

УДК 621.391, 519.725

СИГНАЛЬНО-КODOVAYA КОНСТРУКЦИЯ ДЛЯ СВЕРХНАДЕЖНОЙ ПЕРЕДАЧИ ДАННЫХ НА ОСНОВЕ КОДА С ИРРЕГУЛЯРНЫМИ ПОВТОРЕНИЯМИ-НАКОПЛЕНИЯМИ

© Автор, 2020

А.Э. Жданов, к.т.н., в.н.с., АО «Концерн «Созвездие», e-mail: alexandr_zhdanov@mail.ru

Представлена сигнально-кодovая конструкция для предоставления сервиса, аналогичного режиму URLLC в стандарте 5G NR, основными требованиями которого является вероятность блочной ошибки не более 10^{-5} при задержке не более 1 мс. Дан метод формирования кодовой матрицы для кода иррегулярных повторений с накоплениями. Для длины информационного блока $K = 172$ и скорости кодирования $R = 1/3$ достигается $BLER = 10^{-5}$ при $E_b/N_0 = 3.8$ дБ, что соответствует отношению сигнал-шум в канале передачи $E_{sym}/N_0 = -0.97$ дБ.

Ключевые слова: режим сверхнадежной передачи данных с низкой задержкой, пятое поколение мобильной связи, код с низкой плотностью проверок на четность, код иррегулярных повторений с накоплениями.

CHANNEL CODING FOR ULTRA-RELIABLE DATA TRANSMISSION BASED ON IRREGULAR REPEAT-ACCUMULATE CODES

© Author, 2020

A.E. Zhdanov, Ph.D. in Engineering Sciences, Leading Research Engineer, JSC «Concern «Sozvezdie», e-mail: alexandr_zhdanov@mail.ru

The channel coding scheme for providing a service similar to the ultra reliable low latency communication (URLLC) in the 5G NR standard is presented. The base requirements of the URLLC mode: $BLER = 10^{-5}$ and delay no more than 1 ms are satisfied. For information block length $K = 172$ and coding rate $R = 1/3$ the $BLER = 10^{-5}$ is achieved with $E_b/N_0 = 3.8$ dB which corresponds to the SNR in the channel $E_{sym}/N_0 = -0.97$ dB.

Keywords: URLLC, 5G NR, LDPC, IRA.

Согласно рекомендации ITU-R в пятое поколение мобильной связи введен режим сверхнадежной передачи данных с низкой задержкой (ultra reliable low latency communication – URLLC) [1], основными требованиями которого является вероятность блочной ошибки не более 10^{-5} при задержке не более 1 мс. В связи с малой задержкой затруднено применение алгоритмов повышения достоверности за счет автоматического запроса-повторения, и размеры информационного пакета составляют порядка 20÷200 бит. Для реализации режима URLLC в действующей версии стандарта применены как полярные коды с циклической контрольной суммой [2], так и коды с низкой плотностью проверок на четность и множественными типами ребер (MET-LDPC) [3]; их характеристики могут быть найдены в [4] на стр. 268. В ходе дискуссий по стандартизации предлагалось доработать существующий в LTE (Long Term Evolution – стандарт мобильной связи 4 поколения) параллельный сверточный турбокод (PCTC) [5, 6], который показывал характеристики лучше, чем остальные кандидаты на длинах порядка 200 бит, и

хуже остальных – на длинах порядка 20 бит [7]. Основным вызовом в разработке помехоустойчивых кодов для режима сверхнадежной передачи данных является контроль за явлением, которое известно как «полка ошибок – error floor». Явление заключается в том, что начиная с некоторого значения рост отношения сигнал-шум (SNR) слабо влияет на величину ошибки как блочной (BLER), так и битовой (BER); и это обусловлено влиянием небольшого числа кодовых слов низкого веса. При более низком значении SNR влияние этого эффекта незаметно. В современных кодах, допускающих итеративное декодирование, происходит лавинообразное сокращение ошибок; этот интервал значений SNR носит название «регион водопада» – «waterfall region». Также для турбодекодера PCTC характерна потеря правильно обнаруженного кодового слова, то есть обнаруженное правильное кодовое слово на какой-то полуитерации может быть заменено неправильным при последующих полуитерациях; это является частным случаем эффекта «error floor». Для полярного кода с циклической контрольной суммой «error floor» от-

