МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Курс: Прикладная алгебра, осень 2018

Практическое задание. Конечные поля и коды БЧХ

Работу выполнил:
Бобко Никита,
студент 323 группы

Москва, 2018

Содержание

1 Формулировка задания

В задании выдаётся список всех примитивных многочленов степени q над полем $\mathbb{F}_2$ для всех q = 2, ... , 16. В этом списке каждый многочлен представлен десятичным числом, двоичная запись которого соответствует коэффициентам полинома над $\mathbb{F}_2$, начиная со старшей степени.

Требуется:
1. Реализовать основные операции в поле $\mathbb{F}_2^q$: сложение, умножение, деление, решение СЛАУ, поиск минимального многочлена из $\mathbb{F}_2[x]$ для заданного набора корней из поля $\mathbb{F}_2^q$;

2. Реализовать основные операции для работы с многочленами из $\mathbb{F}_2^q[x]$: произведение многочленов, деление многочленов с остатком, расширенный алгоритм Евклида для пары многочленов, вычисление значения многочлена для набора элементов из $\mathbb{F}_2^q$;

3. Реализовать процедуру систематического кодирования для циклического кода, заданного своим порождающим многочленом;

4. Реализовать процедуру построения порождающего многочлена для БЧХ-кода при заданных n и t;

5. Построить графики зависимости скорости БЧХ-кода $r = \frac{k}{n}$ от количества исправляемых кодом ошибок t для различных значений n. Какие значения t следует выбирать на практике для заданного n?

6. Реализовать процедуру вычисления истинного минимального расстояния циклического кода d, заданного своим порождающим многочленом, путем полного перебора по всем $2^k - 1$ кодовым словам. Привести пример БЧХ-кода, для которого истинное минимальное расстояние больше, чем величина $2t + 1$;

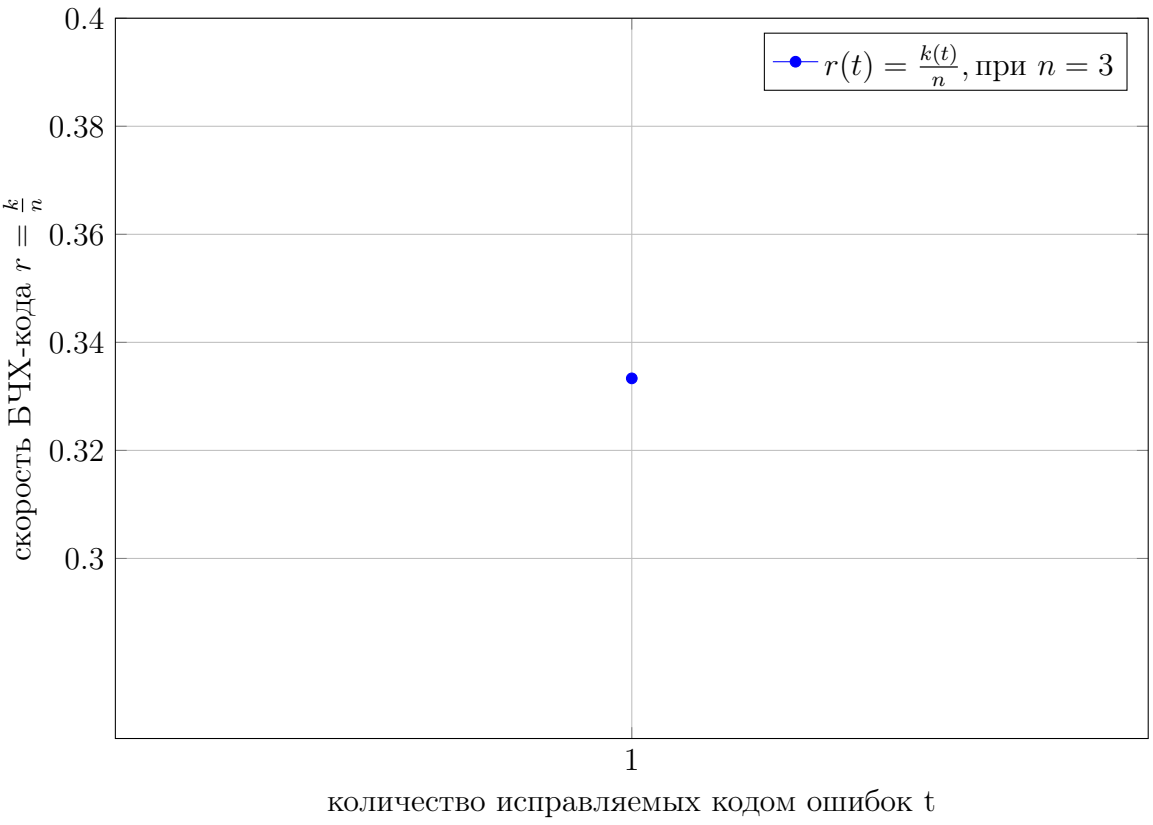
7. Реализовать процедуру декодирования БЧХ-кода с помощью метода PGZ и на основе расширенного алгоритма Евклида. Провести сравнение двух методов декодирования по времени работы;

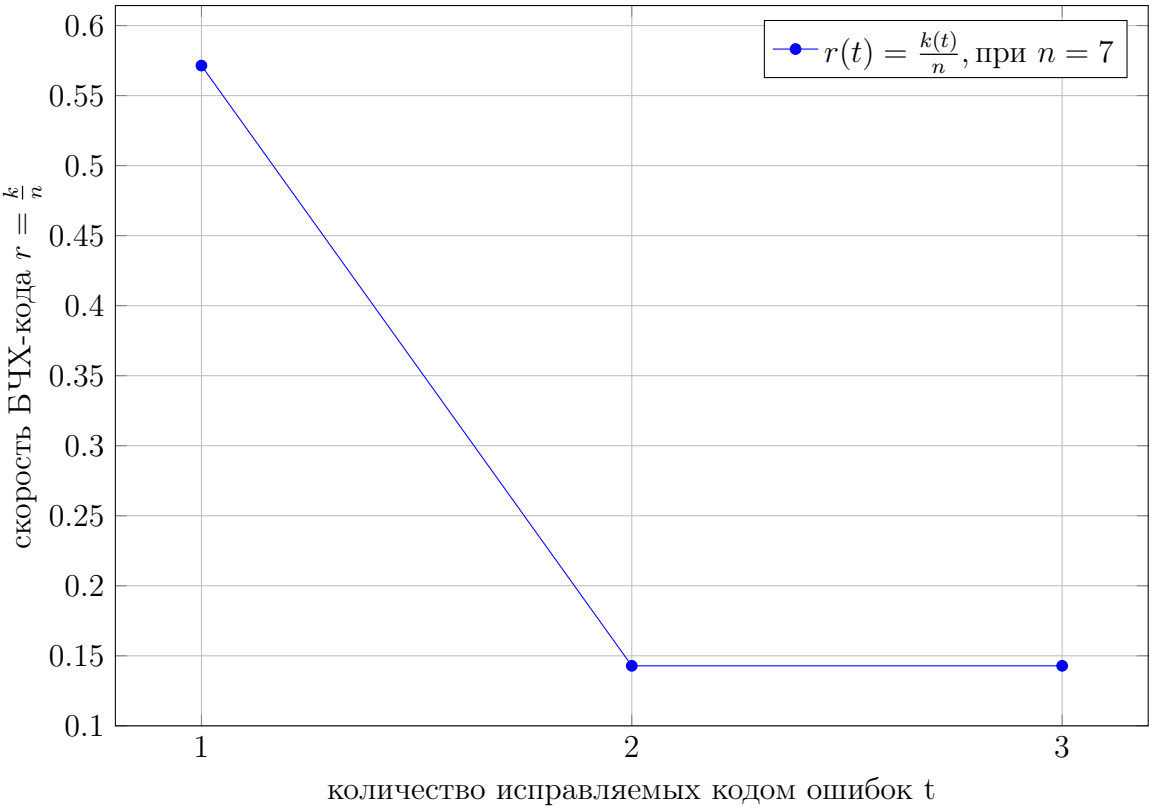
8. С помощью метода стат. испытаний реализовать процедуру оценки доли правильно раскодированных сообщений, доли ошибочно раскодированных сообщений и доли отказов от декодирования для БЧХ-кода. С помощью этой процедуры убедиться в том, что БЧХ-код действительно позволяет гарантированно исправить до t ошибок. Может ли БЧХ-код исправить больше, чем t ошибок? Как ведут себя характеристики кода при числе ошибок, превышающем t?

