

УДК 621.391.15

## КОНСТРУКЦИЯ ПЛЕТЕННЫХ СВЕРТОЧНЫХ КОДОВ НА БАЗЕ КОДОВ ПРОВЕРКИ НА ЧЕТНОСТЬ С ОДНИМ ПРОВЕРОЧНЫМ СИМВОЛОМ

**К. А. Кондрашов,**

младший научный сотрудник

**В. В. Зяблов,**

доктор техн. наук, профессор

Институт проблем передачи информации им. А. А. Харкевича РАН

Предлагается новая конструкция плетеных сверточных кодов с малой плотностью проверок, разработанная на основе кодов проверки на четность с одним проверочным символом. Использование последних в качестве кодов-компонентов позволяет естественным образом варьировать результирующую скорость кода предложенной конструкции без изменения кодера и декодера и уменьшает сложность декодирования. Для декодирования предлагаются два итеративных алгоритма с жестким принятием решений: мажоритарный алгоритм и мажоритарный алгоритм с введением стираний. Выполняется исследование корректирующих свойств при заданных алгоритмах декодирования.

**Ключевые слова** — сверточные коды, плетеные коды, МПП-коды, коды проверки на четность, кодирование, итеративное декодирование, свободное расстояние, активное строчное расстояние.

### Введение

Блочные коды с малой плотностью проверок на четность (МПП-коды) и итеративный алгоритм их декодирования были предложены Р. Галлагером еще в начале 1960-х [1]. Структура этих кодов потенциально позволяет получать малые вероятности ошибок для кодов с высокой скоростью при низкой сложности декодирования. Коды Галлагера являются каскадными кодами, в которых в качестве кодов-компонентов используются коды проверки на четность с одним проверочным символом. Такие коды-компоненты являются кодами с максимально достижимым кодовым расстоянием (МДР), легко декодируемы и существуют на всех длинах, что позволяет получать МПП-коды с произвольными скоростями.

В данной статье мы рассматриваем сверточные варианты МПП-кодов. Мы предлагаем конструкцию сверточных МПП-кодов [2], в которой в качестве кодов-компонентов также используются коды проверки на четность с одним проверочным символом. Последнее позволяет естественным образом, не изменяя кодер и декодер, варьировать скорость получаемых кодов. Для ко-

дов разработанной конструкции мы также предлагаем два итеративных алгоритма декодирования с жестким принятием решений [3, 4]. Для сравнения корректирующих свойств кодов мы рассматриваем похожую конструкцию сверточных МПП-кодов с меньшим числом кодов-компонентов, разработанную К. Зигангировым, Д. Трухачевым и М. Лентмайером [5]. Рассматриваемые сверточные МПП-коды мы называем плетеными сверточными кодами (П-СМПП-кодами).

Статья организована следующим образом. В первой части мы даем общее определение сверточных МПП-кодов, описываем конструкции П-СМПП-кодов, даем описание общей процедуры кодирования. Затем мы исследуем кодовые расстояния представленных кодов и описываем алгоритмы декодирования. В завершении мы приводим результаты моделирования.

### Сверточные МПП-коды

Пусть  $\mathbf{u} = \mathbf{u}_0\mathbf{u}_1 \dots \mathbf{u}_t \dots \mathbf{u}_t = [u_{t,1}u_{t,2} \dots u_{t,b}]$ ,  $\mathbf{v} = \mathbf{v}_0\mathbf{v}_1 \dots \mathbf{v}_t \dots \mathbf{v}_t = [v_{t,1}v_{t,2} \dots v_{t,c}]$ ,  $u_{t,i}, v_{t,i} \in \mathbb{F}_2$  соответственно информационная и проверочная последовательности сверточного кода со скоростью  $R = b/c$ ,  $b < c$ . Пусть

$$H^T = \begin{pmatrix} \mathbf{H}_0^T(0) & \mathbf{H}_1^T(1) & \mathbf{H}_2^T(2) & \dots & \mathbf{H}_{m_s}^T(m_s) & & & & & & \\ & \mathbf{H}_0^T(1) & \mathbf{H}_1^T(2) & \ddots & \ddots & & & & & & \\ & & \mathbf{H}_0^T(2) & \ddots & \ddots & \mathbf{H}_{m_s}^T(m_s+2) & & & & & \\ & & & \ddots & \ddots & \ddots & & & & & \\ & & & & \mathbf{H}_0^T(m_s) & \dots & \mathbf{H}_{m_s}^T(m_s+m_s) & & & & \\ & & & & \ddots & \ddots & \ddots & & & & \\ & & & & & \mathbf{H}_0^T(t) & \dots & \mathbf{H}_{m_s}^T(m_s+t) & & & \\ & & & & & \ddots & \dots & \ddots & & & \\ & & & & & & \dots & \dots & & & \end{pmatrix} \quad (1)$$

