

УДК 621.391.251

## МНОГОПороговое декодирование кодов с низкой плотностью проверок на четность

**А. В. Белоголовый,**

канд. техн. наук, ассистент

**Е. А. Крук,**

доктор техн. наук, профессор

Санкт-Петербургский государственный университет аэрокосмического приборостроения

*В статье предлагается новый алгоритм декодирования. Новый алгоритм принадлежит к классу быстрых алгоритмов, но благодаря использованию порогов при декодировании качество работы приближается к результатам медленных декодеров. В работе приводится описание алгоритма и сравнение его с традиционными и быстрыми декодерами на основе моделирования.*

*In paper the new decoding algorithm is proposed. The new algorithm belongs to fast decoding algorithms class, but due to the usage of threshold in decoding the performance becomes closer to slow decoders results. In paper there are shown the description of algorithm and the comparison with traditional and fast decoders based on simulations.*

### Коды с низкой плотностью проверок на четность

Коды с низкой плотностью проверок на четность были впервые предложены Р. Галлагером в 1963 году как высокоскоростные коды с малой вероятностью ошибки для передачи по каналам с шумом [1, 2].

Коды с низкой плотностью проверок на четность – это линейные коды, определяемые проверочной матрицей, содержащей в основном нулевые элементы и относительно небольшое количество ненулевых. Для случая двоичных кодов с низкой плотностью проверок на четность проверочная матрица состоит из небольшого числа единиц и нулей на всех остальных позициях матрицы. Пример проверочной матрицы с низкой плотностью проверок на четность:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Для всего класса кодов с низкой плотностью проверок на четность [1–3] существуют декодеры, основывающиеся только на свойствах разреженности проверочной матрицы и не использующие никакие свойства самого пространства кодовых слов. Эти алгоритмы декодирования имеют низкую сложность и позволяют достичь малой вероятности ошибки при передаче в канале с аддитивным нормально распределенным (гауссовским) шумом [1–7].

### Традиционное декодирование кодов с низкой плотностью проверок на четность

Рассмотрим итеративное декодирование кодов с низкой плотностью проверок на четность в двоичном симметричном канале и канале с аддитивным белым гауссовским шумом.

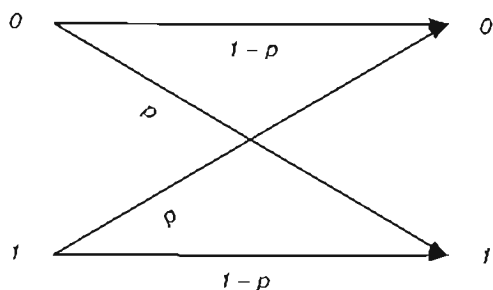
**Симметричные каналы с двоичным входом.** Симметричным каналом с двоичным входом будем называть дискретный по времени канал со следующими свойствами:

входной алфавит  $X$  состоит из двух символов, обозначаемых 0 и 1;

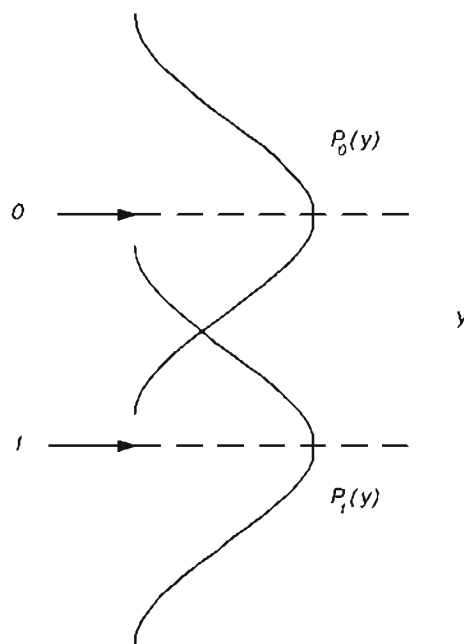
выходной алфавит  $Y$  может быть представлен как дискретное либо непрерывное вещественное множество чисел;

выход  $y$  в заданный отсчет времени зависит только от одного входного символа  $x$ ;

а)



б)



■ Рис. 1. Симметричные каналы с двоичным входом: а – двоичный симметричный канал; б – канал с аддитивным гауссовским шумом

для всех выходов  $y$  выполняется свойство симметрии:  $P_0(y) = P_1(-y)$ . Здесь под  $P_x(y)$  понимается плотность распределения вероятности, если  $y$  – непрерывный выход, и условная вероятность, если  $Y$  – дискретное множество.