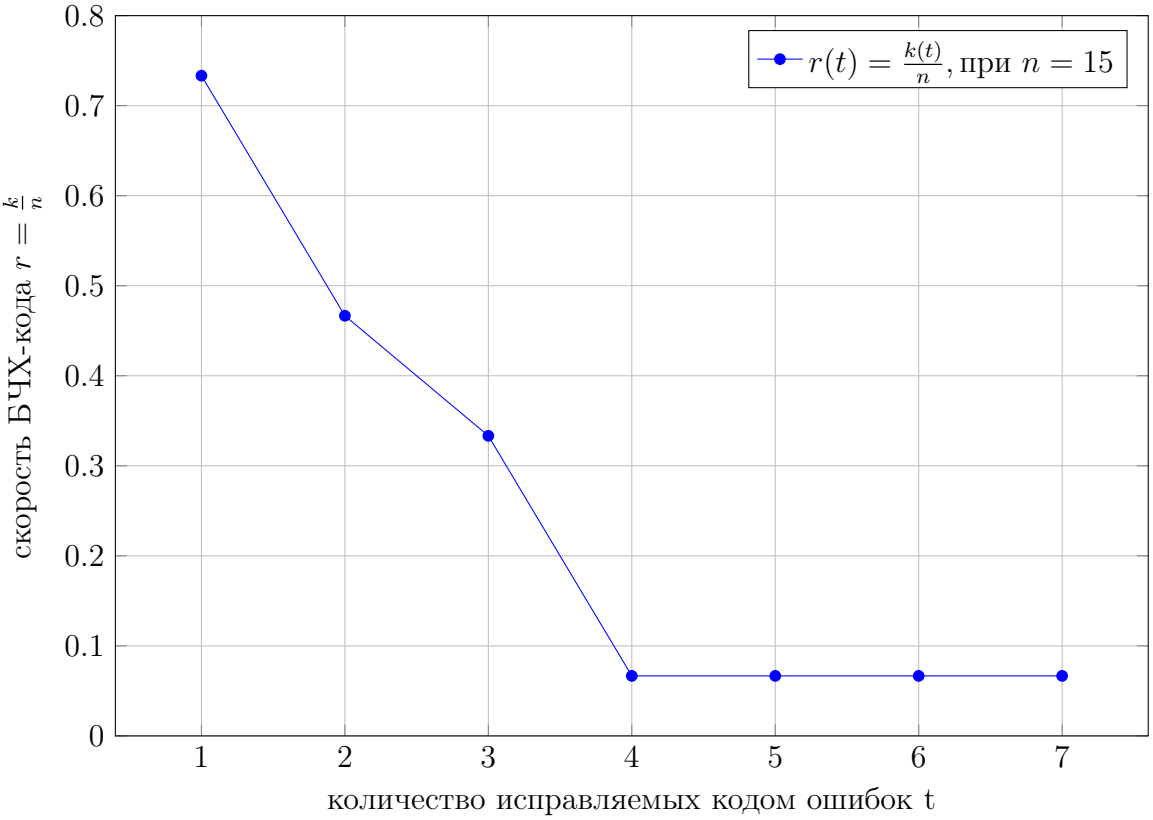
2 Результаты выполнения

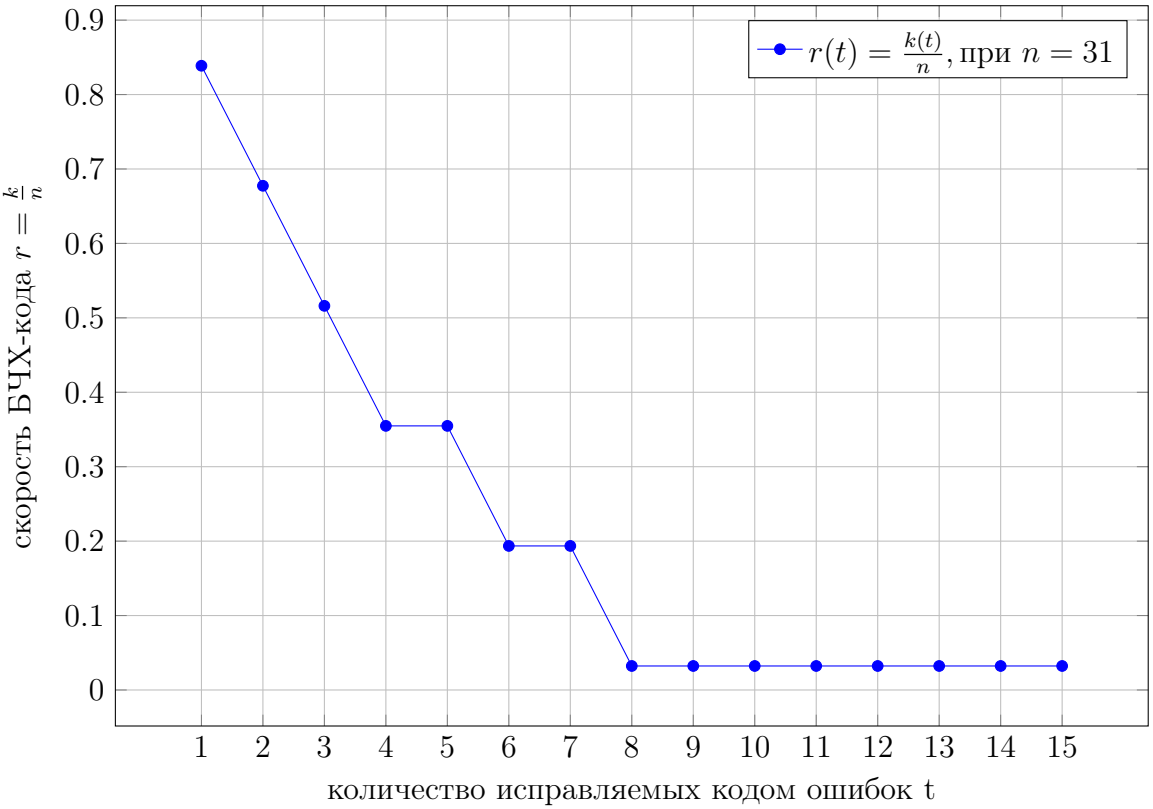
2.1 Скорость БЧХ кода и выбор оптимального кода для заданного n