— транспонированная полубесконечная проверочная матрица этого кода, называемая также *формирователем синдрома*. Подматрицы  $\mathbf{H}_i^T(t)$ ,  $i = 0, 1, \dots, m_s$  — двоичные матрицы размера  $c \times (c - b)$ . Величина  $m_s$  называется *памятью* формирователя синдрома. Мы требуем, чтобы выполнялись следующие два условия:

$$\text{rank } \mathbf{H}_0^T(t) = c - b, t \in \mathbb{N};$$

$$\mathbf{H}_{m_s}^T(t) \neq \mathbf{0}, t \in \mathbb{N}, t \geq m_s.$$

Любая кодовая последовательность  $\mathbf{v}$  удовлетворяет уравнению  $\mathbf{vH}^T = \mathbf{0}$  или, в рекуррентной форме:

$$\mathbf{v}_t \mathbf{H}_0^T(t) + \mathbf{v}_{t-1} \mathbf{H}_1^T(t) + \dots + \mathbf{v}_{t-m_s} \mathbf{H}_{m_s}^T(t) = \mathbf{0}, t \in \mathbb{N}. \quad (2)$$

Если строки  $\mathbf{h}_n$  матрицы  $\mathbf{H}^T$  разрежены, т. е.  $\omega_H(\mathbf{h}_n) \ll (c - b)m_s$ , где  $\omega_H(\cdot)$  — вес Хэмминга, то сверточный код является МПП-кодом.

Для любого МПП-кода (для сверточного — с момента времени  $t = m_s$ ) если в каждом столбце проверочной матрицы ровно  $J$  единиц, а в каждой строке ровно  $K$  единиц, то МПП-код называется *регулярным* (далее мы рассматриваем только регулярные МПП-коды). Такой код можно рассматривать как состоящий из  $J$  внутренних кодов-компонентов проверки на четность длины  $K$ . Каждая строка проверочной матрицы в таком случае трактуется как код проверки на четность, образованный из символов кодового слова с номерами позиций, на которых в этой строке стоит 1.

### Конструкции плетеных сверточных МПП-кодов

В работе Галлагера, посвященной блочным МПП-кодам [1], было показано, что МПП-коды с кодами-компонентами с одной проверкой на четность с минимальным расстоянием  $d_i = 2$  обладают хорошими корректирующими свойствами (кодоевое расстояние растет линейно с длиной кода) в том случае, если используется  $J \geq 3$  кодов-компонентов. И хотя мы строим сверточные МПП-коды, мы все же придерживаемся этого требования. Мы предлагаем конструкцию 4-плетеного сверточного МПП-кода (4-П-СМПП) с  $J = 4$  кодами-компонентами [2].

Опишем конструкцию 4-П-СМПП-кода. В 4-П-СМПП-коде каждый символ кодового слова входит в 4 кода-компонента — в «горизонтальный», «вертикальный» и два «диагональных» кода-компонента, а само кодовое слово образуется из «переплетения» кодов-компонентов. Представим кодовое слово 4-П-СМПП-кода в виде полубесконечного массива двоичных символов (рис. 1, а). В строках такого массива хранятся кодовые слова горизонтального кода-компонента, в столбцах — кодовые слова вертикального кода-компонента, а по диагоналям располагаются кодовые слова диагональных кодов-компонентов. На рисунке изображено кодовое слово 4-П-СМПП-кода с длиной кодов-компонентов 8. Индексами обозначены символы, образующие кодовые слова четырех кодов-компонентов в произвольный момент времени  $t$ . На момент времени  $t$  серые ячейки представляют известные, закодированные ранее символы. На вход подается информационный блок  $\mathbf{u}_t$ ,  $\mathbf{u}_t = [u_{t,1}u_{t,2}u_{t,3}u_{t,4}]$ . Выходом  $\mathbf{v}_t$  служит кодовое слово горизонтального кода-компонента. Память формирователя синдрома представленного кода  $m_s = 7$ , скорость  $R = 1/2$ , кодовое ограничение  $(m_s + 1)c = 64$ , а компонентные проверочные матрицы формирователя синдрома соответственно равны

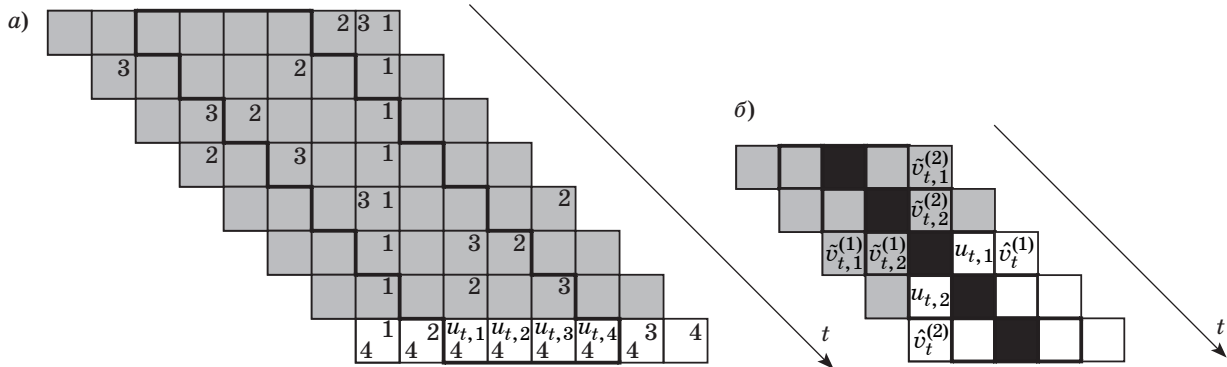
$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Кодирование 4-П-СМПП-кода осуществляется следующим образом. В произвольный момент времени  $t$  параллельно кодируются вертикальный и диагональные коды-компоненты — все их информационные символы уже известны. Затем кодируется горизонтальный код-компонент — полученные от остальных кодов-компонентов проверочные символы вместе с символами кодируемого информационного блока  $u_t$  образуют его информационные символы. Кодовое слово горизонтального кода-компонента формирует выход.

