

Эффективные алгоритмы обращения синдромной матрицы при декодировании кодов Боуза – Чоудхури – Хоквингема

Т. З. Ву

Санкт-Петербургский
государственный университет
телекоммуникаций
им. проф. М.А. Бонч-Бруевича

vutheduyet99@gmail.com

Е. И. Глушанков

Санкт-Петербургский
государственный университет
телекоммуникаций
им. проф. М.А. Бонч-Бруевича

glushankov57@gmail.com

К. К. Фам

Санкт-Петербургский
государственный университет
телекоммуникаций
им. проф. М.А. Бонч-Бруевича

fam.kk@sut.ru

Аннотация. Обеспечение высокой помехоустойчивости передачи данных с использованием кодов Боуза–Чоудхури–Хоквингема (БЧХ) неизбежно сопряжено с ростом вычислительной сложности, что становится ограничивающим фактором для быстрого действия современных систем связи. Критическим этапом, вносящим наибольшую временную задержку, является решение ключевого уравнения декодирования методами общей линейной алгебры, которые игнорируют структурные особенности данных. В данной работе предлагается подход к оптимизации этого процесса за счет перехода к специализированным алгоритмам (Левинсона и Барайсса), эффективно использующим свойство трёхлинейности и симметрию матрицы синдромов. Целью исследования является сравнительный анализ данных методов для выявления стратегии, обеспечивающей кардинальное снижение вычислительных затрат без ущерба для надёжности декодирования

Ключевые слова: коды БЧХ; трёхлинейная матрица; быстрое действие; декодирование

I. ВВЕДЕНИЕ

В современных системах цифровой связи и хранения данных обеспечение целостности информации при передаче через каналы с шумом и замираниями является критически важной задачей. Для решения этой проблемы широко применяются методы помехоустойчивого кодирования, среди которых особое место занимают коды Боуза–Чоудхури–Хоквингема (БЧХ). Благодаря высокой корректирующей способности и математической строгости, данные коды стали стандартом в таких технологиях, как DVB-S2 [1], а также в архитектуре запоминающих устройств (NAND Flash, SSD) [2, 3]. Однако классические методы декодирования кодов БЧХ часто сталкиваются с проблемой высокой вычислительной сложности, особенно при увеличении длины кода. В связи с этим, данное исследование посвящено разработке быстрых алгоритмов декодирования. Основное внимание уделяется оптимизации этапа поиска локаторов ошибок за счет использования структурных свойств синдромных матриц (в частности, их трёхлинейной структуры), что открывает возможности для применения эффективных алгоритмов типа Левинсона и Барайсса.

II. АЛГОРИТМ ДЕКОДИРОВАНИЯ КОДОВ БЧХ

Основой процесса декодирования кодов БЧХ является процедура идентификации позиций ошибок в

принятой кодовой последовательности $v(x)$. Данная задача сводится к нахождению корней полинома локаторов ошибок $\Delta(x)$, коэффициенты которого функционально связаны с компонентами синдрома.

На первом этапе на основе принятого вектора вычисляются синдромы S_j , которые несут информацию о профиле ошибок $e(x)$:

$$S_j = v(\alpha^j) = e(\alpha^j) = \sum_{k=1}^v e_k \alpha^{j \cdot i_k}, \quad j = 1, 2, \dots, 2t,$$

где $v \leq t$ — число фактических ошибок, а α — примитивный элемент поля Галуа.

