О применении декодеров кодов, исправляющих ошибки, в каналах со стираниями

Е.Е. Айдаркин, Н.С. Могилевская

Южный федеральный университет, Ростов-на-Дону

Аннотация: Во всех каналах передачи данных возникают непреднамеренных ошибки. Стандартным способом борьбы с ними является использование помехоустойчивых кодеков, основанных на применении алгебраических кодов исправления ошибок. Существуют каналы передачи, в которых возникает особый вид ошибок – стирания, т.е. разновидность ошибки, при которой известно местоположение ошибки, но не известна ее величина. В теории кодирования утверждается, что методы борьбы с ошибками могут быть применены для защиты данных от стираний, однако, эти утверждения не сопровождаются подробностями. Данная работа восполняет этот пробел. Построены алгоритмы исправления стираний с помощью произвольных декодеров для кодов, корректирующих ошибки. Сформулированы леммы о корректности построенных алгоритмов, получены некоторые оценки вероятности успешного декодирования.
Ключевые слова: каналы со стираниями, помехоустойчивый код, алгебраический код, декодер кода исправления ошибок, алгоритм исправления стираний.

Введение

Во всех системах цифровой связи для защиты передаваемых данных от непреднамеренных ошибок используются специальные кодеки, основанные на применении алгебраических помехоустойчивых кодов (см., например, [1-3, 9-11], стандарты передачи данных CCSDS, DVB-H/T/S, IEEE802.16). Упрощенная схема передачи данных по таким системам представлена на рисунке 1.

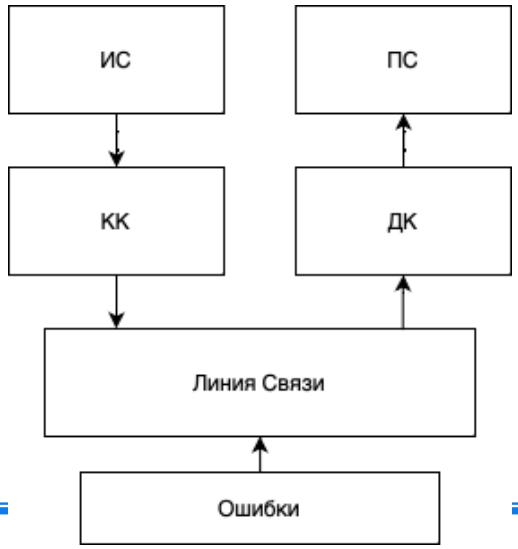

Рис. 1. – Схема передачи данных по каналу с защитой от стираний

Данные из источника сообщений (ИС) поступают в кодер канала (КК), где они кодируются некоторым кодом C, затем закодированные данные проходят через линию связи, где могут быть повреждены ошибками. Искаженные закодированные данные поступают на вход декодера канала (ДК), который в процессе декодирования данных пытается исправить ошибки. Успешность работы ДК зависит от числа и структуры ошибок, повредивших передаваемые данные. Результат работы ДК поступает к получателю сообщений (ПС).

Обычно при обсуждении помехоустойчивой защиты систем связи говорят об аддитивных независимых ошибках, т.е. предполагается, что в потоке передаваемых данных значения некоторых символов изменяются. В этом случае декодеру канала не известны позиции ошибочных символов, а в случае недвоичных каналов и величины ошибок. В ряде работ рассматриваются каналы связи, в которых происходит особый тип ошибок – стирания [4-5]. Стирание – это разновидность ошибки, при которой известно местоположение ошибки, но неизвестна ее величина. Другими словами, в потоке данных, заданном над некоторым алфавитом Σ появляется специальный символ $\alpha(\notin\Sigma)$, который означает, что значение данного символа потока данных неизвестно. Очевидно, что задача исправления стираний легче задачи исправления ошибок, так как позиции стираний известны.