На рис. 1 приведены примеры симметричных каналов с двоичным входом.

Под вектором ошибки для кодового слова  $c = \{c_0, \dots, c_{N-1}\}$  длины  $N$  будем понимать вектор  $e = \{e_0, \dots, e_{N-1}\}$  такой же длины, соответствующие элементы которого равны разностям между вектором  $y = \{y_0, \dots, y_{N-1}\}$ , принятым из канала, и передаваемым кодовым словом:  $e_i = y_i - c_i, 0 \leq i \leq N$ . Эта запись равносильна представлению принятого вектора как суммы кодового слова и ошибки:  $y_i = c_i + e_i, 0 \leq i \leq N$ .

**Декодирование ошибок канала.** Декодирование – это процедура поиска и исправления ошибки, наложенной каналом на кодовое слово, по принятому из канала вектору или собственно поиск кодового слова по вектору, принятому из канала.

Декодирование по максимуму правдоподобия кода  $C$  обозначает нахождение по заданному принятому вектору  $y$  такого кодового слова  $c \in C$ , которое максимизирует вероятность того, что передавалось слово  $c$  при условии принятия вектора  $y: p(c/y) \rightarrow \max$  [51–54, 56, 60]. Задача декодирования по максимуму правдоподобия является NP-полной.

Для оценки качества работы различных декодеров используется оценка вероятности ошибки декодирования на информационный бит, вычисляемая как отношение количества ошибочных информационных бит после декодирования к общему количеству переданных информационных бит:

$$BER = \frac{N_{\text{ошибок}}}{N_{\text{всех бит}}}$$

Итеративные схемы декодирования кодов с низкой плотностью проверок на четность не являются декодерами по максимуму правдоподобия, но позволяют получить разумный баланс по сложности и вероятности ошибки декодирования по сравнению с декодированием по максимуму правдоподобия. Итеративное декодирование подразумевает, что нахождение кодового слова будет производиться не за один проход, а за несколько, с последовательным уточнением результата на каждом шаге [1–3, 7].

**«Жесткое» декодирование** – это схема декодирования для двоичного симметричного канала при небольшом количестве ошибок в канале. «Жесткое» декодирование инвертированием битов – самая простая по сложности схема декодирования кодов с низкой плотностью проверок на четность [1, 2].

Как уже говорилось, под проверкой понимается любая строка  $h = \{h_0, \dots, h_{N-1}\}$  из проверочной матрицы кода с низкой плотностью проверок на четность. Будем говорить, что проверка для некоего вектора  $y = \{y_0, \dots, y_{N-1}\}$  выполняется тогда, когда скалярное произведение вектора  $y$  на проверку дает ноль. Будем говорить, что элемент  $y_i$  принятого вектора  $y$  участвует в проверке  $h = \{h_0, \dots, h_{N-1}\}$  тогда, когда соответствующий элемент проверки  $h_i$  не равен нулю.

Одна итерация «жесткого» декодирования инвертированием битов производится следующим образом.

1. Для принятого вектора вычисляются все проверки.

2. Если некоторый бит принятого вектора участвовал более чем в половине невыполнившихся проверок, бит инвертируется.

3. После такого анализа всех символов принятого вектора вектор проверяется на принадлежность коду. Если вектор является кодовым словом, декодирование заканчивается, в противном случае выполняется следующая итерация алгоритма.

Такая процедура декодирования применима для кодов с низкой плотностью проверок на четность потому, что большинство проверок в таком случае будут содержать одну ошибку или не будут содержать ошибок вообще, и тогда нарушение большого количества проверок для символа принятого слова будет обозначать наличие в нем ошибки.

Сложность одной итерации «жесткого» декодирования инвертированием бит является линейной, количество итераций декодирования обычно выбирается около  $\log_2(N)$ , где  $N$  – длина кодового слова.

**Декодирование по вероятностям** является «мягким» декодированием, т. е. декодированием на основе вектора, состоящего не из дискретных значений (0 и 1), а из вещественных величин, полученных на выходе канала [1–3, 6, 7], путем пересчета вероятностей (англ. – «belief propagation decoding»).