Если из конструкции 4-П-СМПП-кода исключить два диагональных кода-компонента, то получившийся 2-П-СМПП-код будет описываться уже известной конструкцией [5]. В работах [5, 6] в качестве кодов-компонентов 2-П-СМПП-кода рассматривались коды с минимальным расстоянием  $d_i \geq 3$ . В этом случае было доказано, что свободное кодовое расстояние 2-П-СМПП-кода растет линейно с длиной кодового ограничения  $(m_s + 1)c$ . Тем не менее, в рамках поставленной задачи в качестве кодов-компонентов 2-П-СМПП-кода, как и в случае 4-П-СМПП-кода, мы будем использовать коды проверки на четность с  $d_i = 2$ .

Опишем конструкцию 2-П-СМПП-кода со скоростью  $R = 1/2$ . Представим кодовое слово 2-П-СМПП-кода с кодами-компонентами длины 4 в виде полубесконечного массива двоичных символов (рис. 1, б). Конструктивно этот массив разбивается на три «полосы» (на рисунке отделены жирной линией). В нижней полосе хранятся проверочные символы кодовых слов вертикального кода-компонента, в центральной полосе хранятся информационные символы кодируемой последовательности, в верхней полосе — проверочные символы кодовых слов горизонтального кода-компонента. Кодовое слово горизонтального кода-компонента  $v_t^{(1)}$  образуется из символов  $[\tilde{v}_{t,1}^{(1)} \tilde{v}_{t,2}^{(1)} u_{t,1} \hat{v}_t^{(1)}]$ , а символы  $[\tilde{v}_{t,1}^{(2)} \tilde{v}_{t,2}^{(2)} u_{t,2} \hat{v}_t^{(2)}]$  образуют кодовое слово  $v_t^{(2)}$  вертикального кода-компонента. Информационный блок состоит из символов  $u_t = [u_{t,1} u_{t,2}]$ . В любой момент времени  $t$  проверочный символ горизонтального кода-компонента  $\hat{v}_t^{(1)}$  вычисляется с помощью символов информационной последовательности  $u_t$  и закодированного ранее проверочного символа вертикального кода-компонента  $\hat{v}_t^{(2)}$ . Проверочный символ вертикального кода-компонента вычисляется по аналогии с использованием известных символов  $u_t$  и  $\hat{v}_t^{(1)}$ . Закодированный в момент времени  $t$  блок  $v_t = [v_{t,1} v_{t,2} v_{t,3} v_{t,4}]$  состоит из следующих символов:

$$v_{t,i} = \begin{cases} u_{t,i}, & 1 \leq i \leq 2 \\ \hat{v}_t^{i-2}, & 3 \leq i \leq 4 \end{cases}.$$



■ Рис. 1. Представление П-СМПП-кодов в виде полубесконечных массивов: а — 4-П-СМПП-код; б — 2-П-СМПП-код

Память формирователя синдрома представленного кода  $m_s = 3$ , кодовое ограничение  $(m_s + 1) \times c = 16$ , а компонентные проверочные матрицы формирователя синдрома соответственно равны

$$\left( \begin{array}{cc} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{array} \right), \left( \begin{array}{cc} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{array} \right), \left( \begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{array} \right), \left( \begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{array} \right).$$

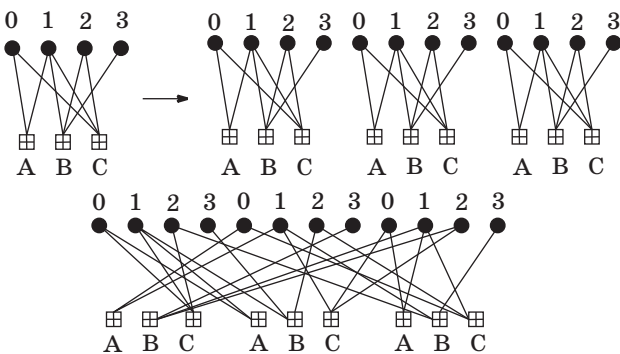
$\mathbf{H}_0 \quad \mathbf{H}_1 \quad \mathbf{H}_2 \quad \mathbf{H}_3$