Теория алгебраических помехоустойчивых кодов, корректирующих ошибки, хорошо разработана и продолжает развиваться. Возможность коррекции ошибок основана на внесении избыточности в исходные данные. В некоторых работах по теории кодирования (см., например, [2, 6-9]) говорится, что методы борьбы с ошибками могут быть применены для защиты данных от стираний, однако, эти утверждения не сопровождаются подробностями.

Целью данной работы является разработка алгоритмов исправления стираний в данных, закодированных произвольным алгебраическим кодом C исправления ошибок. Разрабатываемые алгоритмы исправляют стирания с использованием штатных декодеров используемого кода C.

Универсальный способ декодирования кодовых слов со стираниями

Рассмотрим произвольный блочный $[n,k,d]_q$-код C, где n – длина кода, k – его размерность, d – минимальное кодовое расстояние, а алфавит кода отождествляется с полем Галуа F_q мощности q. Известные теоретические оценки [2, 7, 9, 11] указывают на способности произвольного $[n,k,d]_q$-кода C по исправлению в одном кодовом слове ошибок:

$$t \le \left\lfloor \frac{d-1}{2} \right\rfloor, \quad (1)$$

и стираний:

$$\rho \le d-1. \quad (2)$$

Все известные декодеры помехоустойчивых кодов могут быть разделены на две группы: универсальные и специальные алгоритмы декодирования. Малочисленная группа универсальных декодеров состоит из алгоритмов, которые могут быть применены к произвольному помехоустойчивому коду определенного типа, специальные – разрабатываются для конкретных кодов, и даже, возможно, к кодам с ограниченным набором параметров. Универсальные декодеры проигрывают специальным по скорости работы и требованиям к используемой памяти устройств. Такие декодеры практически никогда не применяются в реальных системах связи.

Среди группы универсальных декодеров особую роль играет декодер по минимуму расстояния Хемминга [6, 10]. Этот декодер зачастую используется в качестве эталона для оценки корректирующих способностей

специальных декодеров. Для того, чтобы этот декодер мог быть применен для исправления стираний, требуются минимальные изменения. Декодер такого типа может быть применен и без изменений, но изменения могут сократить поиск похожих слов, а, следовательно, улучшить скорость работы. Приведем измененный алгоритм, учитывающий наличие ошибок только типа стирание.

Алгоритм 1. Исправление стираний по минимуму расстояния Хемминга.

Для запуска алгоритма декодирования рассматриваемого кода $[n,k,d]_q$-кода C необходимо предварительно создать список T всех кодовых слов $\bar{c}_i = (c_1, c_2, \dots, c_n) \in F_q^n$, $i = \overline{1, q^k}$. Очевидный способ его создания состоит в последовательном умножении всех возможных q^k информационных слов на порождающую матрицу кода.

Вход: список T, пришедшее по каналу кодовое слово со стираниями $\bar{b} \in F_{q^*}^n$, где $F_{q^*}^n$ – это множество векторов длины n, заданных над алфавитом $F_{q^*} = F_q \cup \{\alpha\}$, где α – символ стирания.

Выход: сообщение об ошибке декодирования или такое кодовое слово $\bar{c} \in \mathrm{C}(F_q^n)$, что если $b_i \neq \alpha$, то $b_i = c_i$ для всех $i = \overline{1, n}$.

Шаг 1. Построить множество $M = \{\bar{\mu}_1, \bar{\mu}_2, \dots\}$, содержащее слова таблицы T, чьи координаты совпадают со всеми нестертыми координатами $\bar{b}$.

Шаг 2. Если $|M| = 1$, то подать на выход алгоритма $\bar{c} = \bar{\mu}_1$, иначе вернуть ошибку декодирования. •

Замечание 1.1. Алгоритм не оценивает превышает ли число стираний в векторе $\bar{b}$ границу (2) в явном виде. Неявно эта проверка происходит при оценке мощности множества M на шаге 2. Введение проверки на число стираний, повредивших вектор $\bar{b}$, сократит время работы алгоритма, но может ухудшить результативность алгоритма, связанную с

неравномерностью распределения кодовых слов в пространстве векторов длины n.

