

РАСЧЕТ КОЛИЧЕСТВА АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ ДЛЯ ПОИСКА ОБРАТНОЙ МАТРИЦЫ И ПОВЫШЕНИЯ НАДЕЖНОСТИ ПРИНЯТЫХ СИМВОЛОВ В АЛГОРИТМЕ ПЕРЕСТАНОВОЧНОГО ДЕКОДИРОВАНИЯ НА ПРИМЕРЕ КОДА ХЭММИНГА (7, 4, 3)

Бакурова А.Д.

Ульяновский государственный технический университет, Ульяновск, РФ

E-mail: bakurova.ad@mail.ru

Перестановочное декодирование групповых систематических помехоустойчивых кодов, в сравнении с другими методами декодирования цифровых данных, позволяет использовать избыточность, заложенную в код, с максимальной эффективностью. При этом решается сложная вычислительная задача поиска эквивалентного кода, необходимого для нахождения вектора ошибок. Наиболее затратной частью алгоритма поиска эквивалентного кода являются матричные преобразования, в частности, поиск обратной матрицы. В ряде работ описывается идея внедрения когнитивной карты декодера, однако отсутствуют расчеты, доказывающие кратность выигрыша и показывающие количественную эффективность предложенного решения. В данной работе будет показан выигрыш по количеству операций при применении когнитивной карты декодера. Также, процедура повышения степени надежности принятых символов с помощью алгоритма Бала позволяет избежать повторных определений перестановок и вычисления эквивалентного кода.

Ключевые слова: перестановочное декодирование, обратная матрица, повышение степени надежности, когнитивная карта декодера, эквивалентный код

Введение

На сегодняшний день активно развиваются системы дистанционного управления, передачи команд управления в реальном времени, машинного обучения, роботизированные системы, и т.д. Применяемые в подобных сетях радиоканалы требуют защиты передаваемых по ним данных от ошибок и влияния внешних деструктивных факторов – попыток перехвата сигнала, срывов процедуры управления [1–4]. Эффективным средством защиты радиоканалов является применение средств перестановочного декодирования (ПД). К тому же, ПД является средством повышения энергетической эффективности систем связи, что является одним из основных направлений развития телекоммуникационных систем.

В связи с отсутствием в цифровой информации избыточности, искажение любого переданного символа приводит к потере достоверности данных. В системах управления фактор достоверности является решающим – в качестве управляющих команд могут выступать исключительно достоверные данные. Теория помехоустойчивого кодирования является мощным средством повышения энергетической эффективности систем связи за счет рационального введения в цифровые данные избыточности [5–9]. Использование классического варианта алгоритма ПД позволяет в значительной мере повысить энергетическую эффективность систем связи, однако, алгоритм содержит этапы,

выполнение которых приводит к временным задержкам, что является существенным недостатком в системах передачи команд управления.

Таким образом, снижение вычислительных затрат приведет к еще большему повышению энергетической эффективности систем связи, а также к значительному временному выигрышу и ускорению работы алгоритмов. Полученные результаты ускорения и снижения затрат алгоритма декодирования востребованы в самых популярных областях – системах дистанционного управления беспилотными летательными аппаратами (БПЛА) различного типа, высокоскоростных оптических линиях связи, работающих с упреждающей коррекцией ошибок в условиях применения сложных видов модуляции и т.д. Описанного результата удастся достичь путем применения когнитивной карты декодера (ККД) [10–12], что хранит множество необходимых для вычисления параметров эквивалентного кода (ЭК) матриц, соответствующих ряду возможных ситуаций, что обеспечивает временной выигрыш при реализации перестановочного декодирования за счет ухода от матричных преобразований в процессе выполнения процедуры декодирования.

Алгоритм перестановочного декодирования без применения когнитивной карты декодера

Классический алгоритм перестановочного декодирования описан в работах [13–15]. Вычисления реализуются по 12-ти шагам.

Шаг 1. Принятие поступающих из канала связи битов и фиксация жестких решений вектора приема $V_{пр}$, на который в канале связи действует вектор ошибок V_e . Каждый соответствующий бит сопровождается оценкой надежности принятия символа λ с помощью значений МРС (мягких решений символов).