Будем называть множеством  $N(m)$  множество символов принятого вектора, которые входят в  $m$ -ю проверку:  $N(m) = \{n : H_{m,n} = 1\}$ , где  $H$  – проверочная матрица кода с низкой плотностью проверок на четность. Будем называть множеством  $M(n)$  множество проверок, в которых участвует  $n$ -й символ принятого вектора:  $M(n) = \{m : H_{m,n} = 1\}$ .

На основе принятого из канала вектора формируются два (для двоичного случая) вектора вероятностей того, что в принятом векторе на данной позиции находился заданный символ:

$$f^0 = (f_0^0, \dots, f_{N-1}^0), \quad f_i^0 = P(y_i = 0)$$

и  $f^1 = (f_0^1, \dots, f_{N-1}^1), \quad f_i^1 = P(y_i = 1),$

где  $y_i$  – элементы принятого вектора. Каждому ненулевому элементу проверочной матрицы кода с низкой плотностью проверок на четность приписывается две величины:  $q_{i,j}^x$  и  $r_{i,j}^x$ . Величина  $q_{i,j}^x$  является вероятностью того, что  $j$ -й символ принятого вектора имеет значение  $x$  по информации, полученной из всех проверок, кроме  $i$ -й. Величина  $r_{i,j}^x$  является вероятностью того, что проверка  $i$  выполняется, если  $j$ -й символ принятого вектора равен  $x$ , а все остальные символы проверок имеют распределение вероятностей, заданное величинами  $\{q_{i,j}^x : j \in N(i) \setminus j\}$ .

Перед началом работы алгоритму требуется инициализация, заключающаяся в том, что значения  $q_{i,j}^0$  и  $q_{i,j}^1$  принимаются равными  $f_j^0$  и  $f_j^1$  соответственно, а далее алгоритм работает по принципу пересчета вероятностей символов принятого вектора (belief propagation), используя для пересчета вероятностей правило Байеса для апостериорной вероятности события. Одна итерация алгоритма представляет собой следующую последовательность действий.

1. Для всех проверок вычисляются величины

$$\Delta r_{i,j} = \prod_{j' \in N(i) \setminus j} (q_{i,j'}^0 - q_{i,j'}^1) \quad \text{и пересчитываются вероятности } r_{i,j}^x = \frac{1 + (-1)^x \cdot \Delta r_{i,j}}{2} \quad \text{для } x = \{0, 1\}.$$

2. Для всех символов принятого вектора пере-

считываются вероятности  $q_{i,j}^x = \alpha_{i,j} \cdot f_j^x \cdot \prod_{i' \in M(j) \setminus i} r_{i',j}^x$  для  $x = \{0, 1\}$ , где  $\alpha_{i,j}$  – нормирующие коэффициенты, обеспечивающие равенство  $q_{i,j}^0 + q_{i,j}^1 = 1$ .

3. Формируются векторы псевдоапостериорной вероятности  $q_j^0$  и  $q_j^1$  следующим образом:

$$q_j^x = \alpha_j \cdot f_j^x \cdot \prod_{i \in M(j)} r_{i,j}^x, \quad \text{где } \alpha_j \text{ – нормирующие коэффициенты, обеспечивающие равенство } q_j^0 + q_j^1 = 1.$$

4. Формируется вектор решения  $c^*$  по следующему правилу:  $c_j^* = \begin{cases} 1 & \text{при } q_j^1 > 1/2 \\ 0 & \text{иначе} \end{cases}$ . Если вектор  $c^*$

является кодовым словом, декодирование заканчивается, в противном случае выполняется следующая итерация алгоритма.

Сложность данного алгоритма выше, чем сложность «жесткого» декодирования инвертированием битов, однако качество декодирования повышается за счет использования дополнительной информации на выходе канала. Однако качество работы такого алгоритма зависит от инициализации: чем точнее она произведена, тем точнее будет конечный результат. Для канала с гауссовским шумом инициализация может быть произведена при помощи информации о дисперсии шума в канале. Для других распределений шума в канале или при неизвестных характеристиках шума точная инициализация алгоритма может являться сложной задачей.

### Быстрое декодирование LDPC

Несмотря на то, что декодирование пересчетом вероятностей является эффективным методом для каналов с непрерывным выходом, тот факт, что сложность его значительно выше, чем сложность «жесткого» декодирования, оставляет место для поиска более быстрых алгоритмов декодирования, обладающих приемлемым качеством.