сутствует до $BER = 10^{-10}$ [8] при списочном декодировании с последовательным подавлением и проверкой циклической контрольной суммы (*SCL-CRC*) [2]. Ввиду последовательного и списочного характера алгоритма следует ожидать сложности при его реализации в аппаратуре. Что касается *MET-LDPC*, то они являются развитием классических кодов с низкой плотностью проверок на четность (*LDPC*), в том числе кодов, основанных на повторениях с иррегулярными накоплениями. Подобные коды и их декодирование графически представляют при помощи графа Таннера (двудольного графа, состоящего из переменных и проверочных узлов [9]). При этом степень узла называют количеством связанных с ним ребер. Код базируется на протографе, каждая из вершин которого формирует свой тип вершин. Каждый из типов вершин может быть соединен со строго определенным другим типом вершин. Соединения вершин соответствующих типов протографа перемежаются посредством циклического сдвига. Этот процесс носит название «lifting». Таким образом формируется кодовая матрица с заданными свойствами. Подробнее *MET-LDPC* описаны в [10] на стр. 382. Подобный подход позволяет осуществить перемежение данных параллельно, например, используя векторный подход [11, 12]. Цель данной статьи – показать, что характеристики декодирования, удовлетворяющие режиму *URLLC*, могут быть достигнуты в классической схеме кода повторений с иррегулярными накоплениями [13], где подобные коды характеризуются как турбоподобные коды. Структура кода показана на рис. 1, откуда видно, что код может быть задан при помощи шаблона иррегулярных повторений, перемежителя и периода накоплений. В свою очередь шаблон иррегулярных повторений может быть задан в виде полинома, где степени соответствуют количеству повторений, а коэффициенты – доле символов, повторенных заданное число раз [13]. Сочетание свойств как кодов с низкой плотностью проверок на четность, так и турбокодов с решетчатой диаграммой является уникальным и не до конца использованным при проектировании систем передачи данных. Кодирование кодов с «повторением – накоплением» (*IRA*) осуществляют в следующем порядке: повторение каждого информационного символа, перемежение повторенных символов, накопление суммы по модулю 2 перемеженных символов (рис. 1). Коды *IRA* обладают свойствами как турбо-кодов, так и кодов с низкой плотностью

проверок на четность и могут быть декодированы как в параллельном режиме – алгоритм обмена сообщениями, так и в турборежиме. Следует отметить, что в последнем случае алгоритм по существу является гибридным, состоящим из MAP алгоритма декодирования сверточного кода с двумя состояниями и алгоритма обмена сообщениями с неравномерно повторенными информационными символами [14]. Из [13] известно, что наибольшая эффективность от применения достигается неравномерным или иррегулярным повторением информационных символов. Кроме того, характеристики кодов повторения с накоплением в большой степени характеризуются перемежителем, который совместно с шаблоном иррегулярного повторения и шаблоном накопления контрольных сумм определяет граф Таннера для данного кода [9].

Рассмотрим типичные возникающие конфигурации ошибок (рис. 2). Наличие кодовых слов малого веса следует из существования циклов малой длины. Рассматривая цикл как замкнутый путь, проходящий по ребрам графа Таннера, можно заметить, что изменение всех бит в переменных узлах не вызовет изменения контрольных сумм в проверочных узлах, входящих в цикл. Так, например, в цикле длиной два (рис. 2 а) контрольная сумма в проверочном узле вообще не зависит от значения переменного узла, поскольку они соединены двумя ребрами. В случае цикла-4 (рис. 2 б) совместное изменение значений в двух переменных узлах не может быть детектировано контрольными узлами, входящими в цикл. В случае линейной комбинации циклов формируется «останавливающее множество» (рис. 2 с), которое можно определить как подмножество проверочных вершин, для которого не существует переменных вершин, соединенных единственным ребром с каким-либо из проверочных вершин, входящих в это подмножество [15]. Множество переменных вершин, соединенных только с «останавливающим» подмножеством, содержит наиболее вероятные комбинации ошибок. Действительно, ошибки в переменных узлах, вызванные в том числе циклом, могут быть исправлены за счет других ребер, связанных с данным узлом; их количество может быть оптимизационным критерием. При прочих равных условиях цикл, наружу которого смотрят больше ребер, является более безопасным с точки зрения возникновения ошибок. Данный критерий известен как *ACE* [16], стр. 174. Все ошибочные комбинации, представленные на