**Построение ансамбля плетеных сверточных кодов**

Для построения ансамбля сверточных плетеных МПП-кодов мы воспользуемся методикой, предложенной в работе [7]. МПП-коды можно представить в виде графа Таннера [8] — двудольного графа, где один набор вершин соответствует символам МПП-кода (*символьные вершины*), а второй — проверкам кодов-компонентов МПП-кода (*проверочные вершины*). Выполняя различные преобразования над исходным графом МПП-кода, можно получить новые МПП-коды. Так, для построения ансамбля П-СМПП-кодов  $\mathcal{C}(L)$  мы будем использовать выбираемые равновероятно случайные матрицы перестановок  $\mathbf{P}$  размера  $L \times L$ . Применим к исходному графу  $\mathbf{V}$  операцию «копирование с перестановками» [7]:  $b_{i,j} = b_{i,j}P_{i,j}$ . Каждому ребру графа  $\mathbf{V}$  ставится в соответствие матрица перестановки, все узлы графа копируются  $L$  раз, а конечные точки ребер переставляются. В качестве примера на рис. 2 показано преобразование с коэффициентом  $L = 3$  графа

$$\mathbf{V} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Преобразование увеличивает в  $L$  раз длину кода  $N$ ,  $c$ ,  $b$  и кодовое ограничение.



■ Рис. 2. Копирование протографа с перестановками

**Кодирование плетеных сверточных кодов**

Кодировать П-СМПП-коды можно несколькими способами. Первый из них уже был описан при построении конструкций. Однако он не является оптимальным [9]. С аналогичной сложностью, но с меньшим требованием к памяти можно кодировать П-СМПП-коды с помощью *частичных синдромов*.

Опишем алгоритм кодирования. В любой момент времени  $t$  кодовая последовательность  $\mathbf{v}$  удовлетворяет условию

$$\mathbf{v}_{[0,t-1]} \mathbf{H}_{[0,t+m_s-1]}^T = [\mathbf{0}_{[0,t-1]} | \mathbf{s}_t], \quad (3)$$

где  $\mathbf{s}_t = [s_{t,1} s_{t,2} \dots s_{t,m}]$  — вектор *частичных синдромов*. На самом деле, вектор *частичных синдромов*  $\mathbf{s}_t$  — это не что иное, как состояние  $\sigma_t$  кодера сверточного кода в момент времени  $t$ . Вектор *частичных синдромов* вычисляется рекуррентно в соответствии со следующим правилом:

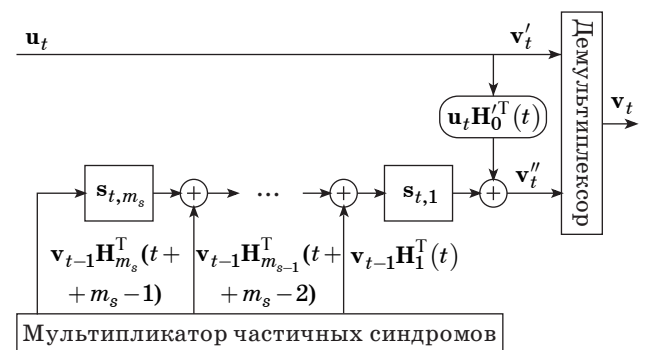
$$\mathbf{s}_{t,i} = \begin{cases} \mathbf{s}_{t-1,i+1} + \mathbf{v}_{t-1} \mathbf{H}_i^T(t+i-1), & i=1, \dots, m_s-1 \\ \mathbf{v}_{t-1} \mathbf{H}_{m_s}^T(t+m_s-1), & i=m_s \end{cases}. \quad (4)$$

Пусть, без потери общности, символы информационного блока  $\mathbf{u}_t$  стоят на первых  $b$  позициях кодового блока  $\mathbf{v}_t$ . Пусть  $\mathbf{v}_t = [\mathbf{v}'_t | \mathbf{v}''_t]$ ,  $\mathbf{v}'_t = \mathbf{u}_t$ . Напомним, что подматрица  $\mathbf{H}_0^T(t)$ ,  $t \in \mathbb{N}$  имеет полный ранг  $c - b$ . Тогда кодовый блок  $\mathbf{v}_t = [\mathbf{v}'_t | \mathbf{v}''_t]$  находится из решения

$$-[\mathbf{u}_t | \mathbf{v}''_t] \mathbf{H}_0^T(t) = \mathbf{s}_{t,1}. \quad (5)$$

Необходимый для кодирования таким способом объем памяти составляет  $(c - b)m_s$  бит. Сложность кодирования линейна относительно длины  $N$  П-СМПП-кода.

При аппаратной реализации кодирования для вычисления *частичных синдромов* можно использовать сдвиговой регистр (рис. 3). В этом



■ Рис. 3. Схема сверточного кодера на основе *частичных синдромов*

случае можно построить кодер, позволяющий «переключать» скорость сверточного кода. Построим кодер на сдвиговом регистре, рассчитанный на максимальную скорость П-СМПП-кода и, соответственно, максимальные значения  $m_s$  и  $s$ . Тогда при уменьшении разрядности сумматоров в сдвиговом регистре и его памяти, что соответствует уменьшению длины кодов-компонентов и памяти формирователя синдрома, мы будем получать П-СМПП-код с меньшей скоростью.

### Оценка кодового расстояния

В отличие от блочных кодов, сверточные коды имеют кодовые слова различной длины. Поэтому использовать определенное для блочных кодов и подразумевающее сравнение кодовых слов одной длины *минимальное кодовое расстояние* — минимальное хэммингово расстояние между любыми двумя кодовыми словами — для них нельзя. Вместо этого для сверточных кодов вводится аналог минимального кодового расстояния — свободное расстояние.

