

В.А. ЛИТВИНОВ, С.Я. МАЙСТРЕНКО, Ю.Г. ПИЛИПЕНКО

ИСПРАВЛЕНИЕ ОШИБОК ПОЛЬЗОВАТЕЛЯ НА ОСНОВЕ СОВМЕСТНОГО ПРИМЕНЕНИЯ ПОМЕХОЗАЩИТНЫХ КОДОВ И ВИРТУАЛЬНОГО СЛОВАРЯ ДОПУСТИМЫХ СЛОВ

Abstract: The described model of the automatic error correction is based on the joint application of the code methods and method of the inverse distortion for increasing validity of the input information and saving labor of error correction. The combined algorithms of automatic and semiautomatic error correction are considered. For the proposed algorithms, probabilistic estimations of the automatic, semiautomatic, manual and false correction, which can be used while deciding which algorithm of adjustment to choose, are analytically derived.

Key words: user errors, automatic correction, data reliability, error-correcting coding.

Аноація: Описана модель автоматичного виправлення помилок ґрунтується на сумісному використанні кодових методів та методу зворотних спотворень для підвищення достовірності вхідної інформації й зниження трудомісткості виправлення помилок. Розглядаються комбіновані алгоритми автоматичної та напіваавтоматичної корекції помилок. Для запропонованих алгоритмів аналітично отримані ймовірнісні оцінки автоматичної, напіваавтоматичної, ручної та хибної корекції, які можуть використовуватись при прийнятті рішень відносно вибору алгоритму корекції.

Ключові слова: помилки користувача, автоматична корекція, достовірність даних, перешкодозахисні коди.

Аннотация: Описанная модель автоматического исправления ошибок основана на совместном применении кодовых методов и метода обратных искажений для повышения достоверности входной информации и снижения трудоемкости исправления ошибок. Рассматриваются комбинированные алгоритмы автоматической и полуавтоматической коррекции ошибок. Для предлагаемых алгоритмов аналитически получены вероятностные оценки автоматической, полуавтоматической, ручной и ложной коррекции, которые могут быть использованы при принятии решений относительно выбора алгоритма корректировки.

Ключевые слова: ошибки пользователя, автоматическая коррекция, достоверность данных, помехозащитные коды.

1. Введение

Известны и описаны в литературе методы помехозащитного кодирования, предназначенные для повышения достоверности входной алфавитно-цифровой информации путем автоматического исправления ошибок пользователя. Наиболее полный обзор таких методов приведен в [1]. Большинство этих методов ориентировано на автоматическое исправление однократных ошибок типа замены значения одного символа и обнаружение с последующим «ручным» исправлением ошибок иных классов.

Помехозащитные коды с исправлением ошибок (КИО) применяются для защиты информации в каналах связи, где вероятность появления ошибок, кратности более 1, значительно, на несколько порядков меньше вероятности однократной ошибки. Поэтому относительное количество не автоматически исправленных ошибок сравнительно невелико. Для информации, вводимой пользователем, положение иное. Здесь, как известно, ошибки типа, например, пропуска символа имеют кратность от 1 до n в зависимости от позиции пропущенного символа, а вероятность такой ошибки не намного меньше вероятности однократной транскрипции. По этим причинам относительное количество вручную исправляемых ошибок здесь может быть значительным.

Метод, позволяющий в этих случаях уменьшить долю ошибок, исправляемых вручную, и снизить тем самым общую трудоемкость ввода и корректировки информации, заключается в расширении штатной процедуры декодирования КИО за счет привлечения метода обратных искажений («вариаций» ошибочного слова) и проверки корректности значения вариации по

словарю допустимых слов [2]. При отсутствии реального словаря в качестве такового может быть использован *виртуальный* словарь (см. далее).

В статье решается задача построения модели и оценки существенных характеристик метода.