Ключевая взаимосвязь между известными значениями синдромов S_j и искомыми коэффициентами полинома локаторов ошибок $\Delta(x) = 1 + \Delta_1 x + \dots + \Delta_v x^v$ описывается тождествами Ньютона. Для определения коэффициентов $\{\Delta_1, \Delta_2, \dots, \Delta_v\}$ необходимо решить систему линейных алгебраических уравнений (СЛАУ), которая в компактной матричной форме имеет следующий вид [4]:

$$\begin{pmatrix} S_v & S_{v-1} \dots & S_1 \\ S_{v+1} & S_v \dots & S_2 \\ \vdots & \dots \ddots & \vdots \\ S_{2v-1} & S_{2v-2} \dots & S_v \end{pmatrix} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_v \end{pmatrix} = \begin{pmatrix} -S_{v+1} \\ -S_{v+2} \\ \vdots \\ -S_{2v} \end{pmatrix}. \quad (1)$$

III. ОПРЕДЕЛЕНИЯ ПОЛИНОМА ЛОКАТОРОВ ОШИБОК НА ОСНОВЕ LU-РАЗЛОЖЕНИЯ

В качестве базового подхода к решению ключевого уравнения (1) рассматривается метод прямой линейной алгебры, основанный на гауссовском исключении. Для повышения численной устойчивости и алгоритмической гибкости данный метод реализуется через процедуру LU-разложения с частичным выбором ведущего элемента [5].

Суть метода заключается в факторизации исходной матрицы синдромов S в произведение нижнетреугольной матрицы L , верхнетреугольной матрицы U и матрицы перестановок строк P :

$$PS = LU,$$

где матрица P гарантирует, что на диагонали не возникнут нулевые элементы, минимизируя вычислительные погрешности.

После выполнения факторизации задача нахождения вектора коэффициентов $\Delta = [\Delta_\nu, \dots, \Delta_1]^T$ сводится к решению двух треугольных систем уравнений методом прямой и обратной подстановки, вместо трудоемкого прямого вычисления обратной матрицы S^{-1} .

Алгоритмическая реализация данной процедуры представлена в алгоритме 1.

Алгоритм 1. Процедура нахождения коэффициентов $\Delta(x)$ на основе LU-разложения:

Входные данные: Матрица синдромов S размера $\nu \times \nu$ и вектор правой части $B = [-S_{\nu+1}, \dots, -S_{2\nu}]^T$.

Этап 1: Факторизация – выполнить разложение $PS = LU$ с выбором главного элемента по столбцу.

Этап 2: Прямая подстановка – решить систему $Ly_j = P_j$ относительно промежуточного вектора y .

$$y_i^{(j)} = P_i^{(j)} - \sum_{k=1}^{i-1} l_{ik} y_k^{(j)}.$$

Этап 3: Обратная подстановка – решить систему $Ux_j = y_j$.

$$x_i^{(j)} = \frac{1}{u_{ii}} (y_i^{(j)} - \sum_{k=i+1}^n u_{ik} x_k^{(j)}).$$

Этап 4: Определение коэффициентов Δ из $x_i^{(j)}$ – это обратная матрица S^{-1} :

$$\Delta = S^{-1}B.$$

Рассмотренный метод, использующий полное матричное обращение на основе LU-разложения, имеет кубическую вычислительную сложность $O(\nu^3)$. Это является платой за универсальность подхода, применимого к системам общего вида. Однако, данный алгоритм не учитывает специфическую тёплицевую структуру матрицы синдромов, характерную для кодов БЧХ, что оставляет потенциал для дальнейшей оптимизации быстродействия.

IV. БЫСТРОЕ ОБРАЩЕНИЕ СИНДРОМНОЙ МАТРИЦЫ НА ОСНОВЕ РЕКУРРЕНТНОГО АЛГОРИТМА ЛЕВИНСОНА

Учитывая, что матрица синдромов S обладает тёплицевой структурой, для удобства формализации вычислительного алгоритма целесообразно представить её в каноническом виде, обозначенном далее, как A . Такой переход позволяет описать матрицу размером $n \times n$ через ограниченный набор элементов a_k , определяющих её первую строку и первый столбец:

$$A = \begin{pmatrix} a_0 & a_{-1} \dots & a_{-n+1} \\ a_1 & a_0 \dots & a_{-n+2} \\ \dots & \dots & \dots \\ a_{n-1} & a_{n-2} \dots & a_0 \end{pmatrix}.$$