*Определение 1.* Минимальное расстояние между любыми различными кодовыми последовательностями сверточного кода называется свободным расстоянием:

$$d_{free} = \min_{\mathbf{v} \neq \mathbf{v}'} \{d_H(\mathbf{v}, \mathbf{v}')\}.$$

Математический аппарат для исследования сверточных кодов не так хорошо развит, как для блочных. В общем случае теоретическая оценка свободного расстояния сверточного кода является сложной исследовательской задачей, сильно зависящей от используемой кодовой конструкции. Однако часть точных значений характеристик сверточного кода можно получить методом компьютерного моделирования. Для получения оценки свободного кодового расстояния мы будем исследовать вспомогательную величину — активное строчное расстояние.

*Определение 2.* Активным строчным расстоянием  $d_j$  сверточного кода называется минимальный вес терминированного кодового слова  $\mathbf{v}_{[1, j]}$ , не проводящего кодер через два последовательных нулевых состояния:

$$d_j = \min \left\{ \omega_H(\mathbf{v}_{[1, j]}) : \mathbf{v}_{[1, j]} \mathbf{H}_{[1, j+m_s-1]}^T = \mathbf{0} \right\}.$$

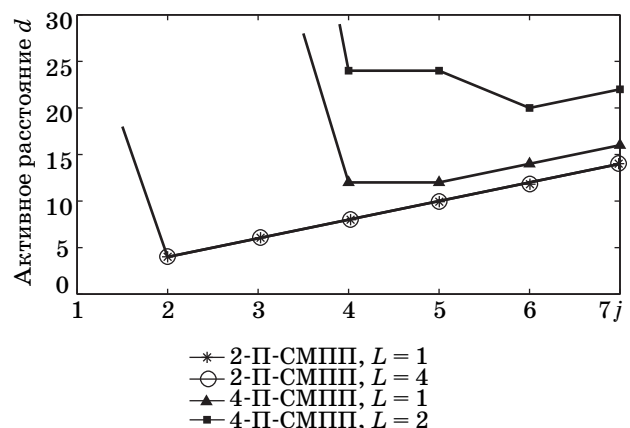
Свободное расстояние связано с активными расстояниями следующим отношением:

$$d_{free} = \min_j \{d_j\}. \quad (6)$$

Для нахождения активных расстояний  $d_j$  мы будем решать для различных длин  $j$  систему линейных уравнений

$$\mathbf{x} \mathbf{H}_{[1, j+m_s-1]}^T = \mathbf{0}, \quad (7)$$

из которой исключены первые несколько строк, отвечающие нулевым проверочным символам вертикального и диагональных кодов-компонентов в первом кодовом блоке. Система (7) имеет или единственное нулевое решение, или множество решений. В последнем случае найденные линейно независимые решения системы (7) образуют фундаментальную систему решений (ФСР), линейная оболочка которой дает все решения системы (7). Найдем среди векторов линейной оболочки ФСР ненулевой вектор с минимальным весом. Его вес даст точное значение активного расстояния  $d_j$ . Однако таким способом можно рассчитать лишь ограниченное число начальных активных расстояний. Сложность перебора векторов линейной оболочки растет экспоненциально с числом найденных решений, поэтому, по мере увеличения  $j$ , которое также увеличивает размерность ФСР, мы быстро сталкиваемся с вычислительным ограничением. Результаты моделирования по вычислению активных расстояний рассмотренных П-СМПП-кодов представлены на рис. 4. Больше число кодов-компонентов и возросшая память повлияли на увеличение активных и свободных расстояний 4-П-СМПП-кодов по сравнению с 2-П-СМПП-кодами. Расширение 4-П-СМПП-кодов при  $L > 1$  также увеличивает активные расстояния. Однако при расширении 2-П-СМПП-кода расстояния не изменились (на рис. 4 активные расстояния 2-П-СМПП-кодов с  $L = 1$  и  $L = 4$  совпадают). Отметим, что на графиках для некоторого  $j_{max}$ ,  $j < j_{max}$  для каждого кода наблюдаются значения  $d_j = \infty$ . Это означает, что при данных значениях  $j$  система (7) не имеет никаких решений, кроме тривиального. Следовательно, для рассматриваемых П-СМПП-кодов не существует пакетов ошибок с длиной до  $j_{max}$ .



■ Рис. 4. Активные расстояния П-СМПП-кодов

### Декодирование плетеных сверточных кодов

Для представленных П-СМПП-кодов мы предлагаем два итеративных алгоритма декодирования с жестким принятием решения [3, 4]: мажоритарный алгоритм  $\mathcal{A}_1$  и его расширенный вариант  $\mathcal{A}_2$  с введением стираний.