Среди известных алгоритмов быстрого декодирования кодов с низкой плотностью проверок на четность для каналов с непрерывным выходом наиболее известен алгоритм «min-sum», являющийся упрощением декодера «belief propagation», а также алгоритм «UMP» (Uniformly Most Powerful [7]), который рассматривается ниже.

Пусть из канала был принят вещественный («мягкий») вектор  $y = (y_0, \dots, y_{N-1})$ . Будем считать надежностью  $i$ -го символа вектора  $y$  величину  $r_i$ , характеризующую удаление принятого значения  $y_i$  от

некоего порогового значения  $\text{thr}$ , при котором все значения переданных символов равновероятны:  $\text{thr} : P(1/y_j) = P(0/y_j)$ ,  $r_j = |y_j - \text{thr}|$ . Для случая двоичной модуляции, при которой все биты кодового слова отображаются во множество  $\{-1, +1\}$  и передаются по симметричному каналу с двоичным входом, надежность принятого символа  $y$  будет, таким образом, равна абсолютной величине принятого значения:  $r_j = |y_j|$ .

Быстрое декодирование по надежностям является собой баланс между «жестким» декодированием и декодированием по вероятностям [1–3]. При декодировании используется вектор «жестких» решений  $x = \{x_0, \dots, x_{N-1}\}$  (вектор, состоящий из 0 и 1 и представляющий собой набор дискретных решений) и вектор надежностей  $r = \{r_0, \dots, r_{N-1}\}$  (вектор, состоящий из вещественных величин из интервала  $[0, +\infty)$ ). По аналогии с вероятностным декодированием каждому ненулевому символу проверочной матрицы приписываются два числа:  $x_{i,j}$  и  $r_{i,j}$ , представляющие собой «жесткое» решение относительно символа  $i$ , полученное при помощи всех проверок, кроме  $i$ -й, и надежность символов  $i$ -й проверки без учета  $j$ -го символа соответственно. Декодирование также является итеративным, одна итерация представляет собой следующую последовательность действий.

1. Для всех проверок и всех символов проверок вычисляется надежность проверок относительно символа  $j$ , представляющая собой минимальное значение надежностей всех символов, входящих в проверку, кроме самого символа  $j$ :

$$\hat{r}_{i,j} = \min_{j' \in N(i) \setminus j} r_{i,j'}$$

2. Для всех символов принятого вектора и всех проверок производится пересчет надежностей символов проверок: если проверка, в которую входил некоторый символ, не выполняется, то надежность  $r_{i,j}$  уменьшается на величину надежности проверки  $\hat{r}_{i,j}$ , в противном случае надежность  $r_{i,j}$  увеличивается на  $\hat{r}_{i,j}$ ; учитываются все проверки, кроме  $i$ -й. Если после применения всех проверок для какого-то символа  $j$  величина  $r_{i,j}$  стала меньше нуля, то обновленная надежность  $r_{i,j}$  берется по модулю, а жесткое решение  $x_{i,j}$  инвертируется

3. Аналогично пересчитываются надежности принятого вектора: если  $i$ -я проверка, в которую входит  $j$ -й символ, нарушается, то из надежности  $r_j$  вычитается надежность  $i$ -й проверки, иначе надежность символа увеличивается на надежность проверки. Если после применения всех проверок для какого-то символа  $j$  величина  $r_j$  стала меньше нуля, то обновленная надежность  $r_{i,j}$  берется по модулю, а жесткое решение  $x_{i,j}$  инвертируется.

4. Если «жесткое» решение  $x$  является кодовым словом, декодирование заканчивается, иначе декодер начинает следующую итерацию.

Сложность декодера UMP (быстрого декодирования по надежностям) значительно ниже, чем

сложность декодера, пересчитывающего вероятности, за счет того, что пересчет надежностей выполняется по упрощенной схеме (схеме «взвешенного» мажоритарного голосования, в качестве «весов» используется надежность проверок), а также за счет возможности использования исключительно целочисленных операций сложения и сложения по модулю два. Общую сложность одной итерации можно оценить как сумму сложностей  $M(2J + \log 2J - 2)$  сравнений,  $2NJ$  сложений по модулю два и  $2N(J + 4)(J - 1)$  сложений, где  $N$  – длина кодового слова,  $M$  – количество информационных бит в кодовом слове,  $J$  – количество единиц в столбце проверочной матрицы кода с низкой плотностью проверок на четность.