Шаг 2. Ранжирование значений МРС в порядке убывания. Слева записываются наиболее надежные символы с высоким значением λ . Данный способ расположения наиболее надежных символов связан с размещением слева единичной матрицы, входящей в состав порождающей матрицы в систематической форме для групповых кодов.

Шаг 3. После выполнения ранжирования символов в соответствии с убыванием МРС произведение биективных преобразований вида $f: V_{\infty} \rightarrow V_{\perp}$, где $V_{\perp}$ – переставленный вектор. На этом шаге также происходит формирование соответствующей биекции матрицы перестановок P .

Шаг 4. Фиксирование в виде информационного вектора $V'_{инф}$ выделенных в векторе $V_{\perp}$ левых k наиболее надежных символов.

Шаг 5. Осуществление вычисления переставленной матрицы G' путем умножения порождающей матрицы исходного кода G на матрицу перестановок P ($G \times P$).

Шаг 6. Выделение левых k столбцов переставленной матрицы G' для получения ключевой матрицы Q размерности $k \times k$. Вычисление определителя матрицы $Q - \Delta$. Если $\Delta \neq 0$, переходим к шагу 7. Если $\Delta = 0$, отказываемся от декодирования и переходим к шагу 2 с выполнением новых перестановок, меняя местами символы с номером k с символом с номером $k+1$. Таким же образом происходит трансформирование матрицы перестановок P .

Шаг 7. Вычисление матрицы миноров MQ для ключевой матрицы Q .

Шаг 8. Нахождение обратной матрицы Q^{-1} . Для этого элементы матрицы Q^{-1} необходимо разделить на значение Δ .

Шаг 9. Приведение переставленной матрицы G' к систематическому виду $G'_{сист}$ с использованием полученной обратной матрицы Q^{-1} .

Шаг 10. Вычисление вектора эквивалентного кода $V_{эКВ}$ путем умножения нового информационного вектора $V'_{инф}$, полученного в ходе проведения процедур четвертого шага, на матрицу $G'_{сист}$.

Шаг 11. Умножение вектора эквивалентного кода $V_{эКВ}$ на транспонированную матрицу перестановок P^T . После этого осуществляется нахождение обратной биекции $f: V_{\infty} \rightarrow V_{\perp}$ и получается эквивалентный переставленный вектор $V_{\perp}^{\infty}$.

Шаг 12. Поразрядное сложение векторов $V_{пр}$ и $V_{\perp}^{\infty}$ для вычисления вектора V_e , характеризующего воздействие ошибок в канале связи во время фиксации битов вектора $V_{пр}$.

Видно, что наиболее затратными в энергетическом и временном плане являются шаги 6-8, на которых осуществляется вычисление определителя ключевой матрицы и нахождение ее обратной матрицы. Далее будет выполнен расчет количества простейших арифметических операций для худшего случая.

Поиск обратной матрицы для нахождения эквивалентного кода

Рассмотрим 6-8 шаги декодирования кода Хэмминга (7, 4, 3). Подсчитаем количество элементарных операций, необходимых для нахождения определителя и обратной матрицы. Ключевая матрица кода Хэмминга (7, 4, 3) имеет формат (4×4) .

Находим определитель матрицы A :

$$A = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}.$$

В качестве примера разложим операцию вычисления определителя матрицы методом разложения по первой строке на простые арифметические операции.