2. Постановка задачи и алгоритм декодирования и исправления ошибок

Введем следующие обозначения и допущения:

q, n_u, n_k – соответственно алфавит, количество информационных символов, количество контрольных символов входного слова, $n_u + n_k = n$;

$\mathbf{E} = \{E_i\}$ – множество классов возможных ошибок;

p_i – относительное количество ошибок класса E_i в потоке вводимых слов;

\mathbf{E}_k^1 – подмножество классов ошибок, на автоматическое исправление которых ориентирован конкретный КИО, и назовем их первично корректируемыми ошибками;

\mathbf{E}_k^2 – подмножество классов ошибок, идентифицируемых виртуальным словарем (вторично корректируемые ошибки);

T – виртуальный словарь – множество слов, удовлетворяющих условию отсутствия ошибки, определенному контрольным соотношением (алгоритмом кодирования-декодирования) используемого КИО. Мощность T не превышает q^n ;

$r = \frac{1}{q^{n_k}}$ – коэффициент избыточности кода, определяющий относительное количество

всевозможных комбинаций значений n символов, не удовлетворяющих условию вхождения в T .

Примем допущение, что значения вариаций ошибочного слова случайно распределены среди q^n всевозможных значений слов. В этом случае процесс генерации вариаций и их сравнения с T мы можем описать моделью испытаний Бернулли, для которой вероятность $P(g, r, V)$ случайного совпадения в точности g вариаций из V проверяемых, при условии, что вероятность одного случайного совпадения равна r определяется известной формулой биномиального распределения:

$$P(g, r, V) = C_V^g r^g V^{V-g}.$$

Общая схема комбинированной процедуры включает два этапа.

1. Декодирование слова (кода) стандартной процедурой, присущей используемому КИО.

1.1. Идентификация слова как правильного или ошибочного.

1.2. Если в слове имеется ошибка, принадлежащая множеству \mathbf{E}_k^1 , то она обнаруживается и автоматически исправляется. Иначе, слово идентифицируется как неисправимо ошибочное (в смысле возможностей кода). Переход к этапу 2.

2. Генерация V вариаций в классах ошибок, принадлежащих множеству E_k^2 , и проверка каждой вариации слова на принадлежность T (т.е., по сути, многократное повторение этапа 1 для генерируемых вариаций). В зависимости от результатов проверки возможно автоматическое или полуавтоматическое исправление в соответствии с алгоритмами, описанными в [2].

При этом возможны следующие финальные исходы.

R_{AK} , $R_{ПАК}$ – ошибка правильно идентифицирована и исправлена автоматически (вероятность P_{AK}) или полуавтоматически, т.е. с подтверждением пользователя (вероятность $P_{ПАК}$);

R_{PK} – идентифицировать ошибку не удастся и она исправляется полностью „вручную” (вероятность P_{PK});

$R_{ЛК}$ – ошибка идентифицирована неверно и исправлена ложно (вероятность $P_{ЛК}$).

Определение этих вероятностей и является целью построения следующих моделей. Под вероятностями здесь и далее понимается относительное количество исходов, „благоприятных” в определенном конкретном смысле.

3. Общая модель

3.1. Вероятностные характеристики метода (значения P_{AK} , P_{PK} , $P_{ЛК}$) для автоматической вторичной корректировки (алгоритм А) определяет вероятностно-логический граф, приведенный на рис. 1. За основу принят алгоритм однозначной корректировки вторично корректируемой ошибки [2]. В этом случае автоматическое исправление ошибок E_k^2 производится только при единственном совпадении вариации со словарем (т.е. при отсутствии синонимов).

Последовательность x дуг графа означает совместное наступление x независимых событий; финальная вероятность последнего события равна произведению вероятностей, приписанных каждой дуге; разветвление y дуг означает наступление одного из y несовместных событий; суммарная вероятность событий по y разветвляющимся дугам, равная сумме соответствующих вероятностей, должна быть равна 1, как полная группа событий. Сумма финальных вероятностей по листьям дерева должна быть равна 1, а для искомых вероятностей P должны выполняться условия:

$$P_{AK} + P_{PK} + P_{ЛК} + P_{НО} = 1;$$

$$P_{AK}, P_{PK}, P_{ЛК}, P_{НО} \in [0,1].$$

Таким образом, приведенный граф отражает структуру финальных вероятностей независимых событий.