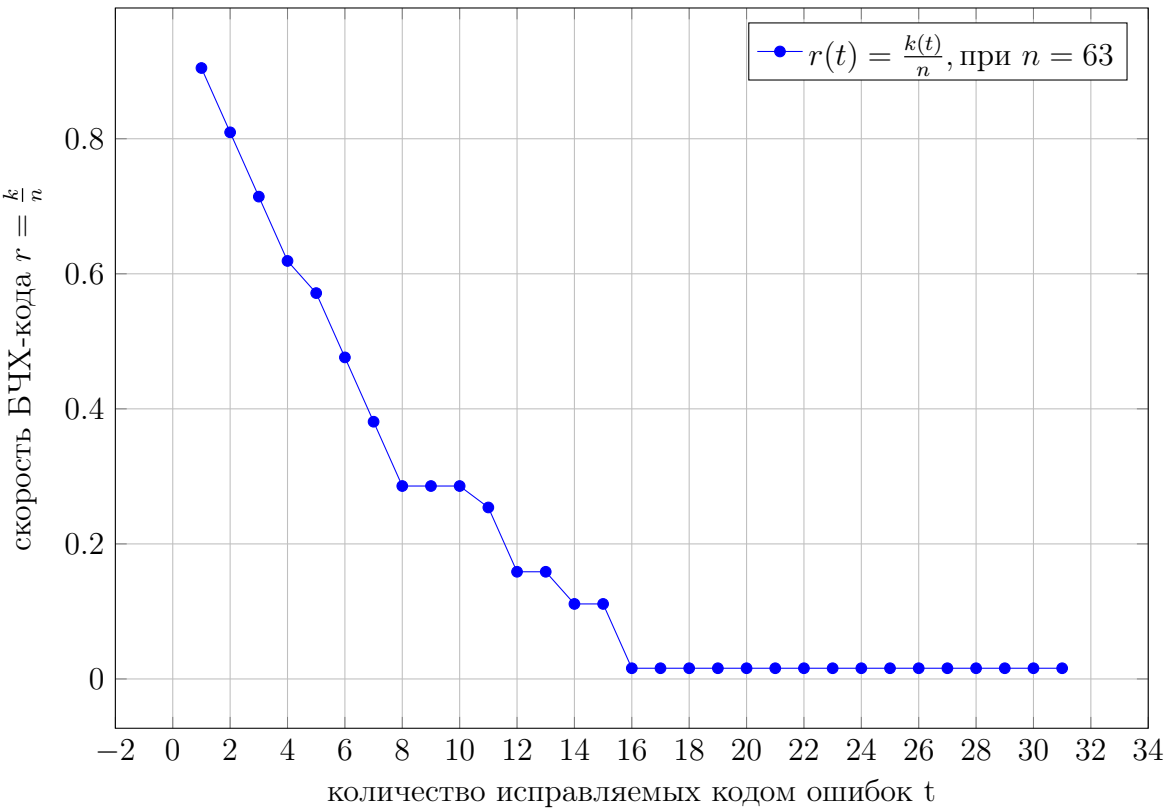
Ниже приведены графики зависимости скорости БЧХ-кода $r = \frac{k}{n}$ от количества исправляемых кодом ошибок t для различных значений n, где k - кол-во "полезных" бит в кодовом слове, несущих само сообщение, а n - длина кодового слова.