Для начала опишем обобщенную процедуру декодирования, которую можно применять к П-СМПП-кодам с произвольными кодами-компонентами. Введем необходимые обозначения. Пусть  $\mathbf{r}$  — принятое из канала слово, содержащее ошибки. На произвольной итерации  $i$ ,  $i \in \mathbb{N}$  на вход декодера подается слово  $\mathbf{r}^{(i)}$ , где  $\mathbf{r}^{(1)} = \mathbf{r}$ , выходом декодера является слово  $\mathbf{r}^{(i+1)}$ . Для декодирования слов, относящихся к определенным кодам-компонентам, используются соответствующие компонентные декодеры  $\{D^{(k)}\}_{k=1}^J$  или  $\{E^{(k)}\}_{k=1}^J$ , если компонентные декодеры способны исправлять как ошибки, так и стирания.

Каждая итерация декодирования обобщенным алгоритмом состоит из двух частей: *внутреннего* декодирования — декодирования слов кодов-компонентов и *внешнего* декодирования — принятия решения по каждому символу слова П-СМПП-кода. При внутреннем декодировании слов кодов-компонентов никакие символы декодируемого слова не заменяются, вместо этого значения символов, полученные от внутренних декодеров, запоминаются в памяти. В результате внутреннего декодирования для каждого символа слова  $\mathbf{r}^{(i)}$  в памяти хранится  $J$  решений — по решению от  $J$  декодеров кодов-компонентов. При внешнем декодировании для каждого символа слова  $\mathbf{r}^{(i)}$  на основании этих  $J$  значений принимается решение об изменении. Результатом становится слово  $\mathbf{r}^{(i+1)}$ .

### Обобщенный алгоритм декодирования

**Внутреннее декодирование.** Для каждого кода-компонента  $k$  с помощью декодера компонента  $D^{(k)}$  декодируются все соответствующие ему слова из  $\mathbf{r}^{(i)}$ . Результаты запоминаются в  $\mathbf{r}_k^{(i)}$ .

**Внешнее декодирование.** Для каждого символа  $r_j^{(i)}$  входного слова  $\mathbf{r}^{(i)}$  значение  $r_j^{(i)}$  вместе со значениями  $\{r_{j,k}^{(i)}\}_{k=1}^J$ , полученными для этого символа при внутреннем декодировании, подается на вход функции *голосования*. Эта мажоритарная функция возвращает  $r_j^{(i+1)}$  со значением, которое встречалось среди ее входных аргументов чаще других. Из всех  $r_j^{(i+1)}$  формируется слово следующей итерации  $\mathbf{r}^{(i+1)}$ .

**Критерии останова.** Логическим критерием завершения декодирования является нулевое значение синдрома, полученного после некоторой ите-

рации  $i$  слова  $\mathbf{r}^{(i+1)}$ . Однако изменение синдрома с каждой итерацией может носить произвольный характер. Поэтому для гарантии останова алгоритма мы вводим ограничение на число итераций. Таким образом, возможны следующие результаты декодирования: успех декодирования; ошибка декодирования; отказ от декодирования.

**Успех декодирования.** Успех декодирования происходит в том случае, если синдром выходного слова  $\mathbf{r}^{(i+1)}$  нулевой и декодированное кодовое слово совпадает с переданным:  $S(\mathbf{r}^{(i+1)}) = \mathbf{0}$ ,  $\mathbf{r}^{(i+1)} = \mathbf{v}$ .

**Ошибка декодирования.** Происходит при нулевом синдроме, когда принятое слово декодировалось в другое кодовое слово, не совпадающее с переданным:  $S(\mathbf{r}^{(i+1)}) = \mathbf{0}$ ,  $\mathbf{r}^{(i+1)} \neq \mathbf{v}$ .

**Отказ от декодирования.** Отказ происходит при ненулевом синдроме, если достигнут предел итераций или результирующее слово итерации  $\mathbf{r}^{(i+1)}$  не отличается от входного слова итерации  $\mathbf{r}^{(i)}$ :  $S(\mathbf{r}^{(i+1)}) \neq \mathbf{0}$ ,  $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} \vee i = I_{\max}$ .

Алгоритм декодирования П-СМПП-кодов с кодами-компонентами с одним проверочным символом  $\mathcal{A}_1$  получается из обобщенного алгоритма декодирования выбором подходящих декодеров для внутреннего декодирования. Сами по себе коды с одним проверочным символом способны лишь обнаруживать ошибки, но не исправлять их. Поэтому внутреннее декодирование несколько отличается от того, что обычно под этим понимается. В алгоритме  $\mathcal{A}_1$  при внутреннем декодировании для каждого слова кода-компонента  $k$  декодер  $D^{(k)}$  изменяет каждый символ слова так, чтобы с учетом этого изменения декодируемое слово удовлетворяло проверке на четность.

Для двоичных П-СМПП-кодов с четным числом кодов-компонентов во время процедуры голосования возможна ситуация, когда декодированные

значения  $\{r_{j,k}^{(i)}\}_{k=1}^J$  символа  $r_j^{(i)}$  делятся поровну.

В этом случае логично подозревать символ  $r_j^{(i)}$  на ошибку, но алгоритм  $\mathcal{A}_1$  не приводит к его изменению. На поздних итерациях, когда большинство ошибок исправлено (или внесены новые устойчивые ошибки), невозможность изменить спорный символ приводит с большой вероятностью к отказу декодирования, так как декодируемое слово не изменяется. Чтобы иметь возможность продолжить декодирование, мы предлагаем ассоциировать с символом новую качественную характеристику и считать такой «ненадежный» символ стертým.