В данном представлении задача нахождения обратной матрицы A^{-1} эффективно решается с помощью рекурсивного алгоритма Левинсона. Суть метода сводится к нахождению первого $x^{(k)}$ и последнего $y^{(k)}$ столбцов обратной матрицы [6–7] для каждого порядка k рекуррентным путем. После нахождения этих граничных векторов для полного размера n , вся обратная матрица A^{-1} восстанавливается по формуле Гохберга–Семенцула [6]:

$$A^{-1} = x_0^{-1} \begin{pmatrix} x_0 & 0 \dots & 0 \\ x_1 & x_0 \dots & 0 \\ \dots & \dots & \dots \\ x_{n-1} & x_{n-2} \dots & x_0 \end{pmatrix} \begin{pmatrix} y_{n-1} & y_{n-2} \dots & y_0 \\ 0 & y_{n-1} \dots & y_1 \\ \dots & \dots & \dots \\ 0 & 0 \dots & y_{n-1} \end{pmatrix}. \quad (2)$$

$$- \begin{pmatrix} 0 & \dots & 0 \\ y_0 & 0 & 0 \\ \dots & \dots & 0 \\ y_{n-2} & y_{n-3} \dots & 0 \end{pmatrix} \begin{pmatrix} 0 & x_{n-1} \dots & x_1 \\ \dots & \dots & \dots \\ 0 & \dots & 0 \\ 0 & \dots & 0 \end{pmatrix}.$$

Данная процедура формализована в алгоритме 2.

Алгоритм 2. Быстрое обращение тёплицевой матрицы (метод Левинсона):

Этап 1: Инициализация ($k = 0$) – задаются начальные значения векторов на основе первого элемента матрицы:

$$x_0^{(0)} = y_0^{(0)} = 1/a_0.$$

Этап 2: Рекурсивный процесс (для $k = 1$ до $n-1$):

На каждом шаге обновляются векторы $x^{(k)}$ и $y^{(k)}$ через значения предыдущего шага:

Шаг 2.1: Вычисление невязок F_k и G_k :

$$F_k = \sum_{i=1}^k a_i x_{k-i}^{(k-1)}, \quad G_k = \sum_{i=1}^k a_{-i} y_{i-1}^{(k-1)}.$$

Шаг 2.2: Вычисление вспомогательных скалярных коэффициентов:

$$r_k = 1 / (1 - F_k G_k), \quad s_k = \frac{\beta}{(1 - F_k G_k)} = -r_k F_k,$$

$$t_k = \frac{\gamma}{(1 - F_k G_k)} = -r_k G_k.$$

(Здесь используются обозначения α, β, γ из промежуточных выкладок, сведенные к коэффициентам обновления).

Шаг 2.3: Обновление векторов – векторы пересчитываются через линейную комбинацию, как показано в системе обновлений:

$$\begin{pmatrix} x_0^{(k)} \\ x_1^{(k)} \\ \dots \\ x_k^{(k)} \end{pmatrix} = \begin{pmatrix} x_0^{(k-1)} \\ x_1^{(k-1)} \\ \dots \\ 0 \end{pmatrix} r_k + \begin{pmatrix} 0 \\ \dots \\ y_{k-2}^{(k-1)} \\ y_{k-1}^{(k-1)} \end{pmatrix} s_k,$$

$$\begin{pmatrix} y_0^{(k)} \\ y_1^{(k)} \\ \dots \\ y_k^{(k)} \end{pmatrix} = \begin{pmatrix} x_0^{(k-1)} \\ x_1^{(k-1)} \\ \dots \\ 0 \end{pmatrix} t_k + \begin{pmatrix} 0 \\ \dots \\ y_{k-2}^{(k-1)} \\ y_{k-1}^{(k-1)} \end{pmatrix} r_k.$$