Также к достоинствам быстрого декодера по надежностям можно отнести то, что декодеру не требуется знать характеристики шума в канале (дисперсию и т. д.), а следовательно, такой декодер может работать в любом симметричном канале с двоичным входом. Далее в работе будет показано, как это свойство быстрого декодера по надежностям может быть использовано при кодовом квантовании изображений.

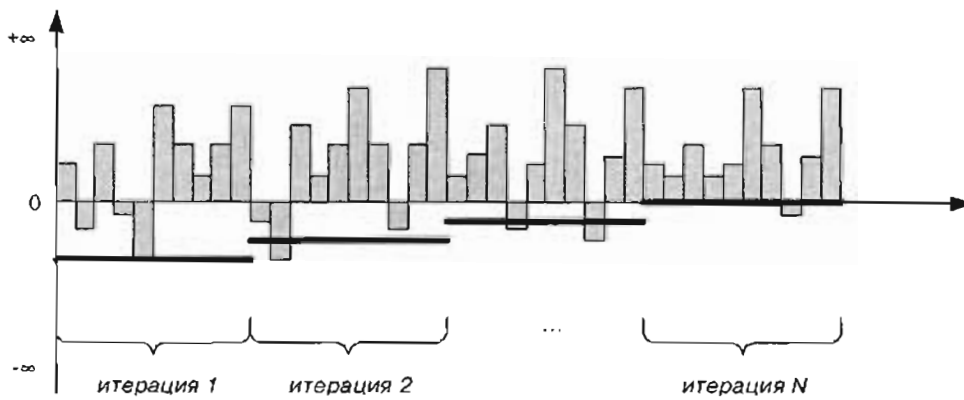
Недостатком быстрого декодера по надежностям является оценка вероятности ошибки декодирования, которая для канала с аддитивным гауссовским шумом оказывается на 0,5 дБ хуже, чем вероятность ошибки декодирования вероятностного декодера. Ниже будет описана модификация алгоритма, позволяющая добиться приближения вероятности ошибки декодирования к вероятности, получаемой при вероятностном декодировании.

### Многopороговое декодирование

Быстрый декодер по надежностям, рассмотренный выше, принимает решения относительно символов принятого вектора на основе пересчета надежностей и инвертирования символов «жесткого» решения, причем инвертирование символов производится всегда, когда соответствующая надежность после пересчета становится меньше нуля. Ноль в данном случае является порогом инвертирования символа, причем на всех итерациях этот порог остается неизменным, несмотря на то, что на первых итерациях декодирования надежности символов битов еще плохо определены, и инвертирование таких символов может привести как к исправлению ошибок, так и, наоборот, к внесению новых и последующему их размножению.

Основная идея многopорогового декодирования по надежностям состоит в том, чтобы изменять значения порогов инвертирования символов от одной итерации к другой следующим образом (рис. 2):

на первых итерациях порог инвертирования символов выбирается так, чтобы количество инвертированных символов было минимальным (вплоть до инвертирования только одного символа на первой итерации);



■ Рис. 2. Пороги для многопорогового декодирования: ■ — пересчитанные надежности; — — пороги

на последующих итерациях пороги инвертирования постепенно повышаются.

Общая схема работы многопорогового декодера по надежностям изображена на рис. 3:

1. Инициализация декодера и вычисление надежностей символов проверок производится так же, как и в алгоритме быстрого декодирования по надежностям: для всех элементов принятого вектора  $Y_i$  вычисляется жесткое решение  $X_i$  и надежность  $R_i$ , надежность является абсолютным значением  $Y_i$ , а также для всех  $m \in M(n)$  присваиваются  $Y_{m,n} = R_n$  и  $X_{m,n} = X_n$ .

2. Для каждой итерации  $i$  вычисляется пороговое значение  $THR_i$ .

3. Производится вычисление новых значений надежностей проверок и символов, так же, как и в алгоритме быстрого декодирования по надежностям: для всех  $n$  и  $m \in M(n)$  вычисляются суммы

$$S_{m,n} = X_n \oplus \sum_{n' \in N(m) \setminus n} X_{m,n'}$$

и значения  $Y_{m,n} \min = \min\{Y_{m,n'}\}$ .