рис. 2, кроме *e*) имеют $ACE = 0$, поскольку не имеют ребер, направленных наружу останавливающего множества. Функцией «соседства» будем называть количество проверочных (проверочных вершин со степенью 2) переменных узлов, вовлеченных в останавливающее множество,

во, как показано на рис. 2 d. Такое останавливающее множество всегда можно сформировать для информационных узлов с одинаковой минимальной степенью 3, однако если степень информационных узлов разная, то $ACE > 0$ (рис. 2 e).

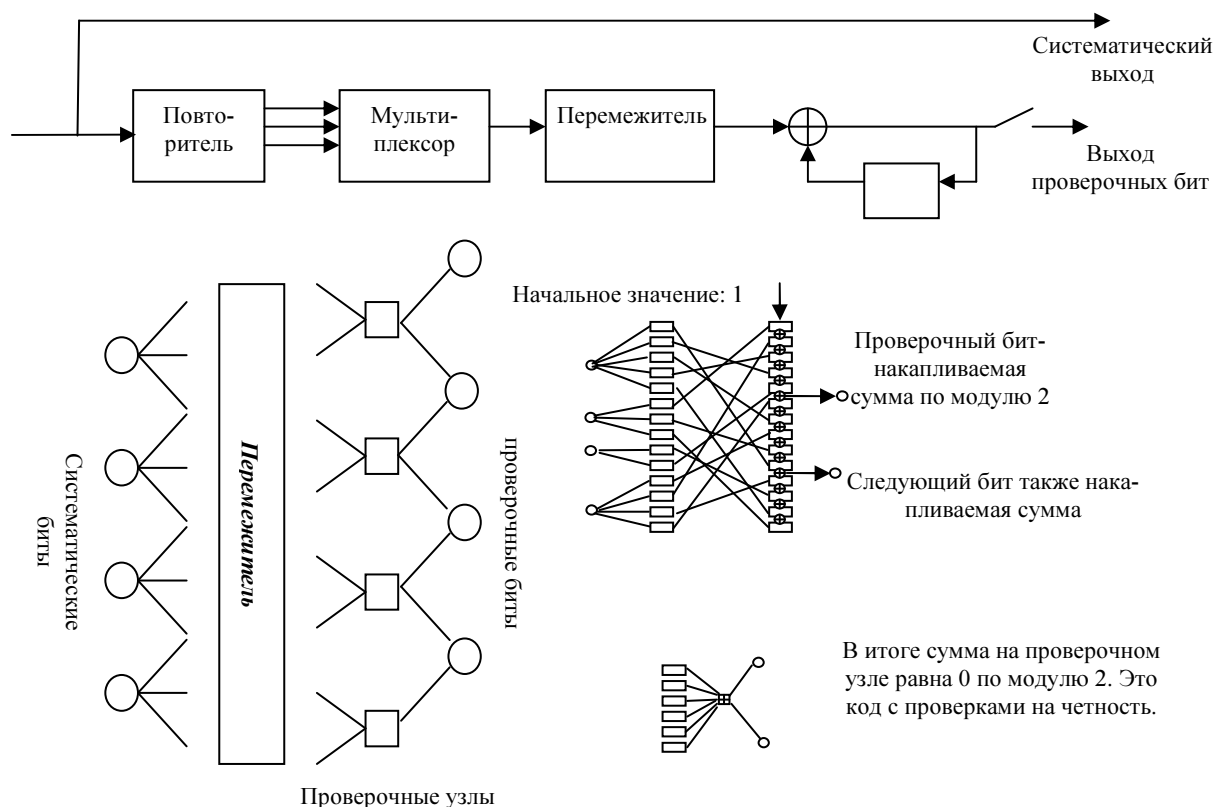


Рис. 1. Граф Таннера и структура кодера для кода повторений с накоплениями

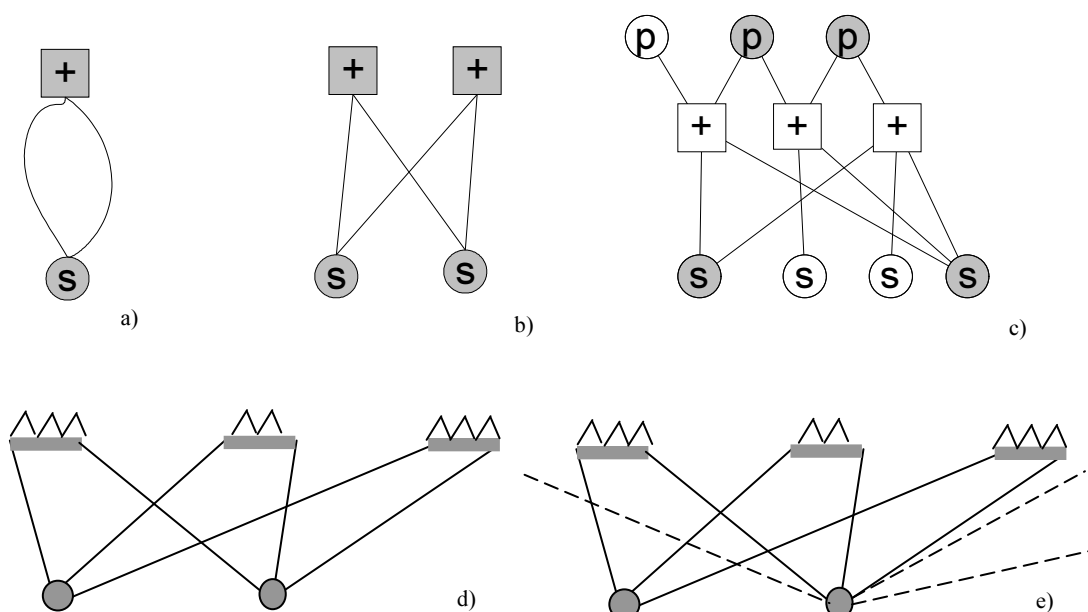


Рис. 2. Конфигурации ошибок

Требования к перемежителю заключаются в том, чтобы обеспечить достаточное разнесение повторенных символов, обеспечив тем самым отсутствие или малое число кодовых слов с низким весом. Существуют два класса перемежителей: псевдослучайный с внесенным детерминизмом и детерминистический с внесенной случайностью. Для формирования псевдослучайного с внесенным детерминизмом перемежителя применяют датчик случайных чисел, при помощи которого определяют номер перемежаемого символа в последовательности после перемежения. Последовательность получаемых номеров подвергают проверке на достаточность разнесения подряд идущих повторенных информационных символов с выборкой (откатом назад) последовательностей, не прошедших подобной проверки, – известен, в частности, алгоритм S-random [17]. Запишем условие для S-random перемежителя в виде, представленном в [10], стр. 368: $|i - j| + |\Pi(i) - \Pi(j)| \geq S$.