$$\begin{aligned} \det(A) &= \begin{vmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{vmatrix} = \\ &= a \times (-1)^{1+1} \times \begin{vmatrix} f & g & h \\ j & k & l \\ n & o & p \end{vmatrix} + b \times (-1)^{1+2} \times \begin{vmatrix} e & g & h \\ i & k & l \\ m & o & p \end{vmatrix} + \\ &+ c \times (-1)^{1+3} \times \begin{vmatrix} e & f & h \\ i & j & l \\ m & n & p \end{vmatrix} + d \times (-1)^{1+4} \times \begin{vmatrix} e & f & g \\ i & j & k \\ m & n & o \end{vmatrix} = \\ &= a \times (-1)^{1+1} \times \\ &\times \left(f \times (-1)^{1+1} \times \begin{vmatrix} l \\ o & p \end{vmatrix} + g \times (-1)^{1+2} \times \begin{vmatrix} j & l \\ n & p \end{vmatrix} + h \times (-1)^{1+3} \times \begin{vmatrix} j & k \\ n & o \end{vmatrix} \right) + \\ &+ b \times (-1)^{1+2} \times \\ &\times \left(e \times (-1)^{1+1} \times \begin{vmatrix} l \\ o & p \end{vmatrix} + g \times (-1)^{1+2} \times \begin{vmatrix} i & l \\ m & p \end{vmatrix} + h \times (-1)^{1+3} \times \begin{vmatrix} i & k \\ m & o \end{vmatrix} \right) + \\ &+ c \times (-1)^{1+3} \times \end{aligned}$$

$$\begin{aligned}
& \times \left(e \times (-1)^{l+1} \times \begin{vmatrix} j & l \\ n & p \end{vmatrix} + f \times (-1)^{l+2} \times \begin{vmatrix} i & l \\ m & p \end{vmatrix} + h \times (-1)^{l+3} \times \begin{vmatrix} i & j \\ m & n \end{vmatrix} \right) + \\
& + d \times (-1)^{l+4} \times \\
& \times \left(e \times (-1)^{l+1} \times \frac{j}{n} \times \frac{k}{o} + f \times (-1)^{l+2} \times \frac{i}{m} \times \frac{k}{o} + g \times (-1)^{l+3} \times \frac{i}{m} \times \frac{j}{n} \right) = \\
& = a \times (-1)^{l+1} \times (f \times (-1)^{l+1} \times (k \times p - l \times o) + \\
& + g \times (-1)^{l+2} \times (j \times p - l \times n) + h \times (-1)^{l+3} \times (j \times o - k \times n)) + \\
& + b \times (-1)^{l+2} \times (e \times (-1)^{l+1} \times (k \times p - l \times o) + \\
& + g \times (-1)^{l+2} \times (i \times p - l \times m) + h \times (-1)^{l+3} \times (i \times o - k \times m)) + \\
& + c \times (-1)^{l+3} \times (e \times (-1)^{l+1} \times (j \times p - l \times n) + \\
& + f \times (-1)^{l+2} \times (i \times p - l \times m) + h \times (-1)^{l+3} \times (i \times n - j \times m)) + \\
& + d \times (-1)^{l+4} \times (e \times (-1)^{l+1} \times (j \times o - k \times n) + \\
& + f \times (-1)^{l+2} \times (i \times o - k \times m) + g \times (-1)^{l+3} \times (i \times n - j \times m)) = \\
& = 111 \text{ операций.}
\end{aligned}$$

При подсчете числа операций следует учитывать, что по времени исполнения и сложности операции умножения и сложения считаются равными, поскольку умножаются целые числа. Операция умножения на 0 учитывается, поскольку она также выполняется процессором.

Для выполнения операции вычисления определителя матрицы потребуется выполнить 111 элементарных арифметических операций.

Находим обратную матрицу A^{-1} . В случаях, когда $\det(A) \neq 0$, обратная матрица для матрицы A существует. Находим обратную матрицу с использованием матрицы алгебраических дополнений.

$$\begin{aligned}
A^{-1} &= \frac{1}{\det(A)} \times C^T = \\
&= \frac{1}{\det(A)} \times \begin{pmatrix} C_{11} & C_{21} & C_{31} & C_{41} \\ C_{12} & C_{22} & C_{32} & C_{42} \\ C_{13} & C_{23} & C_{33} & C_{43} \\ C_{14} & C_{24} & C_{34} & C_{44} \end{pmatrix},
\end{aligned}$$

где C – союзная матрица для матрицы A .