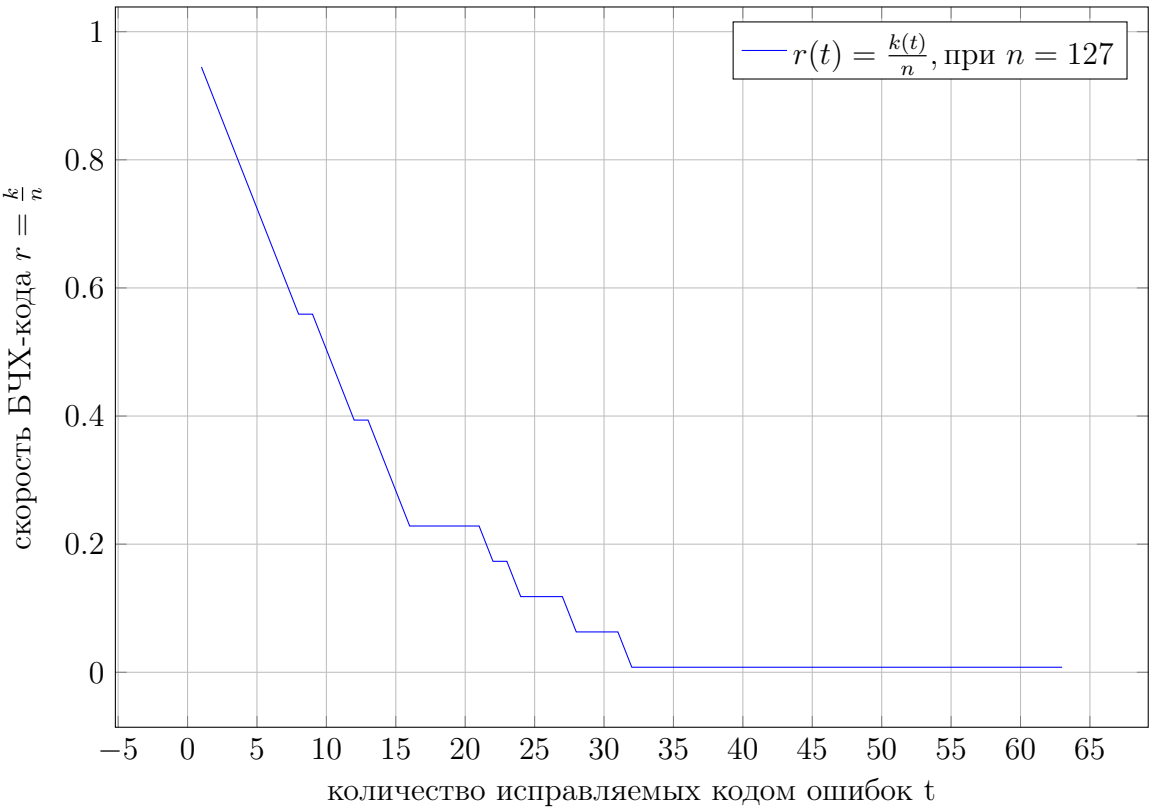


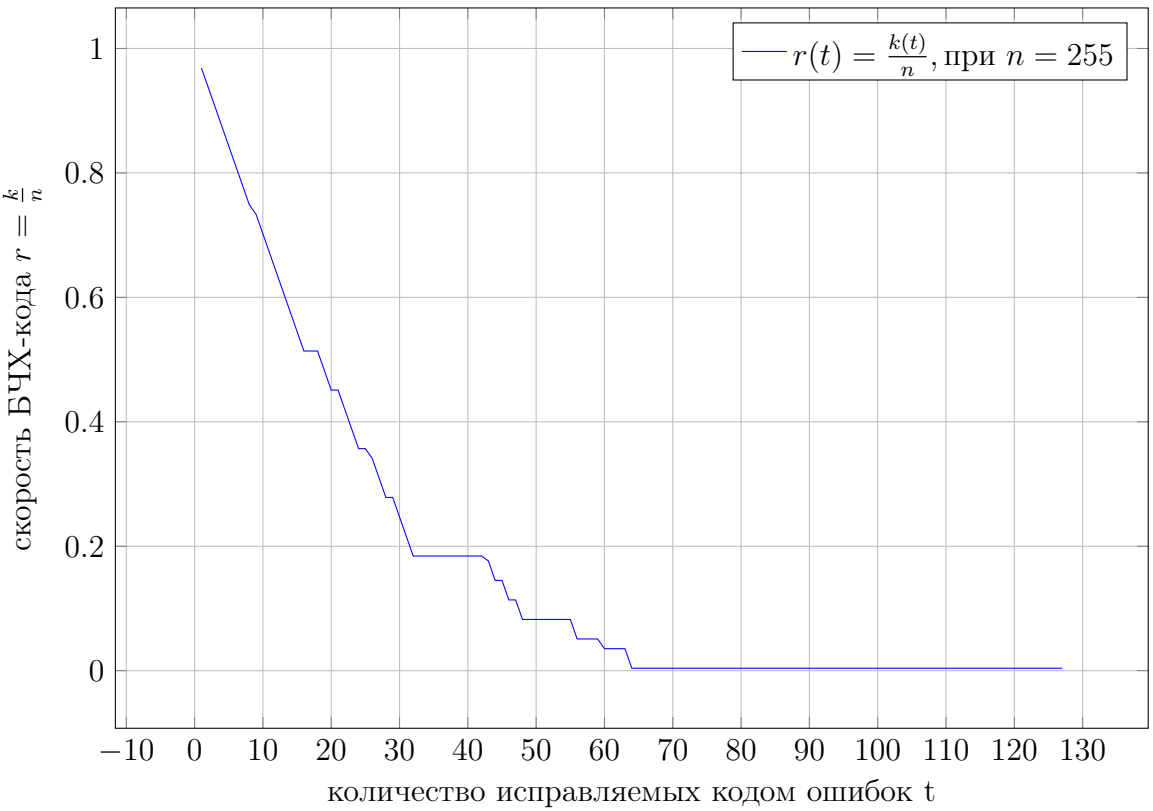

$r(t) = \frac{k(t)}{n}$, при $n = 7$
скорость БЧХ-кода $r = \frac{k}{n}$
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
1
2
3
количество исправляемых кодом ошибок t


$r(t) = \frac{k(t)}{n}$, при $n = 15$
скорость БЧХ-кода $r = \frac{k}{n}$
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
количество исправляемых кодом ошибок t


$r(t) = \frac{k(t)}{n}$, при $n = 31$
скорость БЧХ-кода $r = \frac{k}{n}$
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
количество исправляемых кодом ошибок t


$r(t) = \frac{k(t)}{n}$, при $n = 63$
скорость БЧХ-кода $r = \frac{k}{n}$
0.8
0.6
0.4
0.2
0
−2 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34
количество исправляемых кодом ошибок t


$r(t) = \frac{k(t)}{n}$, при $n = 127$
скорость БЧХ-кода $r = \frac{k}{n}$
количество исправляемых кодом ошибок t
1
0.8
0.6
0.4
0.2
0
−5 0 5 10 15 20 25 30 35 40 45 50 55 60 65


$r(t) = \frac{k(t)}{n}$, при $n = 255$
скорость БЧХ-кода $r = \frac{k}{n}$
количество исправляемых кодом ошибок t
1
0.8
0.6
0.4
0.2
0
−10 0 10 20 30 40 50 60 70 80 90 100 110 120 130

Отметим, что скорость БЧХ-кода это отношение кол-ва полезных бит в кодовом слове k ко всей длине кодового слова n. И по смысловой нагрузке это означает какую долю полезной информации несет каждое кодовое слово. Понятно, что чем выше скорость БЧХ-кода, тем более полезным код является. Именно поэтому графики зависимости $r(t) = \frac{k(t)}{n}$ получились не возрастающими. Так как требуя от кода большего числа исправляемых ошибок мы уменьшаем в коде долю "полезных" бит и увеличиваем долю служебных.

Попробуем ответить на вопрос о том, какое значение t надо выбирать для заданного n. Введем следующие величины: s - пропускная способность канала (биты в секунду). A - кол-во бит, которое надо передать.

Для простоты будем считать, что если произошло больше чем t ошибок, то мы получаем отказ от декодирования и запрашиваем это же сообщение заново.

Разделим A на порции по k бит:

$$\frac{A}{k}$$

Каждая такая порция после кодирования будет содержать n бит, т.е. суммарное кол-во бит, которое нужно передать:

$$\frac{A}{k}n = \left\{ r = \frac{k}{n} \right\} = \frac{A}{r}$$

Время, которое мы потратим на передачу всей этой информации, если считать, что ошибок не произошло (обозначим это время за T):

$$T \stackrel{\text{def}}{=} \frac{A}{r} \cdot \frac{1}{s} = \frac{A}{rs}$$

Далее рассмотрим случайную величину ξ равную номеру первого успеха в серии испытаний Бернули (известно, что такая случайная величина имеет геометрическое распределение). Под испытанием будем понимать попытку передачи одного кодового слова длины n, под успехом испытания будем понимать факт передачи сообщения, с произошедшим количеством ошибок $\in [0, t]$. Если ошибок произошло больше чем t, то мы условились, что мы получаем отказ от декодирования и принимающая сторона запрашивает кодовое слово еще раз, это будем считать за неудачу.

