$ его кодовых слов. На основе списка $List$ определяется как надежность ветвей, так и решения о возможных значениях информационных символов для внутренних полярных кодов. Будем называть структуру данных, сохраняющую списки кодовых слов внешних кодов, деревом решений ОКК (ДОКК). Его пример представлен на рис. 7.

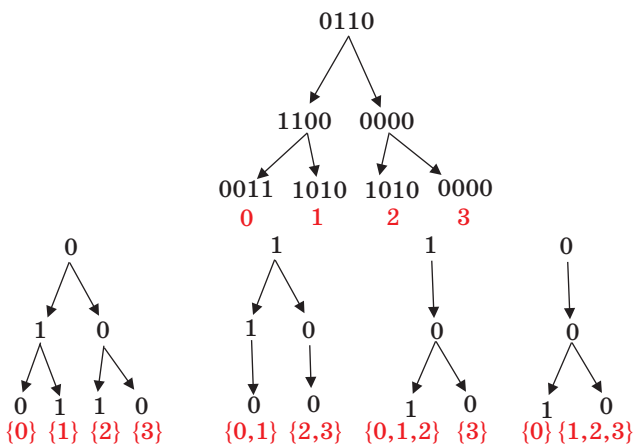


Рис. 7. Промежуточное ДОКК с внутренними полярными кодами

Дерево решений ОКК однозначно задает и ДВК. Будем называть i -й проекцией ДОКК дерево, получаемое при удалении размещенных в узлах кодовых слов всех символов, кроме i -го (см. рис. 7). Проекция соответствует возможным значениям при списочном декодировании полярного кода с помощью решений декодера внешних кодов. В целях исключения избыточности каждая ветвь ДВК соответствует какому-то набору ветвей ДОКК. Чтобы установить такое соответствие, требуется проводить идентификацию узлов в ДОКК. Для узлов на i -м уровне следует воспользоваться следующими способами:

- 1) каждому узлу v может быть поставлена в соответствие пара чисел (b, t) , где b — это номер родительской ветви в предшествующем уровне, а t — это номер ее потомка, соответствующего узлу v ;
- 2) каждому узлу v может быть поставлено в соответствие число t , равное номеру узла в i -м уровне при их нумерации «слева направо», что соответствует естественному перечислению листьев в последнем уровне дерева.

Совместное использование способов 1 и 2 позволяет при удалении каких-то ветвей в дереве единообразно перенумеровывать оставшиеся как $0 \dots n - 1$ без каких-либо пропусков на всех шагах. Возможны и другие способы реализации ДОКК, но перенумерация ветвей при операциях вставки и удаления листьев представляется наиболее удобной. Для примера на рис. 8 для всех ветвей деревьев-проекций приведены наборы им соответствующих номеров ветвей текущего ДОКК. Удаление ветви в ДОКК приводит к изменениям во всех деревьях внутренних кодов. На рис. 8 показан пример таких изменений (удаляемая ветвь в ДОКК и соответствующие изменения в ДВК обозначены зеленым цветом).

Дерево решений ОКК как структура данных должна поддерживать операции *Split* и *Prune*:

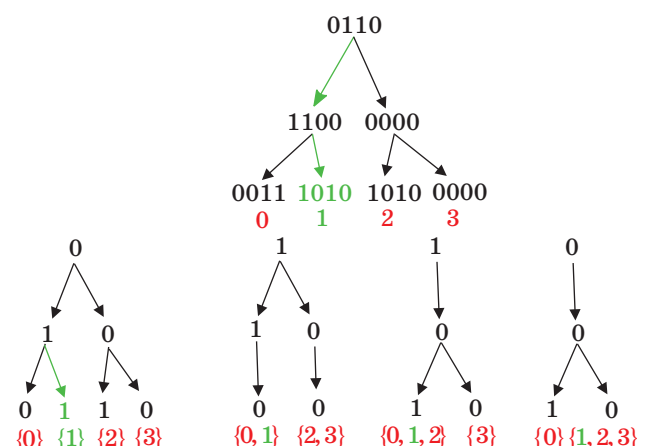


Рис. 8. Изменения в ДВК при удалении ветви в ДОКК

GCCDecoder(L):