Этап 3: Восстановление полной матрицы A^{-1} из (2) – после завершения итераций ($k = n - 1$) получены векторы $x^{(n-1)}$ и $y^{(n-1)}$. Элементы обратной матрицы $a_{i,j}$ вычисляются рекуррентно, используя упрощенную:

$$a_{i,j} = a_{i-1,j-1} + \alpha_i \beta_j$$

где α и β – компоненты, полученные из разложения произведения нижне- и верхнетреугольных матриц $L(u)U(v)$. Данный шаг реализуется двойным циклом по i и j , заполняя матрицу на основе её граничных значений.

Алгоритм Левинсона позволяет найти векторы-столбцы за $O(v^2)$ операций. Этап восстановления полной матрицы также требует $O(v^2)$. Таким образом, суммарная вычислительная сложность составляет $O(v^2)$, что значительно эффективнее метода LU-разложения $O(v^3)$, особенно при больших значениях длины кода и корректирующей способности.

V. БЫСТРОЕ ОБРАЩЕНИЕ СИНДРОМНОЙ МАТРИЦЫ НА ОСНОВЕ АЛГОРИТМА БАРАЙССА

Альтернативой методу Левинсона является алгоритм Барайсса [8], который представляет собой специализированную версию LU-разложения, адаптированную для трёхдиагональных матриц. Главное преимущество данного метода заключается в том, что в процессе исключения элементов сохраняется трёхдиагональная структура промежуточных матриц, что позволяет оперировать не всей матрицей, а лишь её порождающими векторами. Процесс декомпозиции матрицы A сводится к итеративному построению последовательности матриц $T^{(-i)}$ (с i нулевыми диагоналями под главной) и $T^{(i)}$ (с i нулевыми диагоналями над главной) с использованием операторов сдвига Z_k [8–10]. Формализованная процедура, включающая этап восстановления обратной матрицы по методу Тренча-Зохара, представлена в алгоритме 3.

Алгоритм 3. Быстрое обращение матрицы (метод Барайсса + Тренча-Зохара):

Этап 1: Инициализация ($i = 0$) – исходные матрицы инициализируются как $T^{(0)} = A$ и $M^{(0)} = I$.

Этап 2: Цикл редукции (для i от 1 до $n - 1$) – на каждом шаге выполняется обновление порождающих векторов через операции сдвига Z и вычитания:

Шаг 2.1: Вычисление множителей (коэффициентов исключения) на основе ведущих элементов предыдущего шага:

$$m_{-i} = t_{-i}^{(-i+1)} / t_0^{(i-1)}; \quad m_i = t_{-i}^{(i-1)} / t_0^{(-i+1)}.$$

Шаг 2.2: Обновление матриц T (формирование нулей):

$$T^{(-i)} = T^{(-i+1)} - m_{-i} Z_{-i} T^{(i-1)};$$

$$T^{(i)} = T^{(i-1)} - m_i Z_i T^{(-i+1)}.$$

(Здесь $Z_{\pm i}$ – операторы сдвига строк вниз/вверх, заполняющие освободившиеся места нулями)

Шаг 2.3: Обновление матриц преобразования M :

$$M^{(-i)} = M^{(-i+1)} - m_{-i} Z_{-i} M^{(i-1)};$$

$$M^{(i)} = M^{(i-1)} - m_i Z_i M^{(-i+1)}.$$

Этап 3: Восстановление обратной матрицы (Алгоритм Тренча-Зохара) [8] – по завершении цикла, вместо трудоёмкого перемножения факторов $U^{-1}L^{-1}$, используется свойство персимметричности обратной трёхдиагональной матрицы.

Шаг 3.1: Из матриц $M^{(-)}$ и $M^{(+)}$ извлекаются параметры: α (первый элемент), вектор-столбец d и вектор-строка c .