Замечание 1.2. Применение декодера по минимуму расстояния Хемминга в случае стирающего канала не усложняет оригинальный алгоритм. Однако, несмотря на свою простоту этот алгоритм не применим в реальных системах связи: с ростом длины n кодовых слов растет число q^k строк таблицы T, и, как следствие, сложность поиска в ней. Например, для двоичного кода с $n = 1024$ число кодовых слов в таблице T составит $L = 2^{1024} \approx 10^{341}$.

Применение специализированных декодеров, исправляющих ошибки, в стирающем канале

Рассмотрим произвольный помехоустойчивый кодек борьбы с ошибками, основанный на применении блочного $[n, k, d]_q$-кода C, и состоящий из алгоритмов его кодирования и декодирования. Для определенности будем считать, что декодер возвращает кодовые слова. Построим несколько алгоритмов исправления стираний, зависящих от числа стираний, повредивших кодовое слово, и мощности алфавита кода. Если не вносить изменения в имеющиеся алгоритмы кодирования и декодирования данного $[n, k, d]_q$-кода C, то естественным способом применить данный кодек для исправления стираний состоит в том, чтобы заменить стирания в полученных кодовых словах на случайные символы из F_q.

Рассмотрим случай применения $[n, k, d]_q$-кода C, где $q > 2$, и построим два алгоритма, применение которых зависит от числа p стираний, повредивших кодовое слово. Если $p \le t$ (см. (1)), то процедура исправления стираний довольно проста.

Алгоритм 2. Коррекция стираний, число которых не превосходит половины кодового расстояния.

Вход: кодовое слово $\bar{b} = (b_1, \dots, b_n) \in F_{q^*}^n$, где $F_{q^*}^n = F_q \cup \{\alpha\}$, α – символ стирания, p - число стираний в $\bar{b}$, декодер Dec $[n, k, d]_q$-кода C.

Выход: сообщение об ошибке декодирования или такое кодовое слово $\bar{c} \in \mathrm{C}(F_q^n)$, что если $b_i \neq \alpha$, то $b_i = c_i$ для всех $i = \overline{1, n}$.

Шаг 1. Если $p > t$ (см. (1)), то выдать сообщение об ошибке декодирования и завершить работу алгоритма.

Шаг 2. Создать вспомогательный вектор $\bar{r} = (0, 0, \dots, 0) \in F_q^n$. В цикле по $i = \overline{1, n}$: если $b_i \neq \alpha$, то $r_i := b_i$.

Шаг 3. Подать на выход алгоритма $\bar{c} = Dec(\bar{r})$. •

Построенный алгоритм корректен в силу того, что кодовое слово, поступающее на вход алгоритма Dec, повреждено не более чем t ошибками. Такое число ошибок гарантировано исправляется декодером кода C.

Если для числа p стираний, повредивших кодовое слово справедливо $p \leq \rho$ (см. (2)), то для их исправления можно использовать алгоритм 3.

Алгоритм 3. Исправление стираний, число которых достигает ρ.

Вход: кодовое слово $\bar{b} = (b_1, \dots, b_n) \in F_{q^*}^n$, поврежденное стираниями; p - число стираний в $\bar{b}$, декодер Dec $[n, k, d]_q$-кода C.

Выход: сообщение об ошибке декодирования или такое кодовое слово: $\bar{c} \in \mathrm{C}(F_q^n)$, что если $b_i \neq \alpha$, то $b_i = c_i$ для всех $i = \overline{1, n}$.

Шаг 1. Если $p > d - 1$ (см. (2)), то выдать сообщение об ошибке декодирования и завершить работу алгоритма.

Шаг 2. Номера i позиций $\bar{b}$, таких что $b_i = \alpha$, упорядочить по возрастанию и поместить во множество $e = \{e_1, e_2, \dots, e_p\}$.