На рис. 1 приняты следующие обозначения событий и их вероятностей:

S_0 – ошибка произошла;

S_1 – ошибка $\in \mathbf{E}_k^1$, вероятность этого события π_1 равна относительному количеству первично обнаруживаемых ошибок P_1 ;

$S_2 = \bar{S}_1$ – ошибка $\notin \mathbf{E}_k^1$ с вероятностью $\pi_2 = 1 - P_1$;

S_{21} – ошибка обнаружена с вероятностью $\pi_{21} = 1 - r$;

$S_{22} = \bar{S}_{21}$ – ошибка не обнаружена с вероятностью $\pi_{22} = r$;

S_{211} – ошибка принята для идентификации и исправления с вероятностью $\pi_{211} = 1 - \beta$;

$S_{212} = \bar{S}_{211}$ – ошибка ложно исправлена кодом с вероятностью $\pi_{212} = \beta$;

S_{2111} – ошибка $\in \mathbf{E}_k^2$ с вероятностью $\pi_{2111} = P_2$;

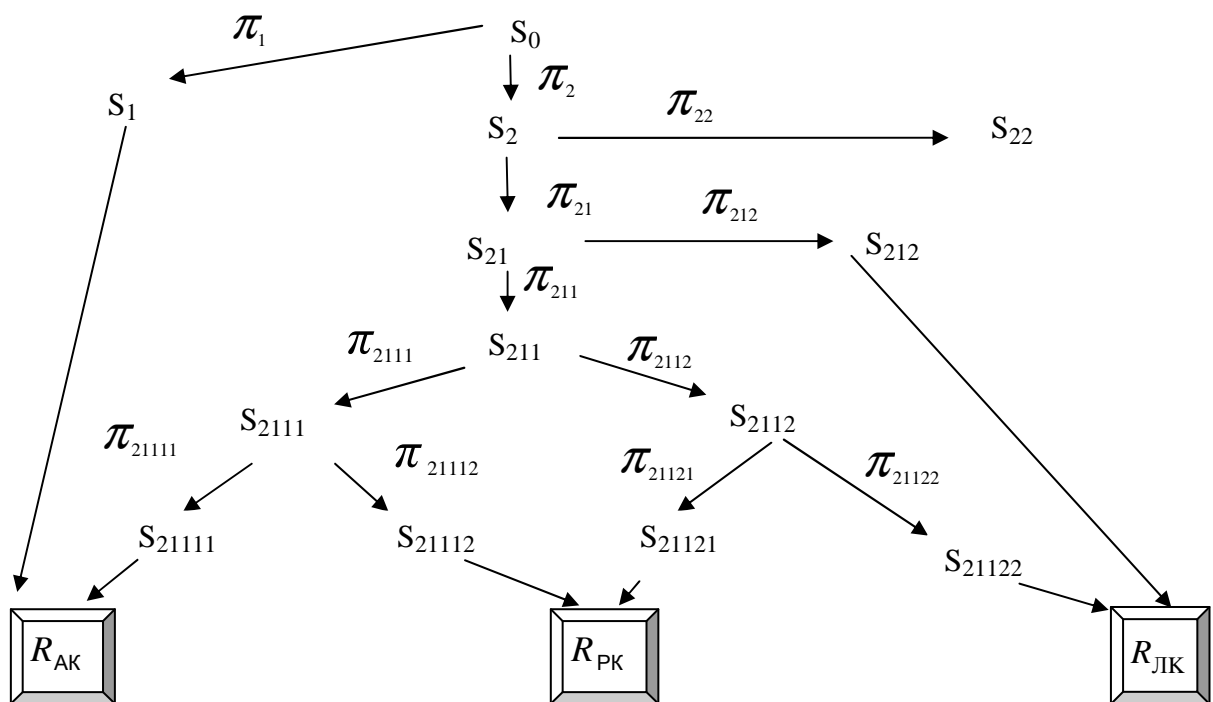


Рис. 1. Граф частных событий для алгоритма автоматического исправления

$S_{2112} = \bar{S}_{2111}$ – ошибка $\notin \mathbf{E}_k^2$ с вероятностью $\pi_{2112} = 1 - P_2$;

S_{21111} – ошибка однозначна ($z = 1$) с вероятностью $\pi_{21111} = P(0, r, V - 1) = (1 - r)^{V-1}$;

$S_{21112} = \bar{S}_{21111}$ – ошибка многозначна ($z > 1$) с вероятностью

$$\pi_{21112} = \sum_{g=1}^{V-1} P(g, r, V - 1) = 1 - P(0, r, V - 1);$$

S_{21121} – для ошибки, которая не принадлежит \mathbf{E}_k^2 , не состоялось ни одного случайного совпадения вариаций со словом словаря ($z = 0$) с вероятностью

$$\pi_{21121} = 1 - P(1, r, V) = 1 - rV(1 - r)^V;$$

$S_{21122} = \bar{S}_{21121}$ – ошибка, которая не принадлежит E_k^2 , однозначна ($z = 1$) с вероятностью $\pi_{21122} = P(1, r, V) = rV(1-r)^V$.

