

А. И. АКМАЛХОДЖАЕВ, А. В. КОЗЛОВ

НОВЫЙ АЛГОРИТМ СПИСОЧНОГО ДЕКОДИРОВАНИЯ ТУРБОКОДОВ

Рассматривается метод параллельного списочного турбодекодирования, в рамках которого предложен оконный списочный декодер сверточного кода с мягким выходом. Предложенный алгоритм позволяет добиться выигрыша на словах не только малой, но и большой длины.

Ключевые слова: турбокоды, турбодекодирование, списочное декодирование.

Введение. Декодирование турбокода — это итеративный процесс, в ходе которого два декодера сверточного кода с мягким выходом обмениваются значениями оценок внешних вероятностей [1—3]. Обычно достаточно 8—10 итераций для того, чтобы изменения оценок декодированных символов стали незначительными, дальнейшее итерирование декодера практически не приводит к уменьшению вероятности ошибки. Одним из способов снижения вероятности ошибки является использование списочного декодирования.

В настоящей работе рассматривается новый метод списочного декодирования турбокодов, основанный на списочном декодере сверточного кода с мягким выходом. Каждый мягкий выход является последовательностью априорных вероятностей, которые подаются на вход независимых турбодекодеров (рис. 1). Предложенный метод обеспечивает сходимость разных декодеров к различным кодовым словам, из которых затем выбирается подходящее. Список декодированных слов с мягкими решениями может быть сгенерирован с использованием первого, второго или обоих сверточных кодов.

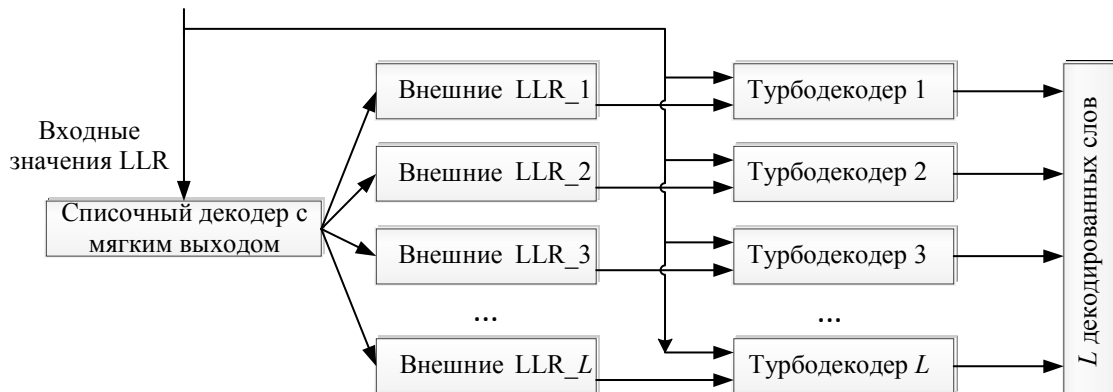


Рис. 1

Впервые подход с использованием списочного декодирования с мягким выходом был рассмотрен в работе [4], в рамках этого подхода был предложен алгоритм декодирования сверточного кода. Однако его использование давало выигрыш лишь на малых длинах, в то время как на больших длинах выигрыш был незначителен. Предложенный в настоящей рабо-

те оконный алгоритм списочного декодирования сверточного кода с мягким выходом позволяет достичь выигрыша и на больших длинах.

Оконный списочный декодер сверточного кода с мягким выходом. Оконный алгоритм строит на участках решетки (окнах) мягкие списки размером L , которые в дальнейшем используются при получении априорных вероятностей для всего информационного слова. Для получения такого списка в предложенном алгоритме используется оконный списочный декодер Витерби и оконный MAP-декодер. Так как в текущем окне не известны начальные и конечные состояния пути, для оконного алгоритма Витерби вводят понятие суффикса (рис. 2). Известно, что если длина суффикса N_{suff} равна 4—5 длинам кодового ограничения сверточного кода, то выжившие пути в конце суффикса, полученные с помощью алгоритма Витерби, с большой вероятностью имеют общий корень в конце окна (N_{win}). Таким образом, суффикс позволяет найти состояние в конце окна, в то время как начальные состояния равновероятны. Пусть t — номер начальной секции окна. Тогда оконный списочный алгоритм Витерби выглядит следующим образом.