Понятно, что вероятность успеха в каждом испытании равна $\frac{t}{n}$. Т.е. случайная величина η_i равная 0 или 1 (неудаче или успеху в соответствующем испытании):

$$\eta_i = \begin{cases} 1, & p = \frac{t}{n} \\ 0, & q = 1 - p \end{cases}$$

И мат. ожидание ξ, как случайной величины имеющей геометрическое распределение:

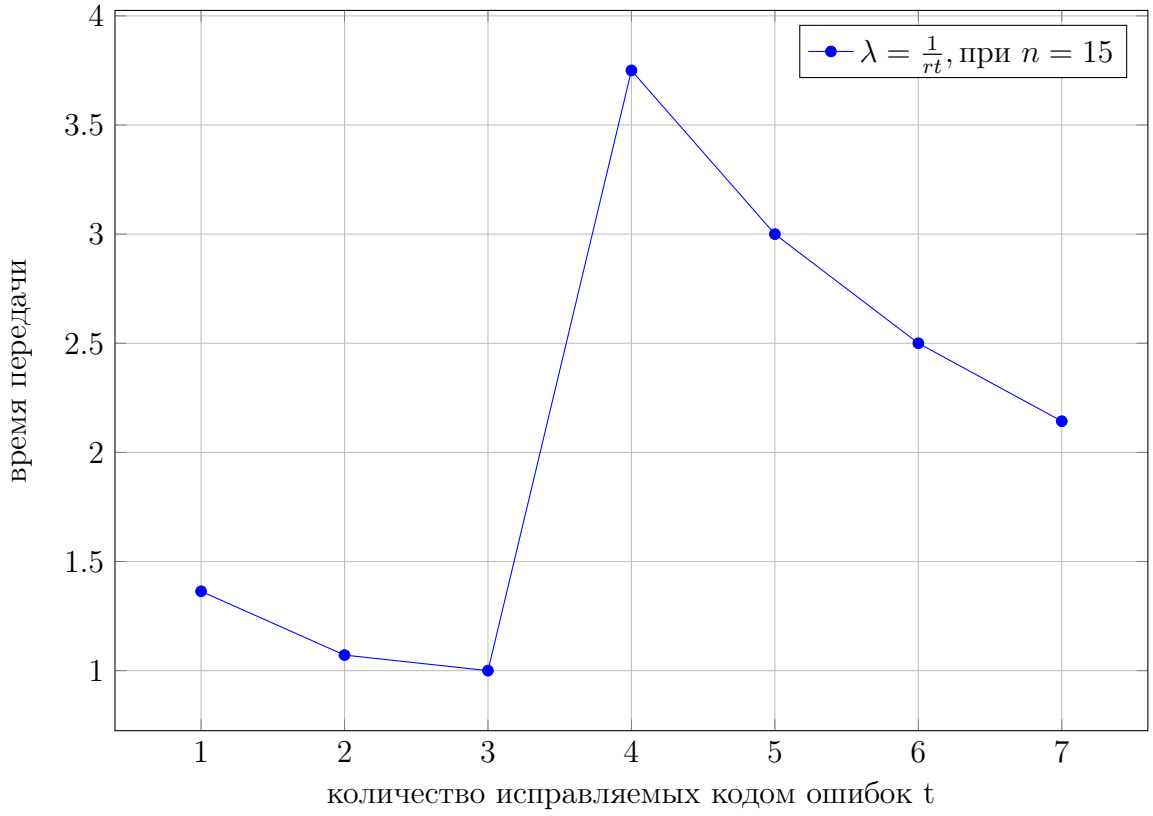
$$\mathbb{E}(\xi) = \frac{1}{p} = \frac{n}{t}$$

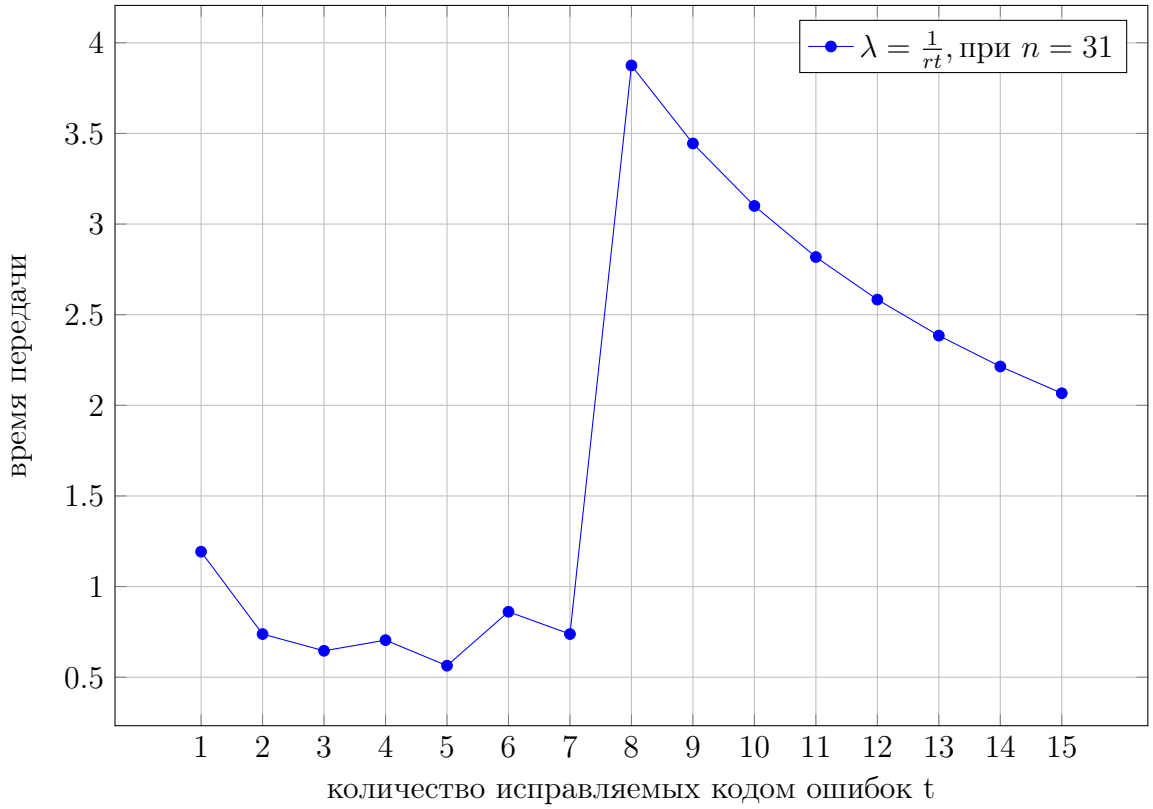
Таким образом кодовое слово будет отправлено лишь с попытки равной ξ. И если мы умножим T на мат. ожидание величины ξ, то мы получим ожидаемое время передачи с учетом произошедших неуспехов при передаче:

