ДЕКОДИРОВАНИЕ МАТРОИДНЫХ КОДОВ

Бодян Г. К., Бодян Д. Г., Дунай Л. Ф.
Технический университет Молдовы ТУМ
Шт. чел Маре, 168, Кишинев – MD2012, Р. Молдова
Тел.: +(37322) 237505; e-mail: gbodean@mail.md

Аннотация – Описана технология декодирования матроидного (М-кода), исследована сложность задачи и процедуры декодирования, представлены схемные решения процесса декодирования М-кодов.

I. Введение

Блок-схема кодера матроидного (М-)кода представлена в работе [1]. Там же, в общих чертах, описан алгоритм декодирования М-кода. Таким образом, решение задачи декодирования М-кода только декларировано, но не рассмотрено детально.

Напомним суть проблемы. Длина n принятого слова **v** равна: n=2t+k, где k – число исходных m–разрядных символов, t - число ошибочных символов. Например, если k=8 и t=8, то n=24. В общем случае, декодер должен найти корректное решение среди $\mathbf{C}_n^k$, т.е. $\mathbf{C}_{24}^8 = 735471$, систем (из k=8) линейных уравнений. На первый взгляд задача представляется сложно разрешимой. (Появляется "соблазн" отдать предпочтение кодам Рида-Соломона).

В данной работе будет представлена общая схема декодирования матроидного кода, а также, детально рассмотрены вопросы и задачи восстановления исходных данных на основе полученных.

II. Основная часть

Матроидные коды относятся к классу линейных кодов. Кодирование и декодирование слов кода выполняется (или должно выполняться) асинхронно (за один такт). Общая схема "матроидного" кодирования и декодирования представлена на рисунке 1.

На вход кодера подается исходный вектор **a**(x), содержащий k символов разрядности m. Кодер выполняет линейное преобразование вида:

$$\mathbf{v}(x) = \mathbf{a}(x) \bullet \mathbf{U}_{k \times n}, \quad (1)$$

где **U** – матрица k×n однородного матроида.

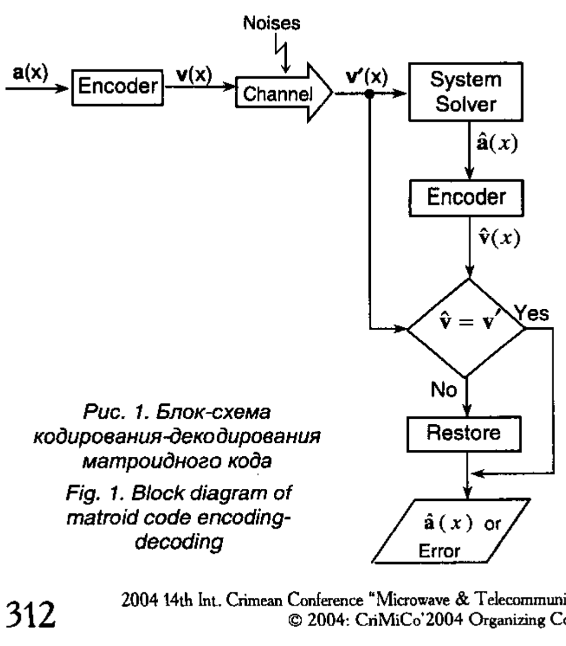

Рис. 1. Блок-схема кодирования-декодирования матроидного кода

Fig. 1. Block diagram of matroid code encoding-decoding

Преобразование (1) легко выполнимо и подробно описано в [1]. На выходе кодера формируется асинхронно вектор **v**(x), который посылается по коммуникационному каналу (линии связи) получателю. Принятый вектор **v**'(x), возможно содержащий ошибки, поступает на вход "решателя" System Solver. Этот решатель представляет собой комбинационную схему, которая содержит блоки умножения (мультипликаторы) и суммирования по модулю p(x), где p(x) – порождающий полином расширенного поля Галуа **GF**^{k}(2^m): $p(x) = 1 + \sum_{i=1}^{m} p_i x^i,\ p_i \in \{0,1\}$. Для k выбранных символов принятого вектора **v**'(x) решается соответствующая система линейных уравнений. В результате получается решение $\hat{\mathbf{a}}(x)$ - вычисленные значения компонент исходного вектора **a**(x).