Также известен патент [18], где реализован модифицированный S-random алгоритм. Псевдослучайные перемежители с внесенным детерминизмом требуют много памяти для хранения; кроме того, перемежение не может быть выполнено параллельно. От этого недостатка свободны детерминистические перемежители с внесенной случайностью, где способ взаимно однозначного соответствия определяется путем задания параметризованного алгоритма, а случайность определяет псевдослучайная последовательность, длина которой меньше длины всего перемежителя. Такой перемежитель представим в виде прямоугольной матрицы, где длина строки соответствует длине псевдослучайной последовательности, а размеры столбца определяют банк памяти для хранения информации. Следует отметить, что алгоритм предполагает параллельный доступ ко всем банкам памяти. Таким образом, длина псевдослучайной последовательности также является фактором параллельности алгоритма. Перемежитель должен обеспечивать возможность параллельной записи и чтения информации, однако для банка памяти в ПЛИС за такт возможно чтение одного элемента хранения и запись одного элемента хранения. Конфликт обращения к памяти состоит в попытке записать (прочитать) больше одного элемента хранения информации из отдельно взятого банка памяти. Предотвращение таких конфликтов должно быть заложено в алгоритм.

Детерминистические перемежители со внесенной случайностью перемежителя могут фор-

мироваться «на лету», то есть непосредственно в процессе декодирования, и задаются путем определения параметров алгоритма. Чаще всего такие перемежители применяют для перемежения турбокодов. Широкоизвестными являются три типа перемежителя:

- почти регулярный перемежитель (ARP) [19];
- квадратично-полиномиальный перемежитель (QPP) [20];
- перемежитель с «размытием простых чисел» (DRP) [21].

В [22] показано, что все их можно свести к перемежителю ARP вида

$$i_1 = \Pi(i) = (Pi + \Theta(i \bmod n_1)) \bmod N, \quad (1)$$

где N – длина перемежителя; P – число взаимно простое с N ; i_1 – индекс после перемежения; i – индекс до перемежения; Θ – псевдослучайный перемежитель длиной n_1 , такой что $N = n_1 \times n_2$. Отличие одного алгоритма от другого заключается в выборе псевдослучайного перемежителя Θ .

Предлагаемый код получен по способу [23] и оптимизирован по методу ACE для снижения полки ошибок. В ходе оптимизации рассматривались только перестановки внутри каждого банка памяти. То есть вместо циркулярного или циклического сдвига внутри банка памяти применялась произвольная перестановка, что является незначительным усложнением при реализации на реконфигурируемой платформе SDR, учитывая малую длину кодового слова. Способ [23] отличается тем, в формуле (1) применяют псевдослучайную последовательность $\Theta(i \bmod n_1)$, обладающую следующими свойствами:

$$\forall i, j_1, j_2 : |i - j_1| < S,$$

$$|i - j_2| < S \Rightarrow |\Theta(i) - \Theta(j_1)| \neq |\Theta(i) - \Theta(j_2)|, \quad (2)$$

где $i, j_{1,2} < n_1$ – номера кодовых символов до перемежения, а $\Theta(i)$, $\Theta(j_1)$, $\Theta(j_2)$ – номера тех же кодовых символов после перемежения, S – целое положительное число. Иными словами, внутри интервала S любая разность между перемеженными номерами должна отличаться от любой другой разности, лежащей в этом интервале, что предотвращает при дальнейшем линейном конгруэнтном преобразовании появление цикла, в который вовлечены два столбца матрицы перемежителя целиком. Последова-

тельность может быть сформирована способом аналогичным S-random перемежителю. Отличие заключается в проверке условия: случайный номер-кандидат попадает в линию задержки (скользящее окно) длиной S , где содержатся предыдущие $S - 1$ отобранных номеров; между ними вычисляют попарно разности, которые должны отличаться. При выполнении условия случайным образом формируют новый номер-кандидат из оставшихся допустимых. При длине $n_1 = 24$ максимально возможное значение параметра $S = 3$, что достаточно для обеспечения дистанционного разнесения переменных узлов со степенью повторения 3. В иррегулярном шаблоне количество узлов с минимальной степенью повторения 3 большинство.