Шаг 3.2: Элементы обратной матрицы $b_{i,j} = (A^{-1})_{i,j}$ вычисляются рекуррентно (начиная с последнего элемента к первому):

$$b_{i,j} = b_{i-1,j-1} + \frac{1}{\alpha} (\underline{dc}^T - \underline{c}^R \underline{d}^{RT})_{i-1,j-1},$$

$$b_{n+1-j,n+1-i} = b_{i,j} \text{ (свойство персимметрии)}.$$

Благодаря использованию порождающих векторов длиной $2n - 1$ вместо полных матриц, алгоритм Барайсса снижает сложность декомпозиции. В сочетании с процедурой реконструкции Тренча-Зохара, общая вычислительная сложность нахождения обратной матрицы составляет $O(v^2)$, что сопоставимо с алгоритмом Левинсона и значительно превосходит классические методы $O(v^3)$.

VI. ЭКСПЕРИМЕНТАЛЬНОЕ МОДЕЛИРОВАНИЕ И АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ

Для верификации теоретических оценок сложности и подтверждения корректности работы исследуемых алгоритмов была реализована программная модель системы передачи данных в среде MATLAB. В качестве объекта исследования выбран код БЧХ (63, 18) с

корректирующей способностью $t_{\max} = 10$, функционирующий в канале с аддитивным белым гауссовским шумом (АБГШ) при использовании модуляции QPSK (абб. от англ. Quadrature Phase Shift Keying). Результаты экспериментов представлены на рис. 1.

Анализ графика количества операций (рис. 1б) демонстрирует фундаментальные различия в алгоритмической сложности. Метод, основанный на классическом LU-разложении, характеризуется кубическим ростом затрат ($O(t^3)$), что приводит к резкому увеличению числа операций – до 2800 при $t = 10$. В противовес этому, алгоритмы, учитывающие тёплицеву структуру матрицы (Левинсона и Барайсса), показывают значительно более пологий, квадратичный рост ($O(t^2)$). При этом алгоритм Барайсса в комбинации с методом Тренча-Зохара демонстрирует наименьшее количество вычислений во всем диапазоне значений, незначительно опережая метод Левинсона.

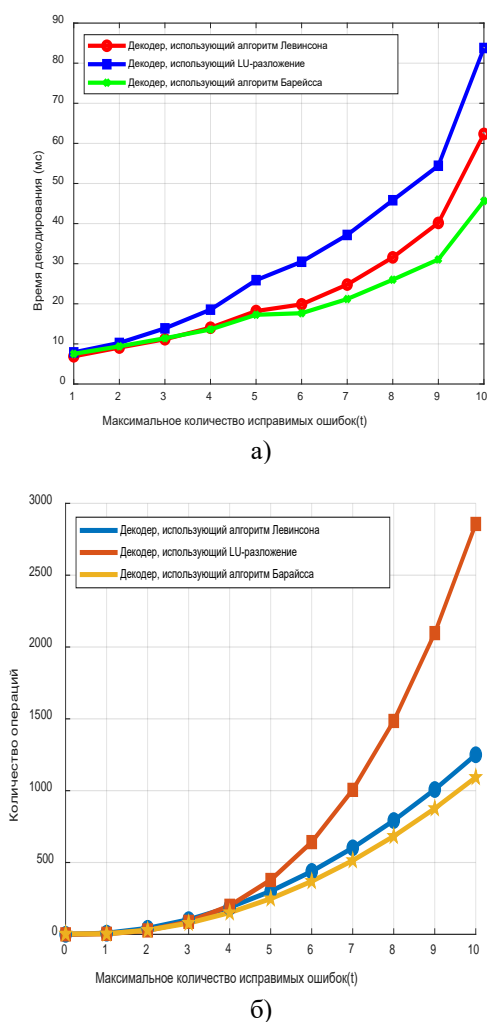


Рис. 1. Сравнительный анализ вычислительной эффективности алгоритмов декодирования кода БЧХ (63, 18): а) зависимость времени декодирования от кратности ошибки; б) зависимость количества арифметических операций от кратности ошибки.