Полученный таким образом вектор $\hat{\mathbf{a}}(x)$ "пропускается" через кодер, на выходе которого формируется вычисленное значение вектора $\hat{\mathbf{v}}(x)$. Вычисленный и принятый вектора соответственно $\hat{\mathbf{v}}(x)$ и **v**'(x) сравниваются (также асинхронная операция!), на основании чего принимается решение: является ли вычисленный вектор $\hat{\mathbf{a}}(x)$ корректным, или полученная информация ошибочна и следует выполнить полную процедуру восстановления исходных данных?

Процедура полного восстановления – блок Restore на диаграмме рис.1, предусматривает решение $N = \mathbf{C}_n^k$ систем линейных уравнений. Среди N решений могут быть корректные и ошибочные. Блок Restore должен выбрать корректное решение, т.е. вычисленный вектор $\hat{\mathbf{a}}(x)$, и выдать его на выход декодера. С ростом n величина N, с точки зрения аппаратных затрат, становится неприемлемо большой. Например, для k=8 и t=8, имеем $N = \mathbf{C}_{24}^8 = 735471$ решений систем линейных уравнений.

Решение системы линейных уравнений представляет собой значения k-значного вектора $\hat{\mathbf{a}}(x) = \langle a_1, \ldots, a_k \rangle$, где a_i=f(**v**'), i=1,...,k. Аппаратная реализация (одного) линейного преобразования $a_{(\bullet)}$=f(**v**') представляет собой k–входовое комбинационное устройство, выполняющее k умножений по модулю p(x) и суммирование по модулю p(x) результатов произведений. Тогда сложность одного преобразователя составит величину:

$$c(k) = k(\text{мультипликаторов}) + 1(\text{сумматор}), \quad (2)$$

а сложность устройства вычисления компонент k-разрядного вектора $\hat{\mathbf{a}}(x)$ составит величину:

$$\mathbf{C}(k) = k \cdot c(k) = k^2\ (\text{мультипликаторов}) + k(\text{сумматор}).$$

Для анализируемого примера, имеем $\mathbf{C}$(8)=64+8, а сложность всех решателей систем линейных уравнений декодера составит величину:

2004 14th Int. Crimean Conference "Microwave & Telecommunication Technology" (CriMiCo'2004). 13-17 September, Sevastopol, Crimea, Ukraine
© 2004: CriMiCo'2004 Organizing Committee; Weber Co. ISBN: 966-7968-69-3. IEEE Catalog Number: 04EX843

$N \cdot C(k)$ или 735471·73= 52953912. (3)

Это весьма внушительное значение даже для современного уровня развития СВЧ техники. В то же время, сложность процедуры восстановления во многом зависит от способа решения систем линейных уравнений.

Кроме указанной проблемы сложности реализации решателей декодера, следует выделить, также, задачу выбора корректных решений. Принятый вектор **v**'(x) может содержать ошибку кратности *e*, где $e \in \{1,\ldots,t\}$. В этом случае, из всего множества решений *N*, корректными окажутся только

$$N(e) = \mathbf{C}^{k}_{n-e} \qquad (4)$$

систем линейных уравнений. Для *k*=8, *n*=24 и *e*=8 имеем $N(8) = \mathbf{C}^{8}_{16} = 12870$, которую умножаем на количество сумматоров-мультипликаторов:

$$N(8) \cdot C(8) = 12870 \cdot 72 = 926640. \qquad (5)$$

В этом случае величина (5) на несколько порядков меньше величины (3).

Дальнейшее уменьшение сложности решателя декодера можно обеспечить за счет усечения решения системы линейных уравнений, а именно, вычисления значения только для одной компоненты $a_{(\bullet)}$ расчетного вектора $\hat{\mathbf{a}}(x)$. Тогда структура решателя редуцируется до схемы одного линейного преобразователя, сложность которого равна $c(k)$, и аппаратные затраты на решатель составят величину:

$$O(k)=N(k)\cdot c(k) \text{ или } O(8)= 12870\cdot(8+1)= 115830. \qquad (6)$$

Далее, устройство выбора корректного решения сравнивает вычисленные значения всех $O(k)$ компонент $a_{(\bullet)}$ и принимает решение: продолжить анализ или считать полученное значение $a_{(\bullet)}$ истинным.