На следующей стадии оптимизации выполняют циклические вращения столбцов матрицы перемежителя, добиваясь максимизации минимального значения метрики «соседства» в строке перемежителя. Достаточно выполнить эту оптимизацию только для одной строки перемежителя, поскольку это значение повторяется и для остальных строк. Следует отметить, что здесь мы выполняем оптимизации, ориентируясь на минимальную степень переменного узла (обычно 3), который равен параметру S в формуле (2). Следует отметить, что первые две стадии оптимизации выполняют, ориентируясь лишь на минимальную степень повторения, но не на весь иррегулярный шаблон повторения, что позволяет с малой вычислительной сложностью сформировать перемежитель и получить удовлетворительные характеристики в режиме лавинообразного сокращения ошибок (регион «waterfall»), за исключением региона «error floor» [24].

Шаблон иррегулярного повторения учитывают на третьей стадии, где для снижения «полки ошибок» выполняют перестановки в банках памяти (столбцах перемежителя). Каждая перестановка затрагивает только один банк памяти (столбец перемежителя) и не может привести к возникновению конфликтов доступа к памяти. Для каждого номера в столбце из соответствующего переменного узла выполняют поиск циклов в ширину. Также выполняют поиск циклов с низким ACE (числом смотрящих наружу цикла ребер). Заметим, что циклы с низким ACE необязательно имеют низкий охват. Каждая перестановка рассматривается индивидуально по следующим критериям: общий охват циклов

должен быть не менее 6 (циклы с охватом 2 и 4 должны быть исключены); каждая перестановка должна увеличивать фактор разнесения, представляющий собой сумму модулей разности номеров до и после перемежения, величину эквивалентную левой части формулы (2), но посчитанную между переставленным номером и любым другим номером в данной строке перемежителя. Той перестановке, которая максимизирует минимальное значение фактора перемежения, отдается предпочтение.

Результатом оптимизации является перемежитель длиной 1032 символа, который представим как матрица 24×43 . Шаблон иррегулярного повторения $\{Q, P\} = \{[3, 6, 12, 24], [72, 68, 30, 2]\}$ взят из [24] для количества информационных символов $K = 172$ со средним числом повторений 6. Результаты моделирования представлены на рис. 3. Для длины информационного блока $K = 172$ и скорости кодирования $R = 1/3$ достигается $BLER = 10^{-5}$ при $E_b/N_0 = 3.8$ дБ, что соответствует отношению сигнал-шум в канале передачи $E_{sym}/N_0 = -0.97$ дБ, однако далее следует «заниженный» регион «error floor». Устройства кодирования и декодирования отличает простота реализации. Разработанный код может быть эффективно применен в системах SDR для предоставления сервиса аналогового URLLC. Основной вывод статьи заключается в том, что характеристики классической схемы кода IRA могут быть улучшены не за счет увеличения охвата минимального цикла, а в первую очередь – за счет увеличения дистанционного разнесения в перемежителе.