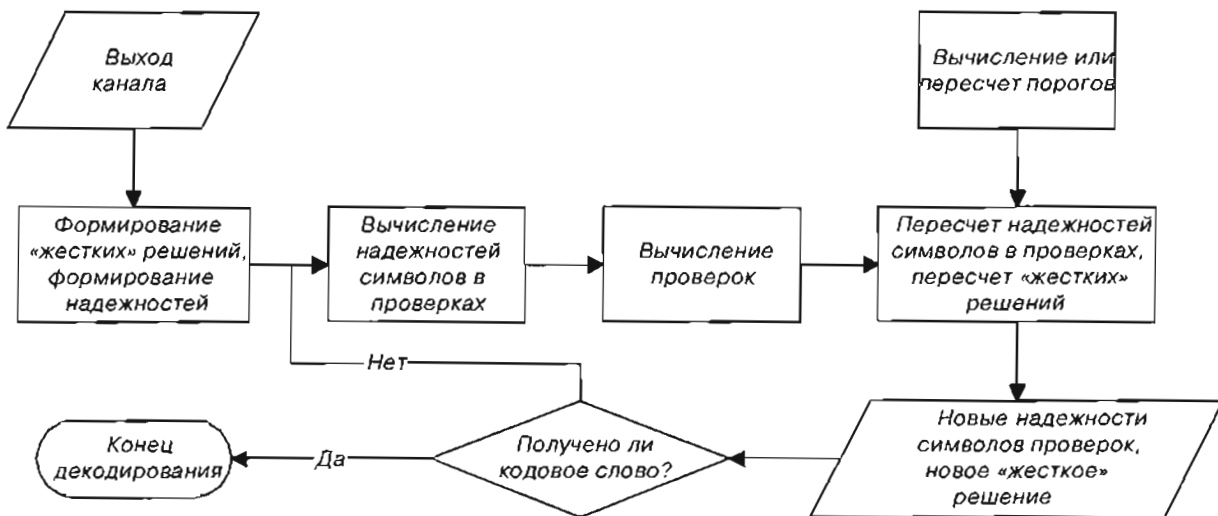
4. Для всех  $n$  и всех  $m \in N(n)$  вычисляются

$$Z_{m,n} = R_n + \sum_{m' \in M(n) \setminus m} (-1)^{S_{m',n}} Y_{m',n} \min.$$

5. Изменение текущих значений символов «жесткого» решения надежностей проверок и символов на новые значения происходит только для тех символов, для которых новое значение надежности оказывается меньше порога  $THR_i$ ; новое значение надежности принимается равным абсолютному значению пересчитанной надежности. Вводится «зона неопределенности»: для всех остальных символов, если значение надежности оказалось меньше нуля, надежность принимается равной нулю: для всех  $n$  вычисляются

$$Y_{m,n} = \begin{cases} Z_{m,n}, & Z_{m,n} > 0 \\ -Z_{m,n}, & Z_{m,n} < THR_i; \\ 0 & \end{cases}$$

$$X_{m,n} = \begin{cases} X_{m,n}, & Z_{m,n} > THR_i, \\ 1 \oplus X_{m,n}, & Z_{m,n} < THR_i, \end{cases}$$



■ Рис. 3. Схема работы многопорогового декодера по надежностям

$$Z_n = R_n \oplus \sum_{m \in M(n)} (-1)^{S_{m,l}} Y_{m,n} \text{ min};$$

$$C_n = \begin{cases} X_n, Z_n > 0 \\ 1 \oplus X_n, Z_n < 0. \end{cases}$$

6. Декодирование итеративно продолжается до тех пор, пока вектор  $C$  не станет кодовым словом или не будет выполнено заданное число итераций.

При многопороговом декодировании, если на первой итерации была исправлена хотя бы одна ошибка, декодирование на последующих итерациях становится значительно проще и общее качество декодирования улучшается [8]. По-прежнему для работы декодеру не требуется информация о шуме в канале, достаточно лишь задать надежности.