Данная теоретическая сложность имеет прямую корреляцию с реальным временем декодирования, показанным на рис. 1а. При малой кратности ошибок

$t = 3$ производительность методов сопоставима из-за накладных расходов на инициализацию рекурсивных процедур. Однако с ростом t происходит существенная дивергенция кривых. При максимальной нагрузке $t = 10$ декодер на базе LU-разложения требует более 80 мс, в то время как применение алгоритма Барайсса позволяет сократить это время практически вдвое – до уровня 45 мс. Полученные данные позволяют сделать вывод, что для реализации высокоскоростных декодеров наиболее эффективным является подход, основанный на алгоритме Барайсса, обеспечивающий оптимальный баланс между вычислительной сложностью и быстродействием.

VII. ЗАКЛЮЧЕНИЕ

В работе проведено сравнительное исследование вычислительной эффективности алгоритмов решения ключевого уравнения декодирования кодов БЧХ. Экспериментально подтверждено, что учет тёплицевой структуры матрицы синдромов является критическим фактором оптимизации: переход от универсального LU-разложения к специализированным быстрым алгоритмам обеспечивает многократный выигрыш в скорости обработки данных.

На основе полученных результатов установлено, что наилучшим балансом производительности обладает алгоритм Барайсса в комбинации с процедурой восстановления Тренча-Зохара. Данный метод продемонстрировал минимальные вычислительные затраты и наименьшее время задержки декодирования среди рассмотренных подходов, что делает его предпочтительным выбором для аппаратной реализации в высокоскоростных системах связи реального времени.

СПИСОК ЛИТЕРАТУРЫ

- [1] Chen Z., Zhang Y., Ying Y., Wu C., Zeng. X. An area-efficient and degree-computationless BCH decoder for DVB-S2 // 2009 IEEE 8th International Conference on ASIC. DOI: 10.1109/ASICON.2009.5351625.
- [2] Sun F., Rose K., Zhang T. On the Use of Strong BCH Codes for Improving Multilevel NAND Flash Memory Storage Capacity // IEEE Workshop on Signal Processing Systems (SiPS): Design and Implementation, 2006. Vol. 5.
- [3] Micheloni R., Marelli A., Ravasio R. Error correction codes for non-volatile memories. Springer Science & Business Media, 2008.
- [4] Кудряшов Б.Д. Основы теории кодирования: учеб. пособие. СПб.: БХВ-Петербург, 2015. 400 с.
- [5] Даутов. Р.З., Тимербаев. М.Р. Численные методы. Решение задач линейной алгебры и дифференциальных уравнений: уч. пособие. Казань: КФУ, 2021. 168 с.
- [6] Воеводин В.В., Тьртышников Е.Е. Вычислительные процессы с теплицевыми матрицами. М.: Наука. Гл ред физ-мат.лит, 1987. 320 с.
- [7] Ву Т.З., Глушанков Е.И. Исследование вычислительной эффективности декодирования кодов Боуза-Чоудхури-Хоквингема с использованием алгоритма Питерсона-Горенштейна-Цирлера // Радиотехнические и телекоммуникационные системы. 2025. Вып. 4. С. 14-25.
- [8] Bareiss E.H. Numerical Solution of Linear Equations with Toeplitz and Vector Toeplitz matrices // Numerische Mathematik. 1969. Vol. 13, Pp. 404-429.
- [9] Sweet D. R. Numerical methods for Toeplitz matrices: A thesis Doctor of Philosophy // Adelaide, 1982274 p.
- [10] Ву Т.З., Глушанков Е.И. Быстрые алгоритмы для определения полинома локаторов ошибок при декодировании кодов Боуза – Чоудхури – Хоквингема // Вопросы радиоэлектроники. Серия Техника телевидения. 2025. Вып. 4. С. 100-109.