1. For $i = 0$ to $2^l - 1$ do
2. For ветвь с номером b в дереве решений ОКК do
3. For $j = 0$ to $m - 1$ do
4. $\left(\log Pr(x_{j,i}^{(b)}=0), \log Pr(x_{j,i}^{(b)}=1)\right) = \text{logSCDecoder}(b, l, \emptyset, i)$
5. $LLR_t = \log Pr(x_{t,i}^{(b)}=1) - \log Pr(x_{t,i}^{(b)}=0), 0 \leq t < m$
6. $List = \text{ListDecode}(A_i, (LLR_0, LLR_1, \dots, LLR_{m-1}))$
7. For кодовое слово c_t в $List$ do
8.
$$m_{b,t} = \sum_{r=0}^{m-1} LLR_r \begin{cases} 1, & c_{t,r} = 1 \\ -1, & c_{t,r} = 0 \end{cases}$$
9. $\text{Split}(i, b, List)$
10. $\text{Prune}(L, \{m_{b,t}\}_{b,t})$
11. return ветви дерева решений ОКК

■ **Рис. 9.** Списочный декодер ОКК с внутренними полярными кодами

1) операция $\text{Split}(i, b, List)$ добавляет элементы списка $List$ в качестве потомков узлу под номером b в i -м уровне дерева;

2) операция $\text{Prune}(L, M)$: пусть π — перестановка, соответствующая упорядочиванию списка M из p вещественных чисел по убыванию, причем p равно текущему числу ветвей в дереве. Тогда Prune удаляет в нем j -ю ветвь, если $\pi(j) > L$ при условии начала нумерации в M с нуля.

В целях наглядности общего алгоритма декодирования будем считать, что обновление ДВК при изменениях в дереве решений ОКК происходит в процедурах Split и Prune . На рис. 9 представлен общий алгоритм списочного декодирования. В нем предполагается, что все ДВК являются проекциями ДОКК.

В строке 4 алгоритма величина $x_{j,i}^{(b)}$ означает i -й информационный символ при условии, что в качестве оценок ему предшествующих символов были выбраны расположенные в ветви b j -го ДВК. Оценка реализации величины $x_{j,i}^{(b)}$ проводится с помощью процедуры LogSCDecoder , вызываемой с дополнительным параметром b , определяющим, какая ветвь j -го ДВК была выбрана в качестве оценок предшествующих информационных символов.

Алгоритм GCCDecoder при $L = 1$ вырождается в традиционный декодер ОКК. Кроме того, процедура оценивания надежности ветвей ДОКК (строки 7, 8 на рис. 9) может быть параметризована. Стоит отметить, что итоговая корректирующая способность, обеспечиваемая GCCDecoder , напрямую зависит от выбора метода списочного декодирования внешних кодов. Например, для коротких линейных кодов в качестве такового могут быть:

1) декодер по максимуму правдоподобия, возвращающий список из кодовых слов с наибольшим правдоподобием;

2) декодер box-and-match [7];

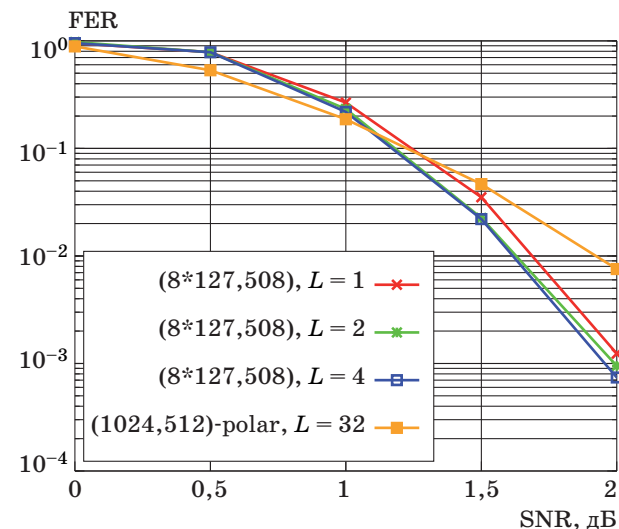
3) декодер порядковых статистик [8].