С учетом приведенных обозначений и вида графа рис. 1 можно записать такие логические выражения:

$$R_{AK} = S_1 \vee (S_2 \wedge S_{21} \wedge S_{211} \wedge S_{2111} \wedge S_{21111});$$

$$R_{PK} = (S_2 \wedge S_{21} \wedge S_{211}) \wedge ((S_{2111} \wedge S_{21112}) \vee (S_{2112} \wedge S_{21121}));$$

$$R_{ЛК} = (S_2 \wedge S_{21} \wedge S_{211}) \wedge ((S_{2112} \wedge S_{21122}) \vee S_{212}).$$

С учетом того, что все события на рис. 1 взаимно независимы, для вероятностей P_{AK} , P_{PK} , $P_{ЛК}$ можно записать

$$P_{AK} = \pi_1 + \pi_2 \cdot \pi_{21} \cdot \pi_{211} \cdot \pi_{2111} \cdot \pi_{21111};$$

$$P_{PK} = \pi_2 \cdot \pi_{21} \cdot \pi_{211} \cdot (\pi_{2111} \cdot \pi_{21112} + \pi_{2112} \cdot \pi_{21121});$$

$$P_{ЛК} = \pi_2 \cdot \pi_{21} \cdot (\pi_{211} \cdot \pi_{2112} \cdot \pi_{21122} + \pi_{212}).$$

Окончательно имеем

$$P_{AK}^{(1)} = P_1 + (1 - P_1) \cdot (1 - r) \cdot (1 - \beta) \cdot P_2 (1 - r)^{V-1}; \quad (1)$$

$$P_{PK}^{(1)} = (1 - P_1) \cdot (1 - r) \cdot (1 - \beta) \cdot \left\{ P_2 \left[1 - (1 - r)^{V-1} \right] + (1 - P_2) \cdot \left[1 - r \cdot V \cdot (1 - r)^V \right] \right\}; \quad (2)$$

$$P_{ЛК}^{(1)} = (1 - P_1) \cdot (1 - r) \cdot (1 - \beta) \cdot (1 - P_2) \cdot r \cdot V \cdot (1 - r)^V + (1 - P_1) \cdot (1 - r) \cdot \beta. \quad (3)$$

Легко показать, что $P_{AK}^{(1)} + P_{PK}^{(1)} + P_{ЛК}^{(1)} + P_{HO}^{(1)} = 1$.

3.2. Полуавтоматическое исправление ошибок (алгоритм В) внешне (в смысле интерфейса с пользователем) подобно алгоритму Spell-Checker'a текстового редактора (например, Word'a). При отсутствии однозначности корректировки пользователю предлагается для подтверждения до m вариантов исправлений. Если среди них нет правильного варианта, ошибка исправляется вручную. В Word'e предлагаются все установленные возможные варианты исправления, что, впрочем, для определенных классов ошибок тоже не освобождает от полностью ручного исправления.

На рис. 2 приведен фрагмент вероятностно-логического графа алгоритма В, который содержит отличия от алгоритма А.

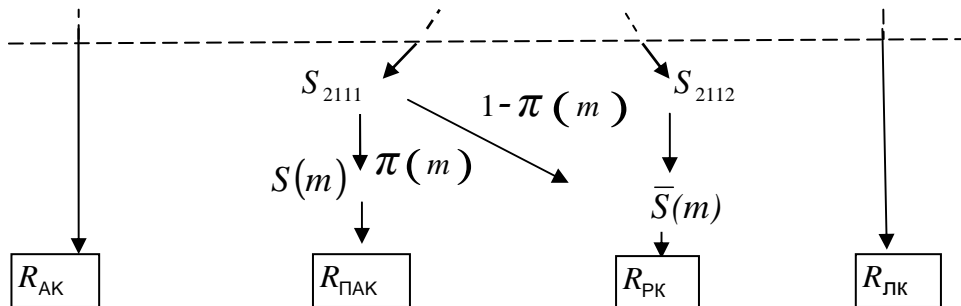


Рис. 2. Фрагмент графа алгоритма полуавтоматического исправления

Здесь приняты следующие дополнительные обозначения:

S_m и \bar{S}_m – соответственно ошибка идентифицирована и не идентифицирована среди m синонимов.