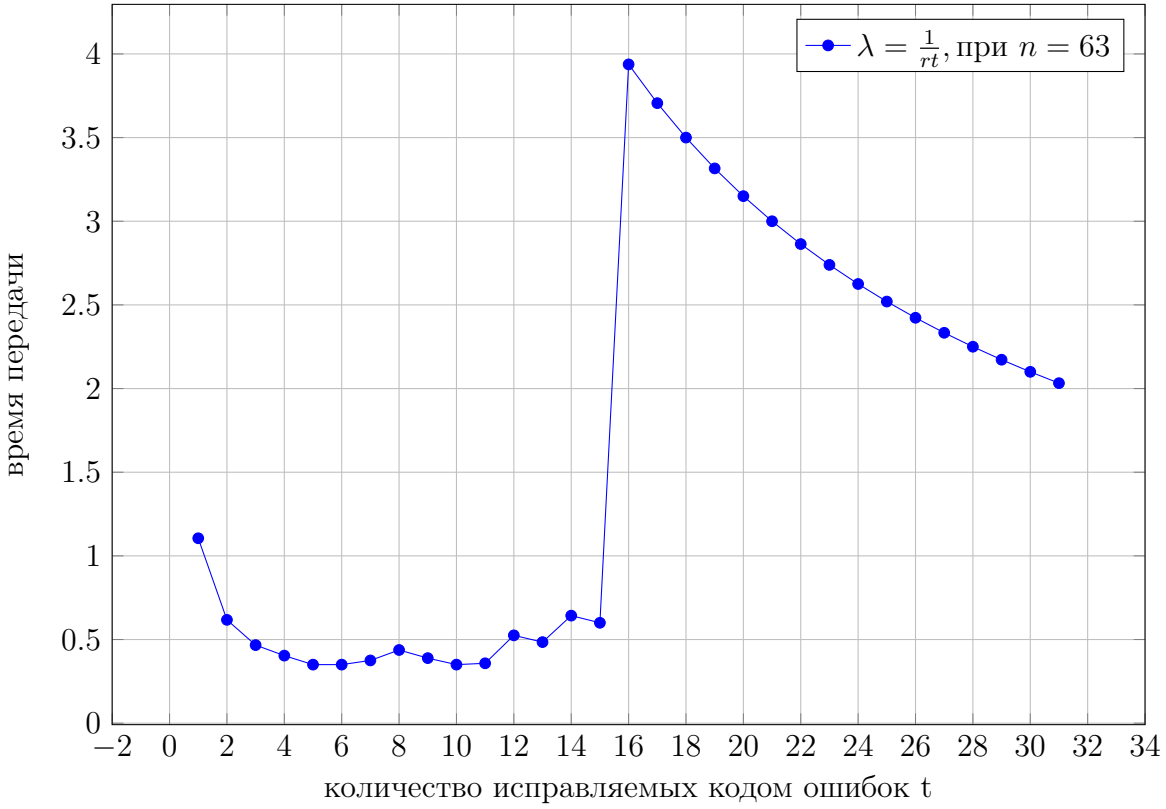
$$T \cdot \mathbb{E}(\xi) = \frac{An}{rst} \sim \frac{1}{rt}, \lambda(t) \stackrel{\text{def}}{=} \frac{1}{rt}$$

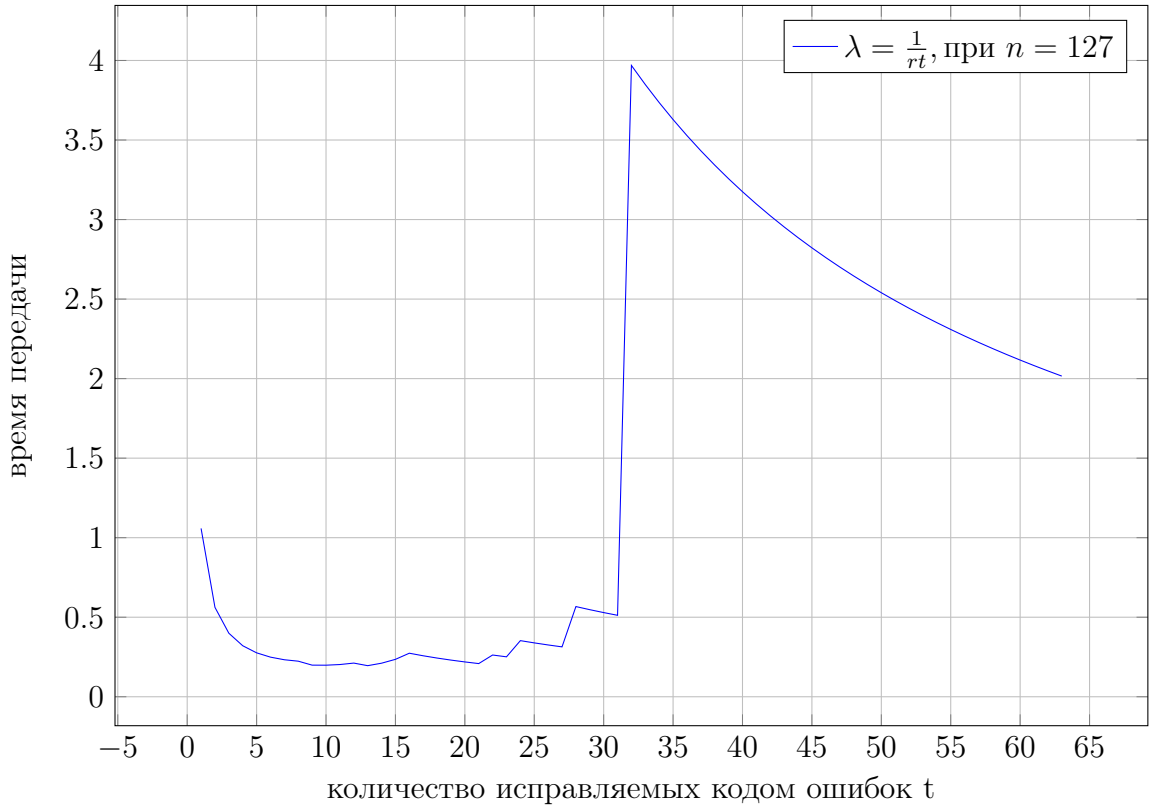
Подытоживая, получаем, что t надо выбирать таким образом, чтобы минимизировать функцию λ.

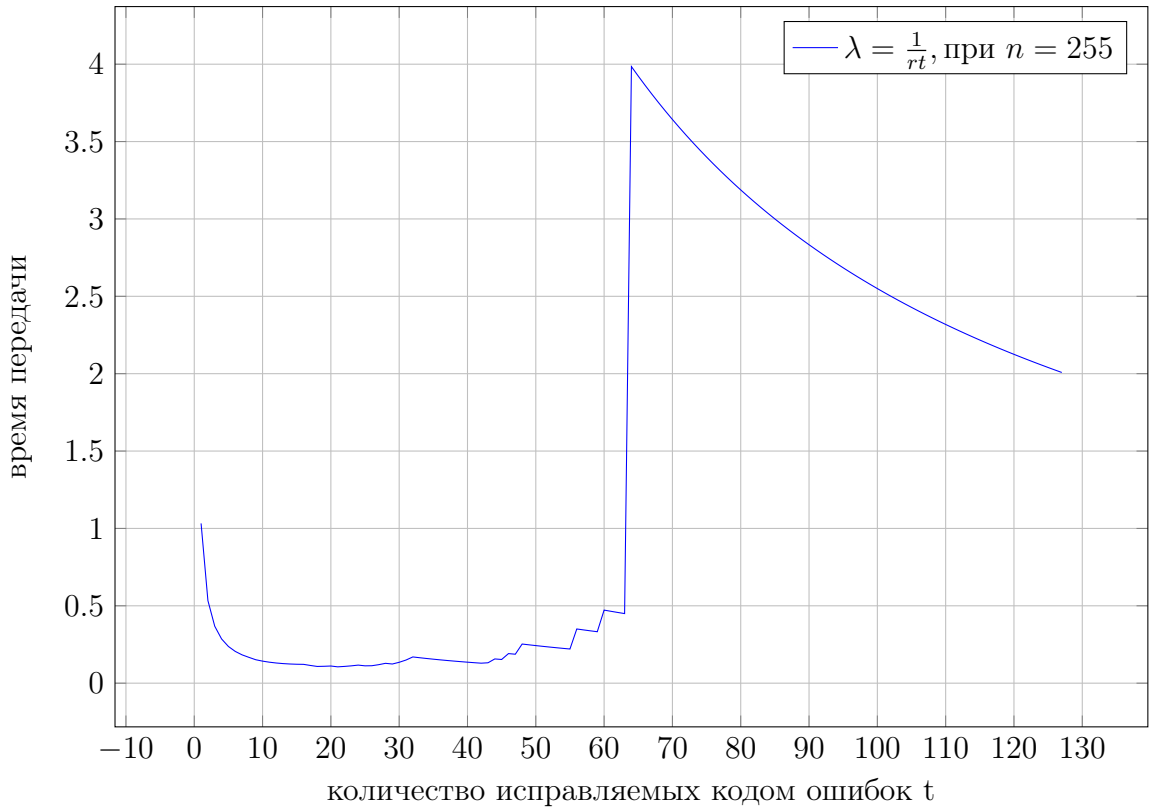
Приведем графики $\lambda(t)$ для $n = 15, 31, 63, 127, 255$.