Приступаем к нахождению алгебраического дополнения для последующего вычисления союзной матрицы C :

$$\begin{aligned}
C_{11} &= (-1)^{1+1} \times \begin{vmatrix} C_{22} & C_{32} & C_{42} \\ C_{23} & C_{33} & C_{43} \\ C_{24} & C_{34} & C_{44} \end{vmatrix} = \\
&= (-1)^{(1+1)} \times \left(C_{22} \times (-1)^{(1+1)} \times \begin{vmatrix} C_{33} & C_{43} \\ C_{34} & C_{44} \end{vmatrix} + \right. \\
& \quad \left. + C_{32} \times (-1)^{(1+2)} \times \begin{vmatrix} C_{23} & C_{43} \\ C_{24} & C_{44} \end{vmatrix} + \right.
\end{aligned}$$

$$\begin{aligned}
& + C_{42} \times (-1)^{(1+3)} \times \begin{vmatrix} C_{23} & C_{33} \\ C_{24} & C_{34} \end{vmatrix} \Big) = \\
& = (-1)^{(l+1)} \times (C_{22} \times (-1)^{(l+1)} \times (C_{33} \times C_{44} - C_{34} \times C_{43}) + \\
& + C_{32} \times (-1)^3 \times (C_{23} \times C_{44} - C_{24} \times C_{43}) + \\
& + C_{42} \times (-1)^{l+3} \times (C_{23} \times C_{34} - C_{24} \times C_{33})).
\end{aligned}$$

Для вычисления одного элемента матрицы C необходимо вычислить алгебраическое дополнение (3×3), выполнив 23 операции. Всего необходимо найти 16 элементов матрицы C , для чего потребуется выполнить 368 операций. Для последнего действия – нахождения обратной матрицы – необходимо умножить союзную матрицу на $1/\det(A)$, для чего потребуется выполнить 17 операций (деление и умножение каждого из 16 элементов на полученное значение).

Для проведения матричных вычислений классического алгоритма перестановочного декодирования кода Хэмминга (7, 4, 3) с шестого по восьмой шаг было выполнено $111+368+17=496$ арифметических операций. При этом извлечение готового результата из априори созданной ККД потребует выполнения всего 3 элементарных операций, которые не связаны с арифметическими действиями. Это способствует ускорению процесса обработки цифровых данных в высокоскоростных системах обмена информацией.

Выигрыш в арифметических операциях при внедрении когнитивной карты декодера

Для сравнения эффективности работы классического алгоритма ПД и алгоритма ПД с ККД, следует сравнить число арифметических операций, необходимых для выполнения матричных преобразований с числом операций перебора внутри ККД.

В работе [14] было установлено, что полное множество комбинаций перестановок для кода Хэмминга (7, 4, 3) включает 35 комбинаций, 20 % из которых приводят к получению нулевого определителя и являются непродуктивными. В случае выпадения непродуктивной перестановки, вместо 111 операций, необходимых для вычисления определителя матрицы, будет выполнено сравнение всего с 7-ю комбинациями, что эквивалентно 7 арифметическим операциям. В случае получения продуктивной перестановки, будет выполнено сравнение с 7-ю непродуктивными перестановками и получен допуск к блоку продуктивных перестановок, где в худшем случае будет выполнено 28 операций сравнения вместо 496 арифметических операций.

Таким образом, выигрыш по количеству элементарных операций составит более 90 %.

Повышение надежности переданных по каналу связи символов с использованием алгоритма Бала

При выполнении процедуры декодирования, отсутствует гарантия наличия достаточно надежных символов для получения ЭК, ввиду чего становится невозможным применение узкого набора перестановок для получения ЭК. Алгоритм Бала – готовое решение повышения надежности символов – можно рассматривать в качестве выбора, как альтернативного поиска новой перестановки.

Алгоритм Бала описывается выражением:

$$L(d_1) \boxplus L(d_2) \approx L(d_1 \oplus d_2) \approx (-1) \times \text{sign}[L(d_1)] \times \text{sign}[L(d_2)] \times \min\{|L(d_1)|, |L(d_2)|\},$$

где обозначение $\boxplus$ символизирует действие, представленное нижней строкой формулы;

$L(d_1)$ – стратегически независимые информационные биты;

$L(d_2)$ – биты четности от сочетаний $L(d_1)$.