Таким образом, имеем следующие вероятностные оценки:

$$P_{AK}^{(2)} = P_1; \quad (4)$$

$$P_{ПАК}^{(2)} = (1 - P_1) \cdot (1 - r) \cdot (1 - \beta) \cdot P_2 \pi(m); \quad (5)$$

$$P_{PK}^{(2)} = (1 - P_1) \cdot (1 - r) \cdot (1 - \beta) \cdot \{P_2(1 - \pi(m)) + (1 - P_2)\}; \quad (6)$$

$$P_{ЛК}^{(2)} = (1 - P_1) \cdot (1 - r) \cdot \beta \quad (7)$$

и соответственно $P_{AK}^{(2)} + P_{ПАК}^{(2)} + P_{PK}^{(2)} + P_{ЛК}^{(2)} + P_{НО}^{(2)} = 1$.

Вероятность $\pi(m)$ равна [3]:

$$\pi(m) = \sum_{g=0}^{m-1} C_{V-1}^g \cdot r^g \cdot (1-r)^{V-g-1} + \sum_{g=m}^{V-1} \frac{m}{g+1} \cdot C_{V-1}^g \cdot r^g \cdot (1-r)^{V-g-1}. \quad (8)$$

3.3. Для дополнительной оценки эффективности алгоритмов автоматического и полуавтоматического исправления введем понятие ориентировочной трудоемкости исправления ошибок F_{AK} , $F_{ПАК}$, F_{PK} и определим ее как среднее количество операций сравнения и нажатий на клавиши, которые выполняет пользователь при соответствующих финальных исходах. Среднюю трудоемкость F определим как взвешенную сумму:

$$F = P_{AK} F_{AK} + P_{ПАК} F_{ПАК} + P_{PK} F_{PK}. \quad (9)$$

Алгоритм А

Очевидно, $F_{AK}^{(A)} = 0$. Далее, предполагая, что при полностью ручном исправлении пользователь в среднем «обрабатывает» половину символов слова, положим $F_{PK}^{(A)} = \frac{n}{2}$.

Алгоритм В

Оценим величины $F_{PK}^{(B)}$ и $F_{ПАК}^{(B)}$ для максимального значения $m = z$, где z – полное количество синонимов. При ручном исправлении пользователю приходится отвергнуть все z синонимов и в среднем обработать $\frac{n}{2}$ символов, т.е. $F_{PK}^{(B)} = z + \frac{n}{2}$, так как в этом случае все синонимы образуются в результате исключительно случайных совпадений:

$$F_{PK}^{(B)} = \sum_{g=0}^V gP(g, r, V) + \frac{n}{2} = rV + \frac{n}{2}. \quad (10)$$

При полуавтоматическом исправлении ошибка принадлежит E_k^2 , и, следовательно, один из синонимов заведомо совпадает с T , т.е. является не случайным. Поэтому, с учетом предыдущих рассуждений,

$$F_{ПМК}^{(B)} = 1 + \sum_{g=0}^{V-1} \frac{g+2}{2} P(g, r, V) = 1 + \frac{1}{2} r(V-1). \quad (11)$$

4. Пример приложения моделей

Рассмотрим результаты приложения моделей к совместному использованию помехоустойчивого удлиненного кода Рида-Соломона [1] и метода анализа обратных вариаций.

Рассматриваемый код является одним из кодов над полем Галуа $GF(p^m)$, где p – целое.

Длина кода $n_u + n_k = p + 1$, $n_k = 2$.

В примере, приведенном в [1] для иллюстрации алгоритмов формирования и декодирования кода, $p = 11$, исходное информационное слово $A_n = 2500000001$; контрольные символы $a_{01} = 3, a_{02} = 0$; закодированное слово $A_{n+n_k} = 25000000130$. Здесь значения информационных символов подобраны так, чтобы при вычислении избыточных символов не выйти за пределы десятичного алфавита. В общем случае, конечно, избыточные символы могут принимать одно из p значений и, следовательно, должны представляться (или могут интерпретироваться) как числа в системе счисления с основанием p . В данном примере это должно быть 11-ричное (или 16-ричное) основание. В связи с этим отметим, что двухразрядное число в системе счисления с основанием $p \leq 33$ можно представить трехразрядным десятичным числом, т.к. $33^2 < 10^3 < 34^2$. Поэтому для $p \leq 33$ два избыточных символа в алфавите p можно заменить тремя десятичными цифрами. Например, для $p = 11$, пару значений $\alpha_{01} = 10$ и $\alpha_{02} = 07$ можно интерпретировать как двухразрядное 11-ричное число $10/07$, десятичный эквивалент которого равен 117. Таким образом, для цифрового кода ($q = 10$) может быть принято $p = 31$ при трех реальных избыточных символах и двух виртуальных (при соответствующем усложнении алгоритмов декодирования). Аналогичные рассуждения применимы и к другим q, p .