Декодер, работающий по многопороговой схеме, позволяет получить вероятность ошибки декодирования на 0,1–0,4 дБ лучшую, чем обеспечивает быстрый декодер по надежности «UMP», практически приближаясь к вероятности ошибки, получаемой при вероятностном декодировании кодов с низкой плотностью проверок на четность. На рис. 4 приведены кривые вероятностей ошибок декодирования (BER) двоичных кодов в канале с аддитивным гауссовским шумом, имеющим математическое ожидание, равное нулю, и энергию шума  $N_0$  при использовании трех различных декодеров: декодера по вероятностям, быстрого декодера по надежности и многопорогового декодера.

Помимо независимости от характеристик канала многопороговый декодер обладает свойством де-

кодеров кодов с низкой плотностью проверок на четность, а именно – универсальность и применимость для любой конструкции таких кодов.

### Критерии качества и сложности декодирования

Так как декодирование LDPC является итеративным, то в общем случае сложность декодирования можно представить как произведение сложности одной итерации на общее количество итераций:

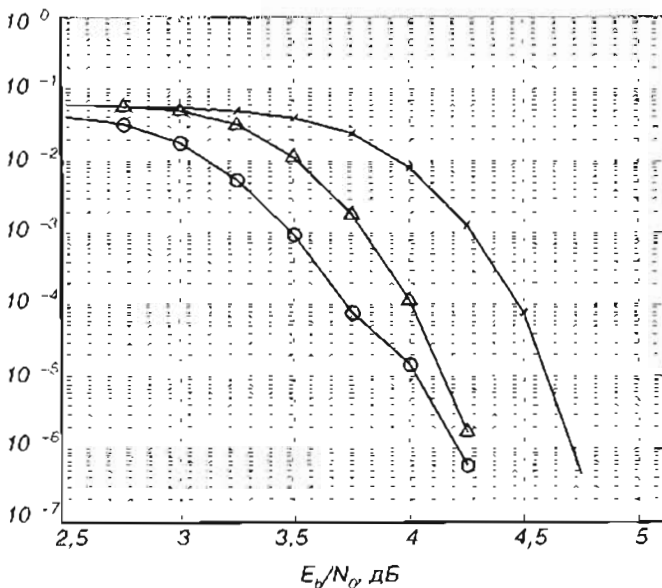
$$C_{\text{complete}} = C_{\text{one iteration}} \cdot N_{\text{iterations}}$$

Сложность одной итерации декодирования пропорциональна числу единиц в проверочной матрице [1–3, 7]. Таким образом, чтобы анализировать всю сложность декодирования, необходимо учитывать не только сложность одной итерации декодера, но и количество итераций, требуемых для обеспечения заданного качества декодирования.

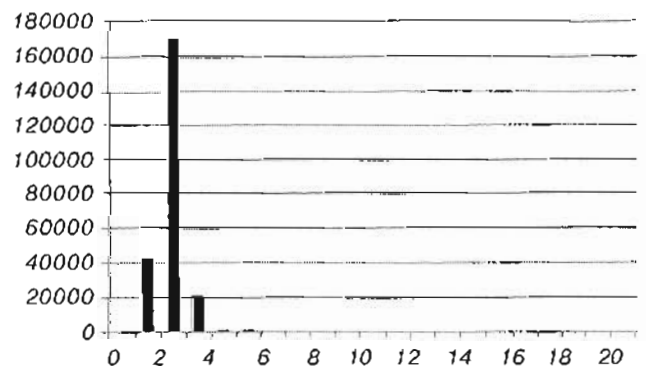
Будем понимать под скоростью сходимости зависимость вероятности ошибки декодирования от количества итераций, за которое данная вероятность ошибки декодирования была получена.

На рис. 5–7 приводятся гистограммы сходимости многопорогового декодера для различных конструкций LDPC кодов из расчета максимального количества итераций, равного 20. На осях гистограмм отложено количество итераций и количество попыток, в которых декодер закончил работу за данное число итераций. Во всех случаях конструкции кодов анализировались на отношении сигнал/шум, обеспечивавшем вероятность ошибки  $10^{-5}$ .