Для осуществления алгоритма Бала требуется определенная подготовка: во время приема кодового вектора, вместе с выработкой МРС также следует провести операцию проверки на четность. В свою очередь, для проверки проверочных символов на четность требуется искусственно ввести надежный проверочный символ h_4 . Код Хэмминга (7, 4, 3) можно представить на рассмотрение в следующем виде: $x_1x_2x_3x_4h_1h_2h_3h_4$, а проверки на четность описываются выражениями: $h_1 + h_2 + h_3 = h_4$, $x_1 + x_2 + x_3 = h_1$, $x_2 + x_3 + x_4 = h_2$, $x_1 + x_2 + x_4 = h_3$, что показано на рисунке 1.

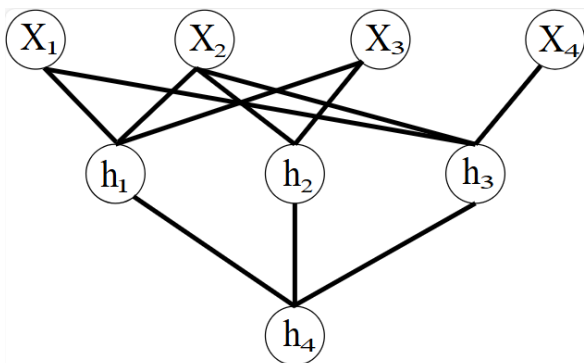


Рисунок 1. Проверка на четность элементов кодового вектора

Декодер со встроенным алгоритмом Бала работает следующим образом. После выработки МРС и первичной проверки на четность информационных символов $x_1x_2x_3x_4$ происходит повышение степени их надежности при выполнении

алгоритма Бала. После первичного применения алгоритма повышения степени надежности символов к информационным разрядам происходит проверка избыточных символов $h_1h_2h_3$ с последующим применением к ним алгоритма Бала в случае, если их МРС принимает значение, отличное от максимального, восстанавливая комбинации $x_1x_2x_3h_1$, $x_2x_3x_4h_2$ или $x_1x_2x_4h_3$ надежность избыточных символов. Любые изменения МРС для каждого символа записывают в общий кодовый вектор V_{MPC} . После того, как вектор выполнит все необходимые проверки и восстанавливающие процедуры, формируется вектор более надежных символов. Это позволяет выбрать подходящую комбинацию перестановок.

Экспериментальным путем было подтверждено, что в некоторых ситуациях выполнение алгоритма Бала до конца не является оптимальным решением. В частности, можно сократить время, затрачиваемое на выполнение алгоритма, если знаки символов, полученных на первом шаге, будут идентичны знакам исходных символов.

Операция повышения степени надежности позволит формировать ККД с минимально возможным количеством перестановок – от одной до $k \times (n - k - 1)$ перестановок, благодаря чему объем занимаемой памяти позволит хранить готовые матрицы (проверочная, транспонированная), соответствующие каждой хранимой перестановке.

Расчет количества арифметических операций, необходимых для выполнения алгоритма повышения надежности

Подготовительными операциями являются операции проверки на четность $h_1 + h_2 + h_3 = h_4$, $x_1 + x_2 + x_3 = h_1$, $x_2 + x_3 + x_4 = h_2$, $x_1 + x_2 + x_4 = h_3$ и для их выполнения потребуется 8 арифметических операций. При возникновении ошибки при проверке на четность начинается выполнение алгоритма Бала. Для расчета повышения степени надежности во внимание принимается значение максимальной оценки МРС $x_1 + x_2 + x_4 = h_3$ и 2 оценки символов – λ_a и λ_b . Формируется список повышающих степень надежности коэффициентов (для λ_a и λ_b соответственно). Значение оценки λ_a следует сложить с повышающим коэффициентом, полученном на предыдущем шаге для λ_b и умножить на -1. При отрицательном значении полученного коэффициента или исходного значения $x_1 + x_2 + x_4 = h_3$ повторить процедуру умножения на -1. Соответствующие действия выполняются для оценки λ_b . Поскольку алгоритм Бала выполняется до тех пор, пока дважды не будут получены идентичные значения восстановления символов, следует вычислить арифметиче-

скую сложность выполнения 1 шага восстановления: от 4 до 6 простых арифметических операций. Поскольку во внимание следует принимать расчеты на худший случай – прием в расчет 6 элементарных арифметических операций.