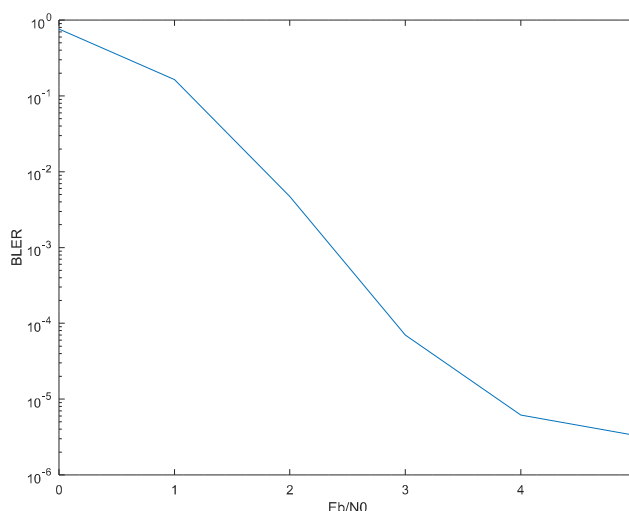


Рис. 3. $BLER$ для блока $K = 172$ и $R = 1/3$, AWGN, QPSK

Литература

1. ITU-R M. Minimum requirements related to technical performance for IMT-2020 radio interface (s). 2017.
2. Li B., Shen H., Tse D. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check // IEEE Communications Letters. 2012. Vol. 16, № 12. Pp. 2044–2047.
3. Richardson T. et al. Multi-edge type LDPC codes // Workshop honoring Prof. Bob McEliece on his 60th birthday, California Institute of Technology, Pasadena, California. 2002. Pp. 24–25.
4. Janyani V. et al. Optical and Wireless Technologies. Springer Singapore, 2020.
5. Maunder R. R1-1702856 Enhanced turbo codes for URLLC. 2017.
6. Garzón-Bohórquez R., Nour C. A., Douillard C. Improving Turbo codes for 5G with parity puncture-constrained interleavers // 2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC). IEEE, 2016. Pp. 151–155.
7. Sybis M. et al. Channel coding for ultra-reliable low-latency communication in 5G systems // 2016 IEEE 84th vehicular technology conference (VTC-Fall). IEEE, 2016. Pp. 1–5.
8. Eslami A., Pishro-Nik H. On finite-length performance of polar codes: stopping sets, error floor, and concatenated design // IEEE Transactions on communications. 2013. Vol. 61, № 3. Pp. 919–929.
9. MacKay D.J.C. Information theory, inference and learning algorithms. Cambridge university press, 2003.
10. Richardson T., Urbanke R. Modern coding theory. Cambridge university press, 2008.
11. Richardson T., Novichkov V. Methods and apparatus for decoding LDPC codes: пат. 6633856 США. 2003.
12. Jin H., Richardson T., Novichkov V. Methods and apparatus for encoding LDPC codes: пат. 6961888 США. 2005.
13. Jin H. et al. Irregular repeat-accumulate codes // Proc. 2nd Int. Symp. Turbo codes and related topics. 2000. Pp. 1–8.
14. Li J. Low-complexity, capacity-approaching coding schemes: Design, analysis and applications. 2003.
15. Richardson T.J., Urbanke R.L. The capacity of low-density parity-check codes under message-passing decoding // IEEE Transactions on information theory. 2001. Vol. 47, № 2. Pp. 599–618.
16. Declercq D., Fossorier M., Biglieri E. Channel Coding: Theory, Algorithms, and Applications: Academic Press Library in Mobile and Wireless Communications. Academic Press, 2014.
17. Dolinar S., Divsalar D. Weight distributions for turbo codes using random and nonrandom permutations. 1995.
18. Sadjadpour H.R., Salehi M., Sloane N.J.A. Two stage S–Random interleaver: пат. 6453442 США. 2002.
19. Berrou C. et al. Designing good permutations for turbo codes: towards a single model // 2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577). IEEE, 2004. Vol. 1. Pp. 341–345.
20. Sun J., Takeshita O.Y. Interleavers for turbo codes using permutation polynomials over integer rings // IEEE Transactions on Information Theory. 2005. Vol. 51, № 1. Pp. 101–119.
21. Crozier S., Guinand P. High-performance low-memory interleaver banks for turbo-codes // IEEE 54th Vehicular Technology Conference. VTC Fall 2001. Proceedings (Cat. No. 01CH37211). IEEE, 2001. Vol. 4. Pp. 2394–2398.
22. Bohorquez R.G., Nour C.A., Douillard C. On the equivalence of interleavers for turbo codes // IEEE wireless communications letters. 2014. Vol. 4, № 1. Pp. 58–61.
23. Жданов А.Э. Способ передачи данных в системе цифровой радиосвязи на основе кодов с низкой плотностью проверок на четность и способ перемешивания кодовых символов. 2019. Сеп. 16. Патент РФ RU2700398. Режим доступа: https://yandex.ru/patents/doc/RU2700398C1_20190916.
24. Жданов А.Э. Способ передачи данных на основе кодов с низкой плотностью проверок на четность. 2019. Дек. 5. Патент РФ RU2708349. Режим доступа: https://yandex.ru/patents/doc/RU2708349C1_20191205.

Статья поступила в редакцию 8 июля 2020 года