Шаг 3. Построить все возможные последовательности символов из F_q длины p: $L = \{l_1, \dots, l_{q^p}\}$.

Шаг 4. Для каждого $l_i = (l_1^i, l_2^i, \dots, l_p^i) \in L$ выполнить:

Шаг 4.1. построить вектор $\bar{r}_i = (r_1^i, r_2^i, \dots, r_n^i) \in F_q^n$, где во все коэффициенты r_k^i, для которых $k \in e$, помещены элементы l_i в их естественном порядке, а в коэффициенты r_k^i, для которых $k \notin e$, помещены элементы b_k входного вектора $\bar{b}$.

Шаг 4.2. Используя стандартный декодер $Dec()$ кода C, декодировать $\bar{r}_i \in F_q^n$: $\bar{z}_i = (z_1^i, z_2^i, \dots, z_n^i) = Dec(\bar{r}_i)$. Если для любого $k(\notin e)$: $z_k^i = b_k$, то завершить алгоритм с результатом $\bar{z}_i$, иначе переход на шаг 4. •

Максимальное число стираний, которыми может быть поврежден входной кодовый вектор $d-1$ (см. (2)). Из теории помехоустойчивого кодирования известно (см., например, [10]), что для пары любых кодовых слов $\bar{c}_1$, $\bar{c}_2$ произвольного $[n,k,d]_q$-кода C выполняется: $d(\bar{c}_1, \bar{c}_2) \geq d$, где $d()$ – функция, определяющая расстояние Хемминга между ее аргументами. Таким образом, на шаге 4.2 во сравнения позиций восстановленного вектора $\bar{z}_i$ с неповрежденными позициями входного вектора $\bar{b}$ происходит проверка корректности восстановления $\bar{z}_i$. Если $\bar{z}_i$ и $\bar{b}$ отличаются хотя бы в одном нестертом коэффициенте, то $\bar{z}_i$ – ошибочно восстановленный вектор, и алгоритм продолжает свою работу. Так как в алгоритме испытываются все возможные варианты заполнения стертый позиций элементами F_q, то, очевидно, что верный вариант $\bar{z}_i$ будет найден в ходе работы алгоритма, что обеспечивает корректность алгоритма.

Оценим количество итераций алгоритма 3, необходимых для успешного декодирования кодового слова $[n,k,d]_q$-кода C в алгоритме 3. Вероятность получить верный ответ на j-той итерации шага 4:

$$\rho_j = \frac{1}{q^{e-j+1}} \sum_{i=e-t}^{e} (q-1)^{e-i} * C_e^{e-i}, \quad (3)$$

где e – количество стираний ($t < e \leq d-1$), t определяется формулой (1). Тогда вероятность того, что алгоритм успешно восстанавливает кодовое слово после r-ой итерации определяется равенством:

$$P = 1 - \prod_{i=1}^{r}(1 - \rho_i).$$

Утверждение. Равенство (3) корректно.

Доказательство. Вероятность успешного декодирования задается отношением числа хороших вариантов комбинаций подстановки к общему числу всех возможных подстановок в стертые позиции кодового слова (см. шаг 3 алгоритма). Хорошими комбинациями подстановки считаем те, которые позволяют алгоритму найти искомое кодовое слово и закончить работу.

Рассмотрим знаменатель (3). q^e – это число возможных вариантов заполнения e стираний элементами алфавита F_q. Если алгоритм выполняет j-тую итерацию 4 шага, то это означает, что на предыдущих итерациях искомое кодовое слово найдено не было. Следовательно, общее количество оставшихся вариантов задается выражением $q^e - j + 1$, где j – номер итерации, при этом нумерация итераций начинается с единицы.