Количество шагов для достижения конечного повышения может быть не достигнуто в принципе, поэтому на практике следует устанавливать ограничение по количеству выполненных шагов повышения степени надежности. Максимальное количество повторений шага восстановления символов без наличия результата (в случае отсутствия остановки) можно рассчитать по формуле размещения с повторениями для величины множества оценок МРС, в том числе и с отрицательным знаком, что для кода Хэмминга (7, 4, 3) составляет $15^2 = 225$ шагов и 1350 арифметических операций. Данный расчет показывает необходимость введения ограничений на восстанавливающие операции, иначе они теряют актуальность в связи с продолжительностью своего исполнения. На практике большинство восстановлений происходят менее чем за n шагов, в связи с этим может быть введена рекомендация об ограничении количества исполняемых циклов до $2n$ шагов, что потребует выполнения не более 90 простейших арифметических операций.

Таким образом, алгоритм восстановления символов за 90 простейших арифметических операций способен повысить достоверность переданных данных, а значит, разрешить применение большего числа перестановок для выполнения процедуры ПД. Это означает, что объем ККД может быть во многом сокращен и вмещать порядка n перестановок, к выбору которых можно будет прийти с применением алгоритма восстановления символов.

Заключение

В статье была доказана необходимость применения ККД и рассмотрен восстанавливающий переданные символы алгоритм Бала. Применение ККД позволяет существенно сократить количество арифметических операций при вычислении параметров ЭК, а значит упростить данную операцию, обеспечив при этом временной и энергетический выигрыш при реализации перестановочного декодирования. Алгоритм восстановления символов, в свою очередь, позволяет существенно сократить объем ККД и сводить алгоритм ПД к применению ряда хранимых комбинаций. Совмещение рассмотренных средств упрощения алгоритма ПД приведет к уменьшению объема ККД и сокращению времени на исполнение декодирования.

Литература

1. Development of network training complexes using fuzzy models and noise-resistant coding / A.A. Gladkikh [et al.] // International Conference on Aviaemechanical Engineering and Transport (AviaENT 2019). 2019. P. 373–379. DOI: 10.2991/aviaent-19.2019.69
2. Бакурова А.Д. Алгоритм реализации интеллектуального декодера с локализацией перестановки // Современные проблемы проектирования, производства и эксплуатации радиотехнических систем: материалы конференции. Ульяновск: УлГТУ, 2022. С. 89–93.
3. Бакурова А.Д. Декодер с локализацией перестановки // Проблемы техники и технологий телекоммуникаций (ПТиТТ-2023): материалы XXV Международной научно-технической конференции. Казань: КНИТУ-КАИ, 2023. С. 529–530.
4. Бакурова А.Д., Саид Б., Бакуров Д.Д. Процедура поиска множества вырожденных матриц в системе перестановок двоичного блочного кода // Современные технологии обработки сигналов: материалы 2-й Всероссийской конференции. М.: Российское научно-техническое общество радиотехники, электроники и связи им. А.С. Попова, 2019. С. 128–132.
5. Гладких А.А., Климов Р.В., Чилихин Н.Ю. Методы эффективного декодирования избыточных кодов и их современные приложения. Ульяновск: УлГТУ, 2016. 258 с.
6. Комашинский В.И., Комашинский Д.В. Когнитивная метафора в развитии телекоммуникационных и промышленных сетевых инфраструктур, или первые шаги к постинформационной эпохе // Технологии и средства связи. 2015. № 1 (106). С. 62–66.
7. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. 2-е изд., испр. М.: Издательский дом «Вильямс», 2003. 1104 с.
8. Свойства циклических структур в системе перестановочного декодирования избыточных кодов / Н.Ю. Бабанов [и др.] // Автоматизация процессов управления. 2020. № 2 (60). С. 101–108. DOI: 10.35752/1991-2927-2020-2-60-101-108
9. Перестановочное декодирование в системе комбинаций кодовых конструкций при оценке биометрических данных / И.Ю. Давыдов [и др.] // Автоматизация процессов управления. 2019. № 2 (56). С. 85–92.
10. Процедура поиска множества вырожденных