**Определение 3.** Стирание — качественная характеристика символа, означающая, что его значение не определено. При отсутствии ошибок код с минимальным расстоянием  $d$  способен исправить до  $d - 1$  стираний.

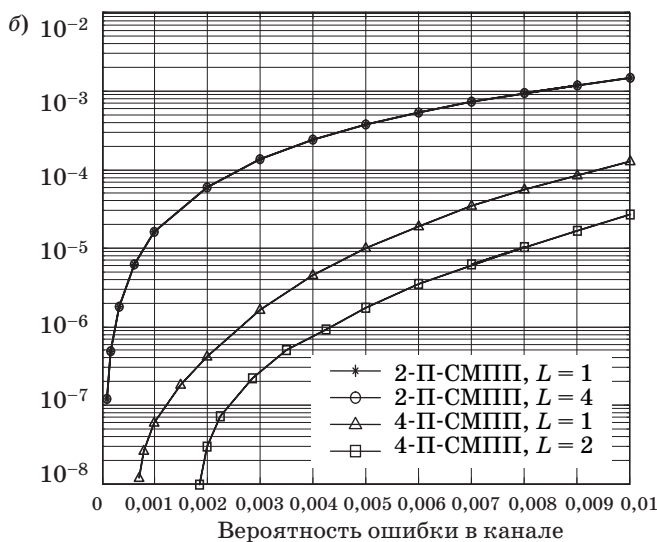
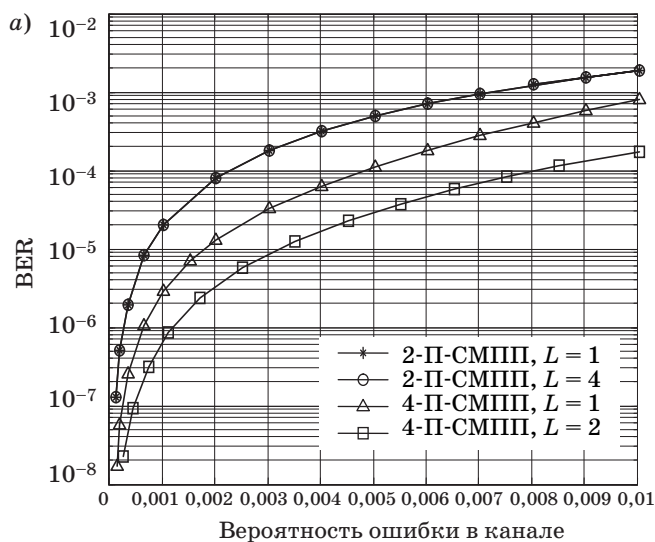
Алгоритм  $\mathcal{A}_2$  также построен на основе обобщенного алгоритма декодирования, но он работа-

ет с учетом стираний. В алгоритме декодирования  $A_2$  при голосовании спорные символы объявляются стертými. На внутреннем декодировании используются компонентные декодеры  $\{E^{(k)}\}_{k=1}^J$ , исправляющие стирания. Если декодируемое слово кода-компонента  $k$  не содержит стираний, то  $E^{(k)}$  работает как  $D^{(k)}$ . Если слово содержит одно стирание, то значение стертого символа заменяется с учетом выполнения проверки на четность и стирание снимается. Если стираний в слове больше одного, то слово не изменяется. При внешнем декодировании значения стертых символов в голосовании не участвуют, а решение для символа принимается, если число совпавших декодированных значений этого символа больше половины числа кодовых компонентов. При исправлении стираний новые

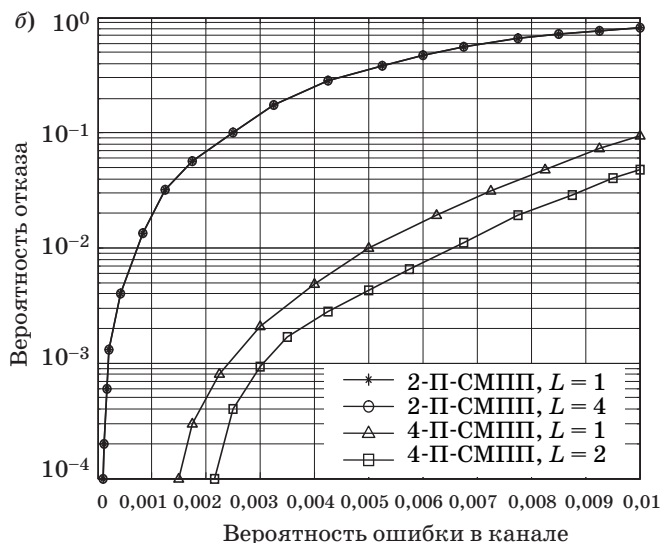
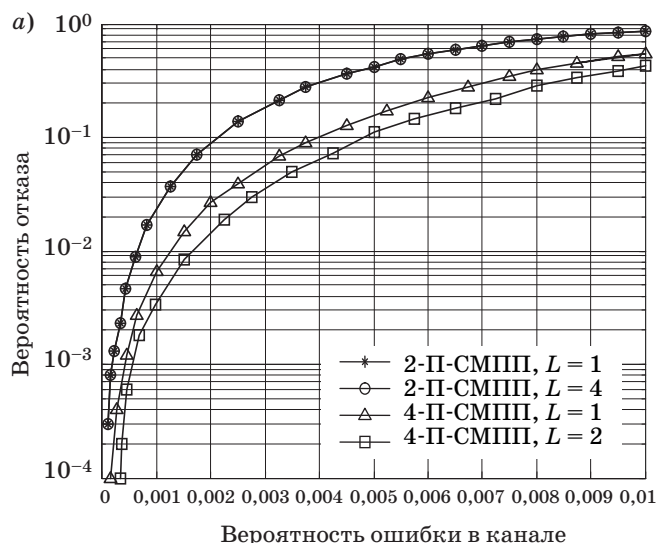
стирания не образуются, а значит, стирания либо будут полностью декодированы, либо будет получена неисправимая комбинация стираний. В последнем случае мы снимаем с оставшихся символов значение стирания: продолжается декодирование с исправлением ошибок. Критерии останова и возможные значения на выходе алгоритма  $A_2$  аналогичны критериям и значениям алгоритма  $A_1$ .