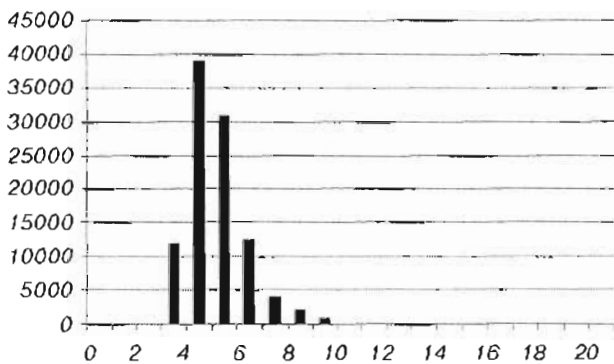
Из гистограмм видно, что требуемое количество итераций не является постоянной величиной для различных конструкций кодов, и наименьшее количество итераций требуется для декодирования Евклидово-геометрических кодов (EG-LDPC). Следовательно, общая сложность декодирования для этой конструкции будет минимальна, несмотря на большее количество единиц в проверочной матрице EG-LDPC кодов.



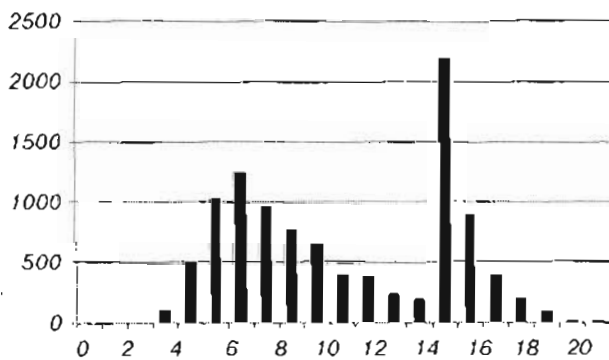
■ Рис. 4. Вероятность ошибки при декодировании (2048, 1649) – кода с низкой плотностью проверок на четность при помощи различных декодеров для различных отношений сигнал/шум в канале: ○ – вероятностный декодер; x – быстрый декодер; △ – многопороговый декодер



■ Рис. 5. Число итераций, затраченных на декодирование LDPC кодов, основанных на Евклидовых геометриях [4]



■ Рис. 6. Число итераций, затраченных на декодирование LDPC кодов, основанных на кодах Рида-Соломона с двумя информационными символами [5]



■ Рис. 7. Число итераций, затраченных на декодирование LDPC кодов, построенных по принципу density evolution [6]

## Заключение

В данной работе предложен алгоритм многопорогового декодирования для кодов с низкой плотностью проверок на четность. Являясь представителем класса «быстрых» декодеров, алгоритм по

сложности равен алгоритму «UMP», рассмотренному в работе, и требует сложности декодирования в 10 раз меньшей, нежели сложность традиционных алгоритмов. При этом многопороговый алгоритм декодирования позволяет в гауссовском канале получить вероятность ошибки декодирования всего на 0,1–0,2 дБ хуже, чем при использовании локально-оптимальных алгоритмов, а по сравнению с быстрыми алгоритмами декодирования новый алгоритм выигрывает 0,3–0,4 дБ по вероятности ошибки.

## Литература

1. Gallager R. G. Low-density parity-check codes. PhD thesis. – 1963. – 90 p.
2. Gallager R. G. Low-density parity-check codes // IEEE Trans. on Inform. Theory. – Vol. IT-8. – Jan, 1968. – P. 21–28.
3. MacKay D. J. C., Neal R. M. Near Shannon limit performance of low density parity check codes // IEE Electronics Letters. – Vol. 32. – N 18. – 29 Aug. 1996. – P. 1645–1655.
4. Djurdjevic I., Xu J., Abdel-Ghaffar K., Lin S. A class of low-density parity-check codes constructed based on Reed-Solomon codes with two information symbols // IEEE Communications Letters. – Vol. 7. – N 7. – July 2003. – P. 317–319.
5. Kou Y., Lin S., Fosserier M. P. C. Low-density parity-check codes based on finite geometries: A rediscovery and new results // IEEE Transactions on Information Theory. – Vol. IT-47. – Nov. – 2001. – P. 2711–2736.
6. Richardson T. J., Shokrollahi M. A., Urbanke R. L. Design of capacity approaching irregular low-density parity-check codes // IEEE Trans. Information Theory. – Vol. 47. – Feb. 2001. – P. 619–637.
7. Fosserier M. P. C., Mihaljevic M., Imai H. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation // IEEE Transactions on Communications. – 47(5). – May 1999. – P. 673–680.
8. Колесник В. Д., Мирончиков Е. Т. Декодирование циклических кодов. – М.: Связь, 1968. – 252 с.