- матриц в системе перестановок двоичного блочного кода / Д.В. Ганин [и др.] // Автоматизация процессов управления. 2019. № 4 (58). С. 82–89. DOI: 10.35752/1991-2927-2019-4-58-82-89
11. Coding methods and permutation decoding in the systems for network processing of data / A.A. Gladkikh [et al.] // International Journal of Control and Automation. 2020. Vol. 13, no. 1. P. 93–110.
 12. Климов Р.В., Гладких А.А. Численное моделирование обобщенной процедуры формирования индексов мягких решений // Инфокоммуникационные технологии. 2013. Т. 11, № 2. С. 22–28.
 13. Конопелько В.К., Липницкий В.А. Теория норм синдромов и перестановочное декодирование помехоустойчивых кодов. 3-е изд. М.: Едиториал УРСС, 2012. 176 с.
 14. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М.: Техносфера, 2005. 320 с.
 15. Математическое моделирование защищенных многоточечных сеансов видеоконференцсвязи многоуровневыми графами / А.А. Гладких [и др.] // Инфокоммуникационные технологии. 2024. Т. 22, № 1 (85). С. 31–43. DOI: 10.18469/ikt.2024.22.1.04

Дата поступления 13.11.2024

Дата принятия 16.12.2024

Бакурова Анастасия Денисовна, старший преподаватель кафедры радиотехники, телекоммуникации и защиты информации Ульяновского государственного технического университета. 432027, Российская Федерация, г. Ульяновск, ул. Северный Венец, 32. Тел. +7 917 603-64-27. E-mail: bakurova.ad@mail.ru

UDC 621.391.037.3

DOI: 10.18469/ikt.2024.22.4.06

CALCULATION OF THE NUMBER OF ARITHMETIC OPERATIONS IN ORDER TO FIND THE INVERSE MATRIX AND INCREASE THE RELIABILITY OF ACCEPTED SYMBOLS IN THE PERMUTATION DECODING ALGORITHM USING THE EXAMPLE OF THE HAMMING CODE (7, 4, 3)

Bakurova A.D.

Ulyanovsk State Technical University, Ulyanovsk, Russian Federation

E-mail: bakurova.ad@mail.ru

Permutation decoding of group systematic noise-resistant codes makes it possible to maximize the use of redundancy inherent in the code, in comparison with other methods of digital data decoding. This solves the complex computational problem of finding the equivalent code needed to find the error vector. The most expensive part of the equivalent code searching algorithm is matrix transformations, in particular, the search for the inverse matrix. The idea of implementing a cognitive decoder card is described in the number of articles, but there are no calculations proving the multiplicity of winnings and showing the quantitative effectiveness of the proposed solution. In this work, we show the gain in the number of operations when using the cognitive card of the decoder. Also, the procedure for increasing the reliability of accepted symbols using the Score algorithm allows to avoid repeated permutation definitions and the calculation of an equivalent code.