### Декодирование итеративным алгоритмом

Моделирование проводилось для 2-П-СМПП-кода длины 2400 бит с кодами-компонентами (4,3)-кодами проверки на четность и 4-П-СМПП-кода длины 2400 с кодами-компонентами (8,7)-кодами проверки на четность. Выходные вероятности ошибки на бит (Bit-Error-Rate — BER) и ве-



■ Рис. 5. Результаты декодирования П-СМПП-кодов: а — алгоритм декодирования  $A_1$ ; б — алгоритм декодирования  $A_2$



■ Рис. 6. Вероятность отказа декодирования: а — алгоритм декодирования  $A_1$ ; б — алгоритм декодирования  $A_2$

роятности отказа декодирования представлены на рис. 5, 6 соответственно.

Обладая большим кодовым расстоянием, 4-П-СМПП-код исправляет больше ошибок, чем 2-П-СМПП-код. Расширение 4-П-СМПП-кода с  $L > 1$  приводит к дальнейшему улучшению результата. Расширение 2-П-СМПП-кода, как и в случае с расстояниями, не повлияло на результаты декодирования (результаты на рис. 4, 5 совпадают). Отметим, что при декодировании алгоритмом  $A_2$  как 4-ПСММ-коды, так и 2-П-СМПП-коды показывали результаты лучше, чем при декодировании по алгоритму  $A_1$ .

### Заключение

В статье была представлена новая конструкция плетеных сверточных МПП-кодов с четырь-

мя кодами-компонентами с одной проверкой на четность. Разработаны два итеративных алгоритма декодирования с жестким принятием решений. Исследованы корректирующие свойства представленных П-СМПП-кодов. Для сравнения была выбрана кодовая конструкция с меньшим числом кодов-компонентов. Эта конструкция аналогична конструкции [5], однако ранее использование кодов с одной проверкой на четность в данной конструкции не исследовалось.

Использование кодов с одной проверкой на четность в 2- и 4-П-СМПП-кодах позволяет адаптивно изменять их скорость без изменения кодеров и декодеров, а также уменьшает сложность декодирования. Увеличение числа кодов компонентов при сохранении сложности декодирования приводит к улучшению кодовых характеристик и результатов декодирования.

### Литература

1. Галлагер Р. Дж. Коды с малой плотностью проверок на четность. — М.: Мир, 1966. — 144 с.
2. Кондрашов К. А., Зяблов В. В. Две конструкции сверточных МПП-кодов // Информационные технологии и системы (ИТиС): сб. тр. Бекасово, 2009. С. 15–18.
3. Kondrashov K., Zyablov V. Binary LDPC Woven Convolutional Codes Decoding // Algebraic and Combinatorial Coding Theory (ACCT 2010), Sep. 2010, Akademgorodok, Novosibirsk, RUSSIA. P. 323–328.
4. Кондрашов К. А., Зяблов В. В. Декодирование Q-ных плетеных сверточных МПП-кодов // Информационные технологии и системы (ИТиС): сб. тр. Геленджик, 2010. С. 85–88.
5. Zigangirov K. Sh., Truhachev D., Lentmaier M. On braided block codes // Proc. IEEE Int. Symp. on Inf. Th., ISIT 2003, Yokohama, Japan, 29 June – 4 July 2003. P. 32.
6. Lentmaier M. et al. Convolutional Codes for Iterative Decoding // Spread Spectrum Techniques and Applications, ISSSTA'08 // IEEE 10<sup>th</sup> Intern. Symp. Aug. 2008. P. 785–789.
7. Thorpe J. Low Density Parity Check (LDPC) Codes Constructed from Protographs // JPL IPN Progress Report 42–154. 2003. P. 1–7.
8. Tanner R. M. A Recursive Approach to Low Complexity Codes // IEEE Trans. Inform. Theory. 1981. Vol. 27. N 5. P. 533–547.
9. Pusane A. E. et al. Implementation aspects of LDPC convolutional codes // Communications, IEEE Transactions. 2008. Vol. 56. N 7. P. 1060–1069.