Обратимся к числителю (3), в нем вычисляется количество хороших исходов. В таких исходах подстановка l_i (шаг 4 алгоритма) содержит от $(e - t)$ до e позиций с верно «угаданными» значениями. Рассмотрим, как организованы слагаемые суммы в (3). На произвольной позиции вектора $r \in F_q^n$ может находиться q различных значений. Если угаданы i позиций вектора, то остальные $(e - i)$ позиций неверны, и существует $(q-1)^{e-i}$ вариантов заполнения этих позиций. Существует C_e^{e-i} различных способов расположить неверно угаданные позиции в r.•

Пример 1. Рассмотрим $[20, 10, 7]_3$-код. Вероятности успешного декодирования кодового слова, поврежденного 6 стираниями (см. (2)), с помощью алгоритма 3 представлены в таблице 1. Общее количество комбинаций подстановок на стертые позиции составляет 729.

Таблица 1. Вероятности успешного декодирования алгоритмом 3 $[20, 10, 7]_3$- кода C.

№ итерации	1	2	3	4	5	6	7	8	9	10	11
Вероятность декодирования	0.32	0.54	0.69	0.79	0.86	0.9	0.93	0.96	0.97	0.98	0.99

Пример 2. Рассмотрим $[31, 17, 10]_9$-код. Вероятности успешного декодирования кодового слова, поврежденного 9 стираниями (см. (2)), с помощью алгоритма 3 представлены в таблице 1. Общее количество возможных вариантов заполнения стертых позиций составляет 387 420 489.

Таблица 2. Вероятности успешного декодирования алгоритмом 3 $[31, 17, 10]_9$- кода C.

№ итерации	1	2	3	4	5	…	57	…	114	…	287
Вероятность декодирования	0.012	0.024	0.036	0.048	0.059	…	0.501	…	0.751	….	0.97

Алгоритм 3 может быть улучшен для случая двоичных кодов.

Алгоритм 4. Исправление стираний в двоичных данных.

Вход: кодовое слово $\bar{b} = (b_1, \dots, b_n) \in F_{2^*}^n$, искаженное стираниями; p - число стираний в $\bar{b}$, декодер Dec $[n, k, d]_q$-кода C.

Выход: сообщение об ошибке декодирования или такое кодовое слово $\bar{c} \in \mathrm{C}(F_2^n)$, что если $b_i \neq \alpha$, то $b_i = c_i$ для всех $i = \overline{1, n}$.

Шаг 1. Если $p > d - 1$, то выдать сообщение об ошибке декодирования и завершить работу алгоритма.

Шаг 2. Позиций стертых символов записать в список $e = \{e_1, e_2, \dots, e_p\}$.

Шаг 3. $\bar{r} := \bar{b}$, во все $r_i \in \bar{r}$, $i \in e$, записать случайные значения из F_2.

Шаг 4. Вычислить $\bar{z} = (z_1, \dots, z_n) = Dec(\bar{r})$ $(\in F_2^n)$.

Шаг 5. Если для всех k, $k \notin e$: $z_k = b_k$, то завершить алгоритм и вернуть $\bar{z}$ в качестве результата. Иначе, инвертировать все позиции вектора $\bar{r} \in F_2^n$, номера которых лежат в множестве e. Перейти на шаг 4. •

Некоторые замечания о таком способе декодирования приведены, например, в [8]. Сформулируем и докажем лемму о корректности алгоритма 4.

Лемма. Пусть C – некоторый алгебраический помехоустойчивый $[n, k, d]_2$-код, гарантированно исправляющий t ошибок (см. (1)) в произвольном кодовом слове. Пусть на вход алгоритма 4 поступает $\bar{b} \in F_{2^*}^n$ – кодовое слово из C, поврежденное p стираниями, где $p \leq d - 1$, тогда алгоритм выдает в результате своей работы кодовое слово $\bar{c} \in C$, которое совпадает с $\bar{b}$ на всех нестертых позициях.