Важно отметить, что существует оптимальная процедура поиска корректного решения, которая исключает дальнейший анализ для произвольного *e*. Она основана на следующем **утверждении**: для выбора корректного решения *необходимо* и *достаточно* определить $\mathbf{C}^{k}_{n-t}$ идентичных (истинных) решений среди всевозможных $\mathbf{C}^{k}_{n}$ решений.

В докладе приводится процедура (последовательно-)параллельного поиска истинных решений среди $\mathbf{C}^{k}_{n-t}$ возможных. Такая процедура может быть реализована асинхронно. Разработана демонстрационная программа, которая визуализирует процесс матроидного кодирования и декодирования зашумленных графических изображений.

III. Заключение

Таким образом, найдена оптимальная стратегия декодирования матроидного кода, позволяющая реализовать декодер М-кода минимальными аппаратными затратами.

IV. Список литературы

[1] *Бодян Г. К., Бодян Д. Г., Чернелев Д. П.* Кодирование и декодирование матроидного кода, исправляющего пакеты ошибок. Материалы Международной Крымской конференции «СВЧ-техника и телекоммуникационные технологии», Севастополь, сентябрь 8-12, 2003 г.

MATROID CODES DECODING

Bodyan Gh. C., Bodyan D. Gh., Dunai L. T.
Technical University of Moldova
TUM, 168, St. cel Mare, Kishinau – MD2012, R. Moldova
phone: (37322) 237505
e-mail: gbodean@mail.md

Abstract –Algorithm of matroid (M-) code decoding is described. The complexity of decoding procedure is analyzed. An optimal decoding algorithm is proposed.

I. Introduction

In [1] was described the matroid code encoding. In this work the M-code decoding is analyzed. Length *n* of matroid codeword **v**(x) is equal to *n*=2*t*+*k*, where *t* is the number of erroneous symbols, *k* – number of original symbols. For example, if *k*=8 and *t*=8 then *n*=24. To find the original codeword **a**(x) is need to solve up to $\binom{k}{n} = \mathbf{C}^{k}_{n}$ systems of *k* linear equations (for above example, $\mathbf{C}^{8}_{24} = 735471$). In this work the main diagram of matroid code decoding is described and theoretically analyzed. Important results for practice are obtained.

II. Main part

M-codes are linear codes. Matroid code encoding and decoding diagram are shown in Figure 1. Encoder performs matrix multiplication (1), where **U** is the uniform matroid matrix. The System Solver gets the transmitted vector **v**'(x), that can contain errors. System-Solver solves a system of linear equations for *k* selected components of **v**'(x). A solution $\hat{a}(x)$ comes on out as a result. This solution is encoded and vector $\hat{v}(x)$ is compared with received vector **v**'(x). If $\hat{v}(x) = v'(x)$ then $\hat{a}(x)$ is accepted as original sequence **a**(x). Otherwise the Restore-block tries to find the correct solution. For this is need to solve $N = \mathbf{C}^{k}_{n}$ systems of linear equations. Among *N* solutions can be erroneous and correct solutions. As *n* grows *N* becomes unacceptable for engineers. In fact, the complexity of the one block of linear transformation that performs calculus of the one components $a_{(\bullet)}$ of vector **a**(x) is equal (2) and the complexity of the hole solver is equal to $C(k)= k^2$(multipliers)+k(adders). The complexity of all decoder's solvers is given by (3).

From the other hand, if received vector **v**'(x) has an *e*-uple error, then Restore-block must select only $N(e)$ correct solutions (4). In this case, complexity of the value $N(e)\cdot C(e)$ is still big; see, for example, (5). Further decreasing of the solver complexity can be provided by cutting obtained solution, namely, by analyzing only solution for one component $a_{(\bullet)}$ of $\hat{a}(x)$. The complexity of solver, in this case, is given by (6).

Generally, next **affirmation** can be made: is needed and sufficient to find $\mathbf{C}^{k}_{n-t}$ identical solutions to restore original data.

The described M-code decoding procedure was implemented in a demo-software that allows visualization of the encoding and decoding noises pictures.

III. Conclusion

In conclusion, an efficient algorithm of matroid code decoding was proposed. This algorithm allows implementing of the M-code decoder with minimal hardware overhead.