Keywords: permutation decoding, inverse matrix, reliability enhancement, cognitive decoder map, equivalent code

Bakurova Anastasiya Denisovna, Ulyanovsk State Technical University, 32, Severny Venets Street, Ulyanovsk, 432027, Russian Federation; Senior Teacher of the Radio Engineering, Telecommunications and Information Security Department. Tel. +7 917 603-64-27. E-mail: bakurova.ad@mail.ru

References

1. Gladkikh A.A. et al. Development of network training complexes using fuzzy models and noise-resistant coding. *International Conference on Aviaemechanical Engineering and Transport (AviaENT 2019)*, 2019, pp. 373–379. DOI: 10.2991/aviaent-19.2019.69

2. Bakurova A.D. An algorithm for implementing an intelligent decoder with permutation localization. *Sovremennye problemy proektirovaniya, proizvodstva i ekspluatatsii radiotekhnicheskikh system: materialy konferencii*. Ulyanovsk: UIGTU, 2022, pp. 89–93. (In Russ.)
3. Bakurova A.D. Decoder with permutation localization. *Problemy tekhniki i tekhnologij telekommunikacij (PTiTT-2023): materialy XXV Mezhdunarodnoj nauchno-tekhnicheskoj konferencii*. Kazan': KNITU-KAI, 2023, pp. 529–530. (In Russ.)
4. Bakurova A.D., Said B., Bakurov D.D. The procedure for searching for a set of degenerate matrices in a system of permutations of a binary block code. *Sovremennye tekhnologii obrabotki signalov: materialy 2 Vserossijskoj konferencii*. Moscow: Rossijskoe nauchno-tekhnicheskoe obshchestvo radiotekhniki, elektroniki i svyazi im. A.S. Popova, 2019, pp. 128–132. (In Russ.)
5. Gladkikh A.A., Klimov R.V., Chilikhin N.Yu. *Methods of Effective Decoding of Redundant Codes and Their Modern Applications*. Ulyanovsk: UIGTU, 2016, 258 p. (In Russ.)
6. Komashinsky V.I., Komashinsky D.V. The cognitive metaphor in the evolution of telecommunication and industrial network infrastructures or the first steps towards the post-information society. *Tekhnologii i sredstva svyazi*, 2015, no. 1 (106), pp. 62–66. (In Russ.)
7. Sklyar B. *Digital Communication. Theoretical Foundations and Practical Application*. 2nd ed. Moscow: Izdatel'skij dom «Vil'yams», 2003, 1104 p. (In Russ.)
8. Babanov N.Yu. et al. Properties of cyclic structures in the system of permutation decoding of redundant codes. *Avtomatizaciya processov upravleniya*, 2020, no. 2 (60), pp. 101–108. DOI: 10.35752/1991-2927-2020-2-60-101-108 (In Russ.)
9. Davydov I.Yu. et al. Permutation decoding in the system of combinations of code designs in the evaluation of biometric data. *Avtomatizaciya processov upravleniya*, 2019, no. 2 (56), pp. 85–92. DOI: 10.35752/1991-2927-2019-4-58-82-89 (In Russ.)
10. Ganin D.V. et al. A procedure for finding a set of degenerate matrices in a binary block code permutation system. *Avtomatizaciya processov upravleniya*, 2019, no. 4 (58), pp. 82–89. (In Russ.)
11. Gladkikh A.A. et al. Coding methods and permutation decoding in the systems for network processing of data. *International Journal of Control and Automation*, 2020, vol. 13, no. 1, pp. 93–110.
12. Klimov R.V., Gladkikh A.A. Numerical simulation of the generalized procedure of formation of indices of soft decisions. *Infokommunikacionnye tekhnologii*, 2013, vol. 11, no. 2, pp. 22–28. (In Russ.)
13. Konopelko V.K., Lipnitsky V.A. *Theory of Syndrome Norms and Permutation Decoding of Noise-Resistant Codes*. 3rd ed. Moscow: Editorial URSS, 2012, 176 p. (In Russ.)
14. Morelos-Zaragoza R. *The Art of Noise-Tolerant Coding. Methods, Algorithms, Application*. Moscow: Technosphaera, 2005, 320 p. (In Russ.)
15. Gladkikh A.A. et al. Mathematical modeling of secure multipoint video conference sessions using multilayer graphs. *Infokommunikacionnye tekhnologii*, 2024, vol. 22, no. 1 (85), pp. 31–43. DOI: 10.18469/ikt.2024.22.1.04 (In Russ.)

Received 13.11.2024

Accepted 16.12.2024