Доказательство. На шаге 1 алгоритм прекращает свою работу, если число p стираний, повредивших входной вектор, превосходит границу (2), иначе алгоритм переходит на следующий шаг. Рассмотрим случай $p = d - 1$. Если следующие рассуждения верны для этого равенства, то они будут верны и для случаев $p < d - 1$. На шаге 3 во все стертые позиции входного вектора помещаются случайные элементы поля F_2. Пусть u, $0 \leq u \leq p$, из них оказались неверно «угаданными» значениями, тогда $p - u$ значений окажутся «угаданы», т.е. совпадать с соответствующими значениями исходного кодового вектора. Очевидно, что справедливо неравенство $\min(u, p - u) \leq p/2$. Если $p - u \leq p/2$, то алгоритм успешно закончит свою работу за один проход шагов 4–5, иначе, если $u \leq p/2$, то алгоритм успешно закончит свою работу за второй проход шагов 4-5.

Проверка корректности декодирования и принятие решения о необходимости второй итерации шага 4 происходит на шаге 5. Корректность шага 5 основана на свойстве минимального кодового расстояния, а именно,

неверно декодированное слово отличается не менее, чем в одной позиции от слова, принятого по каналу. •

Обсуждение построенных алгоритмов

Построенные алгоритмы 1–4 позволяют применять коды исправления ошибок, а также спроектированные для них кодеры и декодеры для исправления стираний. Эти алгоритмы можно рассматривать, как надстройку над произвольными декодерами кодов исправления ошибок (см. рис. 2).

Рис. 1. – Схема передачи данных по каналу с защитой от ошибок в случае стираний

Можно понизить сложность алгоритмов 3–4, если используемые внутри них декодеры $Dec()$ используют какие-либо методы проверки принадлежности произвольного слова к коду (например, через вычисление синдрома [10, 11]). В этом случае можно не проводить полный процесс декодирования на каждой итерации цикла, а сразу отбрасывать слова, не принадлежащие рассматриваемому коду.

Построенные алгоритмы могут быть полезны, так как существует большое количество декодеров кодов, исправляющих ошибки, реализованных аппаратно, и, используя построенные алгоритмы, можно адаптировать их применение для случая стирающих каналов. Построенные

алгоритмы могут выполнять несколько запусков используемых декодеров, что отрицательно влияет на скорость работы алгоритмов. Однако необходимо отметить, что в алгоритмах 2–4 проводится несколько независимых друг от друга итераций декодирования, следовательно они могут быть распараллелены. Более того, естественным кажется применять алгоритмы 2 и 3 совместно, переключаясь между ними в зависимости от числа стираний, поразивших кодовое слово. Основной недостаток построенных алгоритмов - то, что они никаким образом не используют знание номеров стертых позиций.

Заключение

В работе построены алгоритмы исправления стираний, в основе которых лежит использование декодеров кодов, исправляющих ошибки. При этом не предполагается внесение каких-либо изменений в используемые декодеры, что означает, что построенные алгоритмы могут быть применены в ситуациях, когда необходимо исправлять стирания, но в наличии нет специализированных декодеров для стираний или их аппаратных реализаций. В работе показана корректность построенных алгоритмов, получены некоторые оценки. Необходимо отметить, что исходя из описанных недостатков построенных алгоритмов, можно сделать вывод, что следует развивать специальные методы, ориентированные на исправления стираний, например, [4-5]. Такие методы лишены описанных выше недостатков и могут обладать более простыми алгоритмами кодирования и декодирования.

Литература

1. Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи; пер. с англ. под ред. Б.С. Цыбакова // М.: Радио и связь, 1987. 392 с.

2. Рацеев С.М., Череватенко С.М. Об алгоритмах декодирования обобщенных кодов Рида-Соломона на случай ошибок и стираний // Вестник Самарского университета. Естественнонаучная серия. 2021. Т. 27 №2. С. 7–15.

3. Костюков А. С., Башкиров А. В., Никитин Л. Н., Бобылкин И. С., Макаров О. Ю. Помехоустойчивое кодирование в современных форматах связи // Вестник ВГТУ. 2019. №2. С. 132–139.