1. Выполним параллельный списочный алгоритм Витерби [5, 6] на участке решетки от секции t до $t + N_{\text{win}} + N_{\text{suff}}$. В результате работы этого алгоритма получим для каждого состояния окна и суффикса L лучших путей.

2. Найдем путь с наибольшей конечной метрикой в секции $t + N_{\text{win}} + N_{\text{suff}}$. Пусть S_t — его начальное состояние в окне, а $S_{t+N_{\text{win}}}$ — конечное.

3. Из состояния $S_{t+N_{\text{win}}}$ выполним обратный проход по решетке в окне для оставшегося $L-1$ пути.

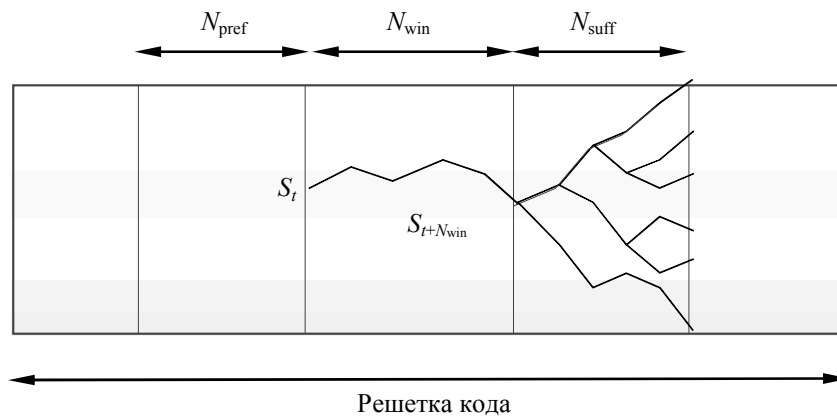


Рис. 2

В результате работы оконного списочного алгоритма получаются L путей в окне, которые могут начинаться в произвольном состоянии решетки, но заканчиваются в состоянии $S_{t+N_{\text{win}}}$.

Поскольку в MAP-алгоритме выполняется два прохода по решетке для нахождения прямых и обратных метрик, в оконном варианте помимо суффикса вводят префикс, который служит для более корректного вычисления метрик в окне (рис. 2), т.е. в оконном MAP [7] алгоритме расчет метрик начинается в секции $t - N_{\text{pref}}$ и заканчивается в секции $t + N_{\text{win}} + N_{\text{suff}}$, а начальные и конечные состояния равновероятны.

Используя полученные пути в окне и оконный MAP-алгоритм, можно получить список мягких решений, выполнив следующие шаги.

1. Найдем с помощью списочного оконного алгоритма Витерби L путей в решетке.
2. Определим первый элемент списка как результат работы алгоритма MAP в окне.
3. Обозначим как Γ_l все ребра, которые принадлежат l -му пути. Для нахождения l -го элемента списка:

— исключим в окне все ребра, которые принадлежат $l-1$ лучшему пути, но не принадлежат оставшимся $L-1$ путям, т.е. исключим все ребра из множества $\left(\bigcup_{i=1}^{l-1} \Gamma_i - \bigcup_{i=1}^{l-1} (\Gamma_l \cap \Gamma_i) \right)$;

— выполним MAP-алгоритм в окне с исключенными ребрами. Выходные надежности алгоритма и будут искомым элементом списка.

Удаление из решетки ребра лучших путей, которые будут учтены в соответствующих элементах списка, позволит рассмотреть менее вероятные решения. Это обеспечит схождение последующих процессов турбодекодирования к другим кодовым словам, среди которых, возможно, будет правильное.