Описанный код полностью обнаруживает и исправляет однократные транскрипции.

Примем $E = \{E_1, E_2, E_3, E_4, E_5, E_6\}$; смысл классов ошибок и значений связанных с ними параметров приведен в табл. 1.

Таблица 1. Классы ошибок

Класс ошибок E_i	Характер ошибок	Вероятность появления, p_i [4]	Количество генерируемых вариаций
E_1	Однократные транскрипции	0,5557	$(q-1) \cdot n$
E_2	Вставка символа	0,1567	n
E_3	Выпадение символа	0,1204	$q \cdot (n+1)$
E_4	Транспозиция соседних символов	0,0664	$n - 1$
E_5	Двукратные транскрипции	0,0322	$(q-1)^2 \cdot C_n^2 - n + 1$
E_6	Множественные ошибки	0,0686	–

Для рассматриваемого кода $\mathbf{E}_k^1 = \{E_1\}$, $\mathbf{E}_k^2 = \{E_i\}$.

Результаты иллюстративных расчетов для моделей автоматического (алгоритм А) и полуавтоматического (алгоритм В) исправления приведены в табл. 2 и 3 соответственно. Сводная таблица результатов, отражающая вероятностные характеристики вариантов декодирования и соответствующие показатели трудоемкости для значения $r = 0,001, q = 37, n = 10$, приведена в табл. 4. В данных табл. 4 учтено, что для варианта декодирования, ограниченного рассматриваемым кодом, как видно из рис. 1, $P_{AK} = P_1$; $P_{PK} = 1 - P_1$; $P_{PK} = (1 - P_1)(1 - r)(1 - \beta)$; $P_{JK} = (1 - P_1)(1 - r)\beta$; $F = P_{PK} \frac{n}{2}$.

При расчетах принято $\mathbf{E}_k^{21} = \{E_4\}$; $\mathbf{E}_k^{22} = \{E_2, E_3, E_4\}$; $\beta = r$ при допущении о псевдослучайном характере распределения результатов декодирования кода, искаженного ошибкой, не принадлежащей к $\mathbf{E}_k^1 = \{E_1\}$.

Таблица 2. Иллюстративные расчеты для алгоритма А

\mathbf{E}_k^2	$r = 0,001, q = 10, n = 10$			$r = 0,001, q = 37, n = 10$		
	P_{AK}	P_{PK}	P_{JK}	P_{AK}	P_{PK}	P_{JK}
\mathbf{E}_k^{21}	0,62144	0,37431	0,0038077	0,62144	0,37431	0,0038077
\mathbf{E}_k^{22}	0,85731	0,13040	0,011850	0,77977	0,19136	0,028427

Таблица 3. Иллюстративные расчеты для алгоритма В

m	\mathbf{E}_k^2	$P_{AK} = 0,55570, P_{JK} = 0,001,$ $r = 0,001, q = 10, n = 10$			$P_{AK} = 0,55570, P_{JK} = 0,001,$ $r = 0,001, q = 37, n = 10$		
		P_{IAK}	P_{PK}	$\pi(m)$	P_{IAK}	P_{PK}	$\pi(m)$
1	\mathbf{E}_k^{21}	0,066003	0,37741	0,99601	0,066003	0,37741	0,99601
	\mathbf{E}_k^{22}	0,32177	0,12164	0,93863	0,27926	0,16415	0,81461
2	\mathbf{E}_k^{21}	0,066267	0,37715	0,99999	0,066267	0,37715	0,99999
	\mathbf{E}_k^{22}	0,34194	0,10147	0,99746	0,33444	0,10897	0,97558
4	\mathbf{E}_k^{21}	0,066267	0,37714	1,0000	0,066267	0,37714	1,0000
	\mathbf{E}_k^{22}	0,34281	0,10060	1,0000	0,34274	0,10067	0,99980
Z	\mathbf{E}_k^{21}	0,066267	0,37714	1,0000	0,066267	0,37714	1,0000
	\mathbf{E}_k^{22}	0,34289	0,10060	1,0000	0,34282	0,10060	1,0000