4. Айдаркин Е. Е., Могилевская Н. С. Экспериментальное исследование корректирующей способности матричного метода равновесных столбцов защиты данных от стираний // Компьютерная оптика. 2022. Т. 46. № 5. С. 840-847. DOI 10.18287/2412-6179-CO-1122. – EDN XVMPHM.

5. Luby M., Mitzenmacher M., Shokrollahi M. A., Spielman D. Efficient erasure correcting codes // IEEE Transactions Information Theory. 2001. Vol. 47, pp. 569–584.

6. Колесник В. Д. Кодирование при передаче и хранении информации (Алгебраическая теория блоковых кодов) : учебное пособие // М.: Высшая школа. 2009. 549 с. ISBN 978-5-06-005917-5.

7. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки // М.: Мир, 1976. 594 с.

8. Tomlinson M., Tjhai C. J., Ambroze M. A., Ahmed M., Jibril M. Error-correction coding and decoding: bounds, codes, decoders, analysis and applications // Springer International Publishing. 2017. 522 p.

9. Вернер М. Основы кодирования. Учебник для ВУЗов // Мир программирования. М.: Техносфера, 2006. ISBN: 5-94836-019-9.

10. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение // М.: Техносфера. 2006. 320 с.

11. Деундяк В.М., Маевский А.Э., Могилевская Н.С. Методы помехоустойчивой защиты данных // Ростов-на-Дону: Издательство Южного федерального университета. 2014. 309 с.

References

1. Clark J., Kane J. Codirovanie s ispravleniem oshibok v sistemah cifrovoy svyazy; per. s angl. pod red. B.S. Cybankova [Error-correction coding for digital communiations]. M.: Radio I Svyaz. 1987. 392 p.

2. Ratseev S.M., Cherevatenko O.I. Vestnik Samarskogo Universiteta. Estestvennonaunchnaya seria. 2021. T. 27, №2. Pp. 7-15.

3. Kostyukov A.S., Bashkyrov A.V., Nikitin L.N., Bobylkin I.S., Makarov O.U. Vestnik VGTU. 2019. №2. Pp. 132 – 139.

4. Aydarkin E.E., Mogilevskaya N.S. Computernaya optika. 2022. T. 46. № 5. Pp. 840-847. DOI 10.18287/2412-6179-CO-1122. EDN XVMPHM.

5. Luby M., Mitzenmacher M., Shokrollahi A., Spielman D. IEEE Transactions Information Theory. 2001. Vol. 47. Pp. 569–584.

6. Kolesnik V.D. Kodirovanie pri peredache I chranenii informatsii (Algebraicheskaya teoria blockovyh codov) : uchebnoe posobie [Coding in the transmission and storage of information (Algebraic theory of block codes): study guide] M.: Vyshaya Shkola. 2009. 549 p. ISBN 978-5-06-005917-5.

7. Piterson Y., Yeldon A. Cod'y, ispravlayushie oshibki [Error-correcting codes] M.: Mir. 1976. 594 p.

8. Tomlinson M., Tjhai C. J., Ambroze M. A., Ahmed M., Jibril M. Error-correction coding and decoding: bounds, codes, decoders, analysis and applications. Springer International Publishing. 2017. Pp. 522.

9. Verner M. Osnovy kodirovanya. Uchebnik dlya VUZov. [Coding Basics. Textbook for universities] Mir programmirovanya. M.: Tehnosfera. 2006. ISBN: 5-94836-019-9.

10. Morelos-Saragosa R. Iskusstvo pomehoustoichivogo kodirovania. Metody, algoritmy, primenenie. [The art of error-correcting codes. Methods, algorithms, applications.] M.: Tehnosfera. 2006. 320 p.

11. Deundyak V.M., Maevskiy A.E., Mogilevskaya N.S. Metody pomehoustoychivoy zachity dannykh [Methods of noise-resistant data protection]. Rostov-na-Donu: Izdatelstvo Uznogo federal'nogo universiteta. 2014. 309 p.