Стоит отметить, что помимо MAP-алгоритма можно использовать его подоптимальные варианты, такие как Max-Log-MAP или Scaled-Max-Log-MAP [8].

Списочный декодер турбокода. Для того чтобы найти мягкое решение для всего информационного слова, решетка сверточного кода разбивается на равные отрезки. В каждом из отрезков с помощью описанного алгоритма находится список мягких решений, из которых в дальнейшем формируется мягкое решение для всего слова. Однако при разбиении слова на N окон с длиной списка L в каждом окне, при полном переборе элементов списка в каждом окне, общий размер списка равен L^N . Экспоненциальное увеличение размера списка не позволяет использовать полный перебор при нахождении общего списка, поэтому был предложен следующий подход к его уменьшению.

Рассмотрим окно j с длиной списка $L=2$. Вычислим евклидово расстояние между первым и вторым элементами списка:

$$\varepsilon_j = \sum_{i=1}^{N_{win}} \left(E_{1,j}^i - E_{2,j}^i \right)^2,$$

где $E_{1,j}$ и $E_{2,j}$ — первый и второй элементы списка соответственно.

Чем больше евклидово расстояние между векторами, тем с большей вероятностью соответствующие процессы турбодекодирования сойдутся к различным словам. Таким образом, в каждом окне вычисляется евклидово расстояние между элементами списка и оставляются лишь элементы с наибольшим расстоянием. В случае $L=2$ уменьшение размера списка в окне приведет к тому, что останется лишь его первый элемент и размер общего списка сократится в 2 раза.

Результаты моделирования. Предложенный метод анализировался путем моделирования (рис. 3) на примере турбокода LTE [9].

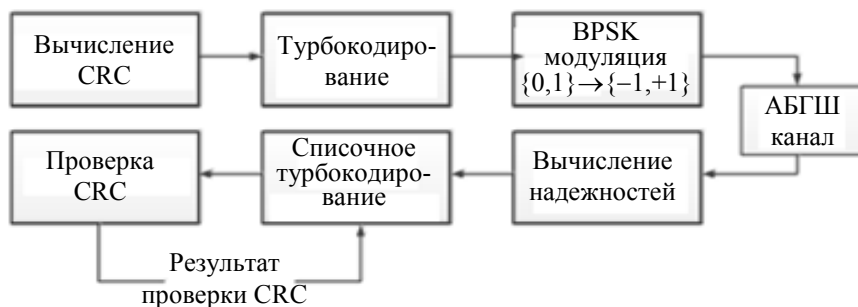


Рис. 3

Как видно из рисунка, информационное слово содержит код CRC, который используется для выбора правильного слова из списка. В качестве CRC используется полином CRC_V из стандарта 3GPP LTE [9].

Для вычисления списка мягких решений использовался первый компонентный код турбокода. Моделирование производилось для информационного слова длиной 512, в качестве

алгоритма декодирования турбокода и генерации списка был выбран алгоритм Scaled-Max-Log-MAP с весовым коэффициентом 0,75. Решетка сверточного кода была разделена на 8 окон по 64 секции в каждом. Размер списка был выбран равным 2, т.е. общий размер списка — 256. При моделировании учитывалась вероятность ошибки на информационное слово (FER).

Из результатов моделирования видно (рис. 4), что предложенный метод по сравнению с алгоритмом Scaled-Max-Log-MAP (кривая 1) дает выигрыш порядка 0,19 дБ для полного списка. Кривые 2—5 характеризуют результаты моделирования для списочного декодера с уменьшенным размером списка: 16 (2), 32 (3), 64 (4) и 256 (5) слов. Заметное уменьшение длины списка лишь незначительно снижает производительность декодера — так, для списка длины 16 выигрыш равен 0,13 дБ.