$\lambda = \frac{1}{rt}$, при $n = 31$
время передачи
количество исправляемых кодом ошибок t


$\lambda = \frac{1}{rt}$, при $n = 63$
время передачи
количество исправляемых кодом ошибок t





Характерным для всех этих графиков является наличие “скачка” - момента, после которого увеличивать число исправляемых кодом ошибок оказывается очень

не выгодно.

Таким образом, для $n = 127$ оптимальным значением оказалось $t = 13$, а для $n = 255$, $t = 21$. Но как видно по графикам нет особо сильной принципиальности в выборе именно таких t. Так для $n = 127$ можно позволить себе выбирать t примерно в промежутке от 9 до 23 и если мы имеем дело с каналом связи с большим количеством помех, то можно отдать предпочтение большему t.

Можно заметить, что на графике зависимости r от t есть участки, когда r остается постоянным некоторое время. Для таких участков очевидно, что если мы и выбрали код, исправляющий кол-во ошибок для t из этого участка, то надо выбирать наибольшее такое t для этого участка постоянства r. Потому что таким образом характеристика кода не ухудшается, а теоретическое кол-во ошибок, которое может исправлять код, растет.

2.2 БЧХ коды, для которых истинное минимальное расстояние больше чем 2t+1

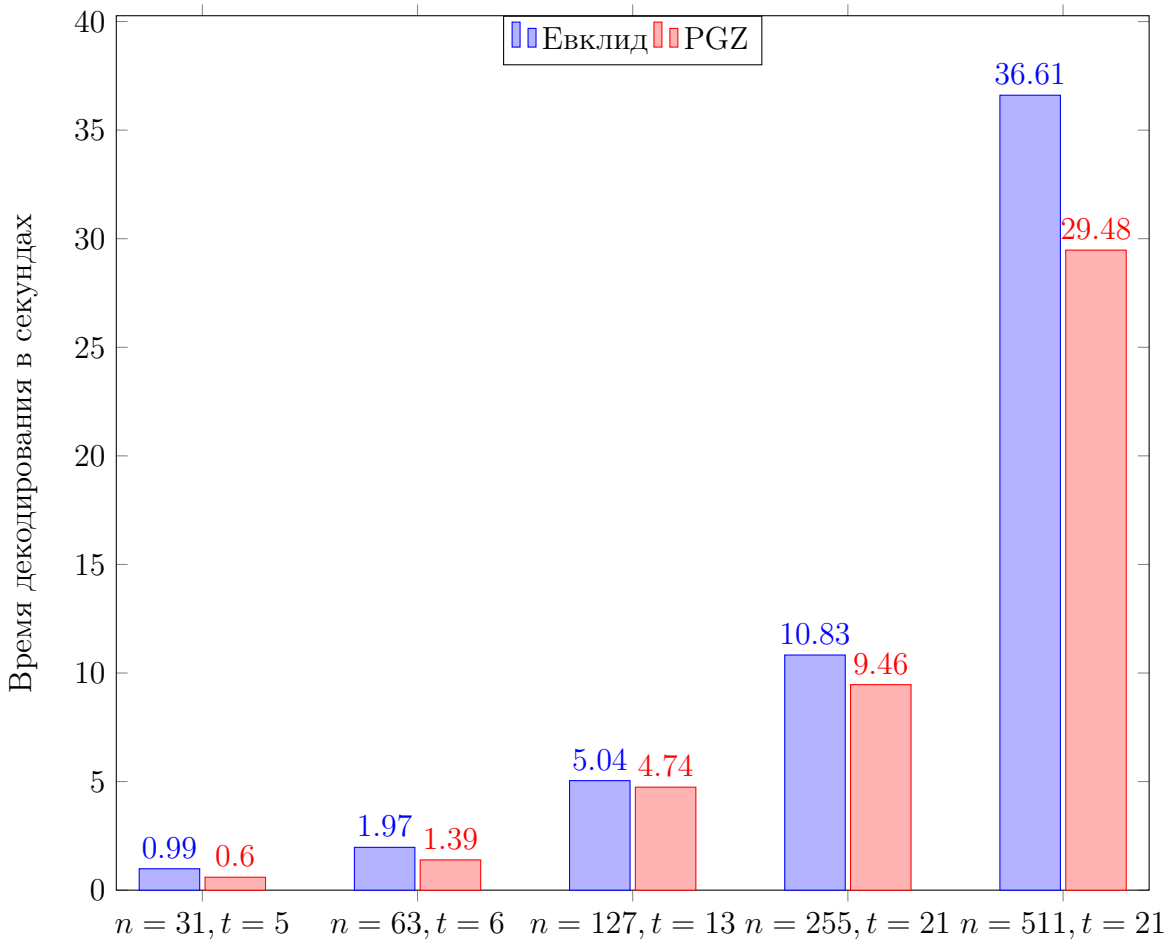
Приведем некотрые примеры БЧХ кодов, для которых путем перебора по кодовым словам было получено, что их истинное минимальное расстояние больше, чем $2t + 1$:

n	t	$2t + 1$	истинное минимальное расстояние кода
7	2	5	7
15	4	9	15
15	5	11	15
15	6	13	15
31	4	9	11
31	6	13	15
31	8	17	31
31	9	19	31
31	10	21	31
31	11	23	31
31	12	25	31
31	13	27	31
31	14	28	31

Если посмотреть на графики зависимости r от t, то можно увидеть, что есть моменты, когда мы увеличиваем t, а скорость БЧХ-кода r при этом не меняется. Логически это воспринимается, как то, что мы требуем от кода большего числа исправляемых ошибок, а код при этом не ухудшает свои характеристики. И интуитивно понятно, что если и существуют коды, с истинным кодовым расстоянием больше чем $2t + 1$, то это такие коды, для которых t это все точки из промежутка постоянства r, кроме крайней правой точки этого промежутка. Что подтвердилось экспериментальным путем, т.е. табличкой, приведенной выше.