Таблица 4. Сводные характеристики алгоритмов

Алгоритм декодирования	E_k^2	P_{AK}	$P_{ПАК}$	P_{PK}	$P_{ЛК}$	F
Код		0,5557	–	0,4434	0,0004	2,22
Код+ алгоритм А	E_k^{21}	0,6214	–	0,3743	0,0038	1,87
	E_k^{22}	0,7797	–	0,1913	0,0284	0,96
Код+алгоритм В	E_k^{21}	0,5557	0,6626	0,3771	0,0004	1,95
	E_k^{22}	0,5557	0,3428	0,1006	0,0004	0,96

5. Анализ данных

1. Как видно из данных табл. 2, 4, комбинированная процедура декодирования анализом вариаций ошибочного слова и автоматическим исправлением по виртуальному словарю позволяет повысить относительное количество исправляемых ошибок и снизить трудоемкость ввода. Так, в зависимости от выбранного множества E_k^2 , значение P_{AK} может быть повышено от 0,5557 до 0,7797 и более.

Соответственно значение показателя трудоемкости F может быть уменьшено с 2,22 до 0,96. Однако при этом повышается относительное количество ложно корректируемых ошибок. Хотя окончательное решение относительно приемлемости цены (увеличение $P_{ЛК}$), которую стоит платить за снижение F , остается за проектировщиком, можно предположить, что практически использование алгоритма А может иметь смысл лишь для режима off-line, когда подтверждение исправления затруднено.

2. Для полуавтоматического исправления (табл. 3, 4) доля ложно корректируемых ошибок не увеличивается, а значение показателя суммарной трудоемкости F существенно снижается по сравнению с чисто кодовой процедурой декодирования и исправления. При этом снижение трудоемкости тем больше, чем больше классов ошибок включено в E_k^2 . Поэтому может оказаться целесообразным дополнение E_k^2 двукратными транскрипциями и/или специфическими ошибками иных классов, выходящих за рамки упрощенного перечня табл. 1.

3. Сделанные заключения базируются на количественных результатах применения кода Рида-Соломона ($E_k^1 = \{E_1\}$), однако они качественно не меняются и при применении других кодов, например, [5] ($E_k^1 = \{E_1, E_4\}$).

6. Выводы

Дополнение помехозащитного кода с исправлением ошибок методом корректировки по виртуальному словарю допустимых слов позволяет улучшить корректирующие и трудоемкостные характеристики кода. Полученные соотношения (1) – (3) и (4) – (7) позволяют получить ориентировочные оценки, характеризующие результаты применения метода в зависимости от заданных условий и используемого алгоритма и принять соответствующие решения.

Рассмотренный подход может оказаться полезным при выборе решения о корректировке ошибок каналов связи и носителей информации.

СПИСОК ЛИТЕРАТУРЫ

1. Бояринов И.М., Давыдов А.А. Мамедли Э.М., Смеркис Ю.Б. Использование помехоустойчивого кодирования для защиты информации от ошибок оператора // АТ. – 1983. – № 2. – С. 5–48.
2. Кузьменко Г.Є., Литвинов В.А., Майстренко С.Я., Ходак В.І. Алгоритми і моделі автоматичної ідентифікації та корекції типових помилок користувача на основі природної надмірності // Математичні машини і системи. – 2004. – № 2. – С. 134–148.
3. Литвинов В.А., Майстренко С.Я., Ступак Н.Б. Некоторые оценки вероятностных характеристик процесса автоматической идентификации ошибок пользователя на основе эталонного словаря // УСИМ. – 2001. – № 2. – С. 21–24.
4. Литвинов В.А., Крамаренко В.В. Контроль достоверности и восстановления информации в человеко-машинных системах. – Киев: Техніка, 1986. – 200 с.
5. Sethi A.S., Rajaraman V., Kenjale P.S. An error-correcting coding scheme for alphanumeric data // Information Processing Letters. – 1988. – Vol. 7. – P. 72–77.