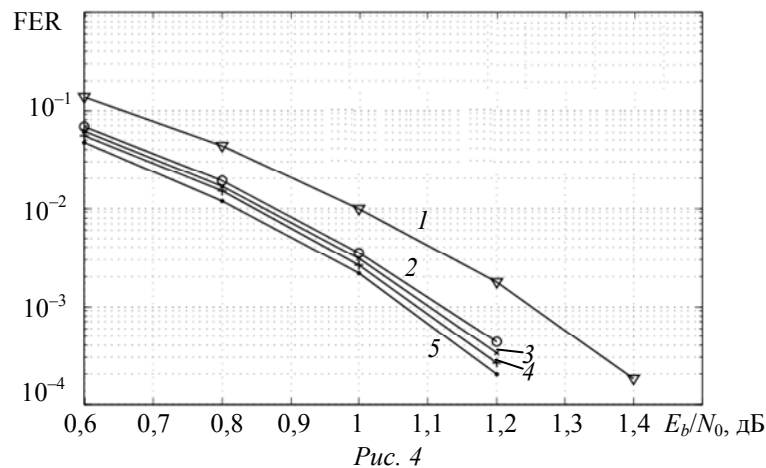


Рис. 4

Заключение. В настоящей работе рассмотрен новый метод списочного декодирования турбокодов, в рамках которого был предложен оконный алгоритм списочного декодирования сверточного кода с мягким выходом. Предложенный метод позволяет добиться выигрыша по сравнению с алгоритмом Scaled-Max-Log-MAP при длине информационного слова 512 вплоть до 0,19 дБ на полном списке. Также рассмотрен подход к уменьшению списка мягких решений на основе расстояния Евклида между элементами списка. Было показано, что при значительном уменьшении списка производительность декодера снизилась незначительно. На списках размером 16 и 32 выигрыш составляет 0,13 и 0,15 дБ соответственно.

СПИСОК ЛИТЕРАТУРЫ

1. Козлов А. В. Декодирование LDPC-кодов в дискретном канале flash-памяти // ИУС. 2007. № 5(30). С. 31—35.
2. Белоголовый А. В., Крук Е. А. Многопороговое декодирование кодов с низкой плотностью проверок на четность // ИУС. 2005. № 1(14). С. 25—31.
3. Berrou C., Glavieux A., Thitimajshima P. Near Shannon limit error-correcting coding: turbo codes // Proc. IEEE Intern. Conf. on Communications. Geneva, Switzerland, 1993. P. 1064—1070.
4. Акмалходжаев А. И., Козлов А. В. Списочное декодирование турбо кодов // СПИСОК-2012. Матер. Межвуз. науч. конф. по проблемам информатики. 2012. С. 194—199.
5. Nill C., Sundberg C.-E. W. List and Soft Symbol Output Viterbi Algorithms: Extensions and Comparisons // IEEE Transactions on Communications. 1995. Vol. 43, N 2. P. 277—287.
6. Narayanan K. R., Stuber G. L. List Decoding of Turbo Codes // IEEE Transactions on Communications. 1998. Vol. 46, N 6. P. 754—762.
7. Bahl L., Cocke J., Jelinek F., Raviv J. Optimal decoding of linear codes for minimizing symbol error rate // IEEE Transactions on Information Theory. 1974. Vol. 20. P. 284—287.
8. Claussen H., Karimi H. R., Mulgrew B. Improved max-log-map turbo decoding by maximization of mutual information transfer // EURASIP J. on Applied Signal Processing. 2005. P. 820—827.

9. 3GPP LTE TS 36.212 V8.3.0: “Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding”.

Сведения об авторах

- Акмал Илхомович Акмалходжаев* — Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра безопасности информационных систем; программист; E-mail: Akmal.ilh@gmail.com
- Александр Владимирович Козлов* — Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра безопасности информационных систем; ведущий программист; E-mail: akozlov@vu.spb.ru

Рекомендована кафедрой
№ 51 безопасности информационных систем

Поступила в редакцию
01.02.13 г.