Вычислительная сложность алгоритма GCCDecoder равна $O(L \cdot l(m \cdot 2^l + \phi(m)))$ при фиксированном размере выхода процедуры ListDecode , где l — это порядок внутреннего полярного кода, m — длина внешнего кода, а $\phi(m)$ — это функция вычислительной сложности декодирования внешних кодов. Таким образом, сложность всего алгоритма существенным образом зависит от сложности списочного декодера внешних кодов.

Для оценки объема используемой памяти уже необходимо знать максимальный размер L_m списка, выдаваемого процедурой ListDecode списочного декодирования внешних кодов. Тогда число ветвей в ДОКК не превосходит величины $L_m \cdot L$. И, следовательно, в целях быстродействия можно использовать массив флагов, указывающих, какие элементы набора из $L_m \cdot L$ ветвей являются активными в текущий момент. Соответственно, число ветвей ДВК никогда не превосходит $2L$. Таким образом, размер требуемой памяти для алгоритма равен $O(L_m \cdot L \cdot l \cdot m)$.

Численные результаты

Согласно процедуре [2], был получен обобщенный каскадный (1016, 508)-код C , состоящий из 8 внешних БЧХ-кодов длины 127 с минимальными расстояниями 0, 30, 43, 8, 47, 9, 12, 3 и внутреннего полярного кода порядка 3. При имитационном моделировании код C был декодирован представленным декодером с размером списка $L = 1, 2, 4$ (рис. 10). В результате был достигнут



■ **Рис. 10.** Зависимость вероятности ошибки декодирования от отношения сигнал/шум

существенный выигрыш по сравнению с алгоритмом Тала — Варди [3] списочного декодирования полярного (1024, 512)-кода, который, как было показано [3], при длине списка $L = 32$ приближается к корректирующей способности по максимуму правдоподобия.

Заключение

В работе представлен алгоритм списочного декодирования обобщенных каскадных кодов с вну-

тренными полярными кодами, позволяющий улучшить корректирующую способность по сравнению с традиционным декодером ОКК.

Данное исследование было выполнено в рамках работ по госконтракту 07.514.11.4101, исполняемому в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса РФ на 2007–2013 годы» при финансовой поддержке Министерства образования и науки Российской Федерации.

Литература

1. **Arikan E.** Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels // *IEEE Transactions on Information Theory*. 2009. N 7(55). P. 3051–3073.
2. **Trifonov P., Semenov P.** Generalized Concatenated Codes Based on Polar codes // *Proc. of IEEE Intern. Symp. on Wireless Communication Systems*. 2011. P. 442–446.
3. **Tal I., Vardy A.** List decoding of polar codes // *Proc. of the IEEE Intern. Symp. on Information Theory (ISIT)*. 2011. P. 1–5.
4. **Семенов П.** Практические аспекты алгоритма Тала — Варди списочного декодирования полярных кодов // *Научно-технические ведомости СПбГПУ*. Информатика. Телекоммуникации. Управление. 2012. № 4(152). С. 40–44.
5. **Зиновьев В. А.** Обобщенные каскадные коды // *Проблемы передачи информации*. 1976. № 1. С. 5–15.
6. **Bossert M.** Channel coding for telecommunications. — Wiley, 1999. — 512 p.
7. **Valembois A., Fossorier M.** Box and match techniques applied to soft-decision decoding // *IEEE Transactions on Information Theory*. 2004. N 5. P. 796–810.
8. **Fossorier M., Lin S.** Soft-decision decoding of linear block codes based on ordered statistics // *IEEE Transactions on Information Theory*. 1995. N 5. P. 1379–1396.