2.3 Сравнение времени работы декодирования БЧХ-кода с помощью PGZ и на основе расширенного алгоритма Евклида

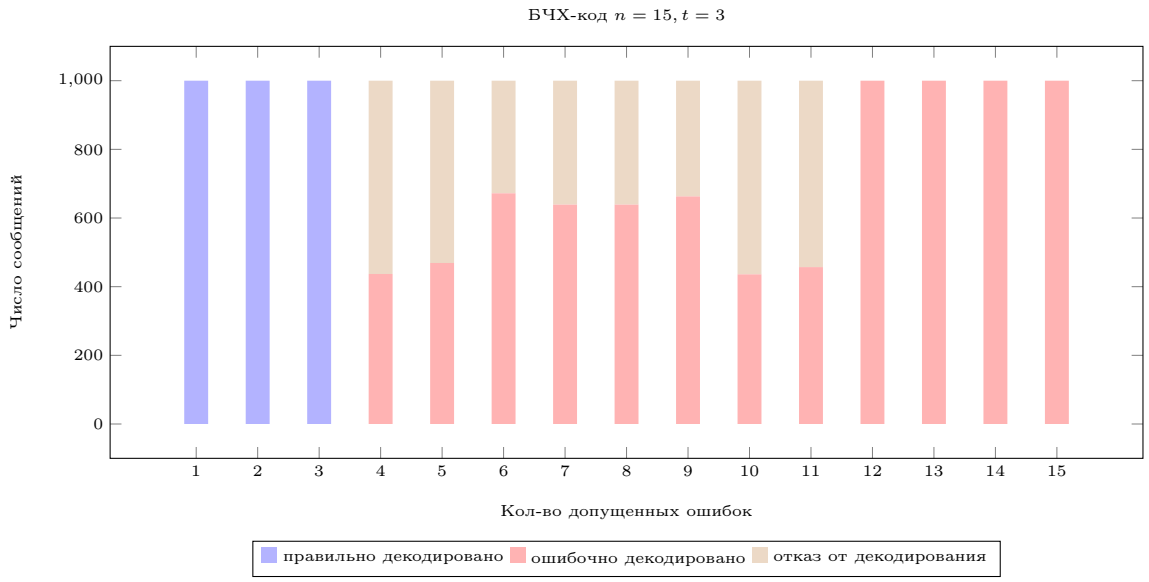
Ниже приведен график, отражающий время работы алгоритма декодирования с помощью PGZ и на основе расширенного алгоритма Евклида. Замеры времени проводились для 100 кодовых слов, в которые было добавлено случайное (до t) кол-во ошибок.

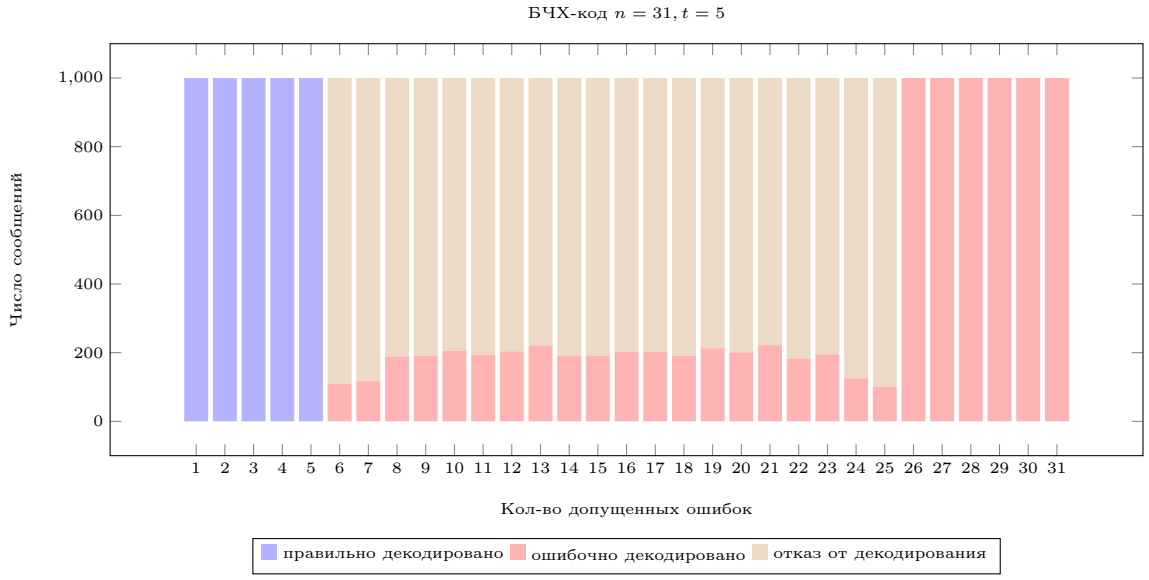


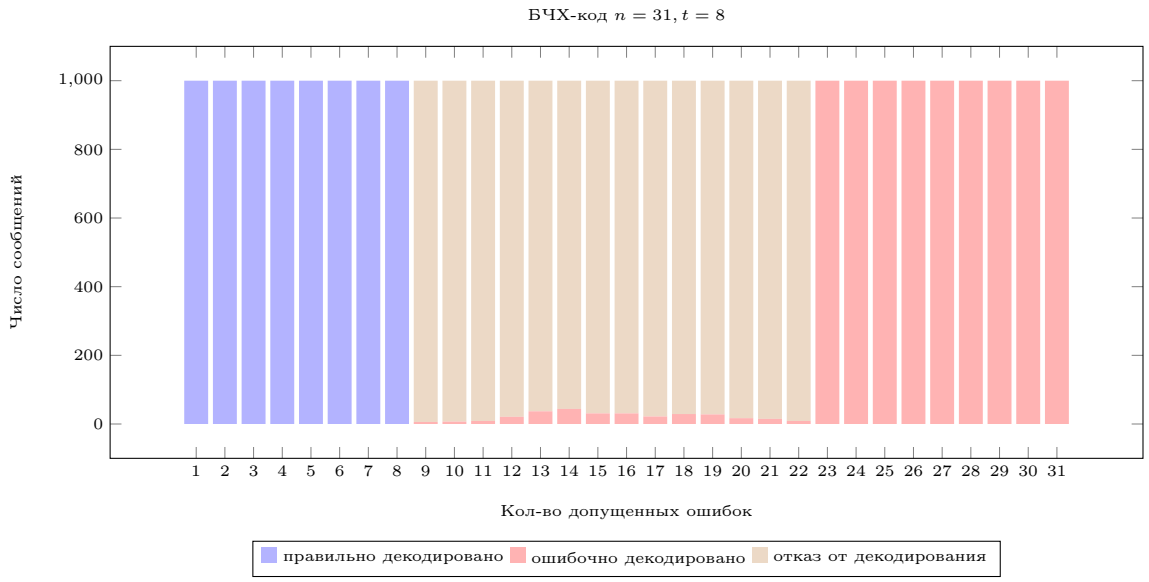
2.4 Статистические испытания

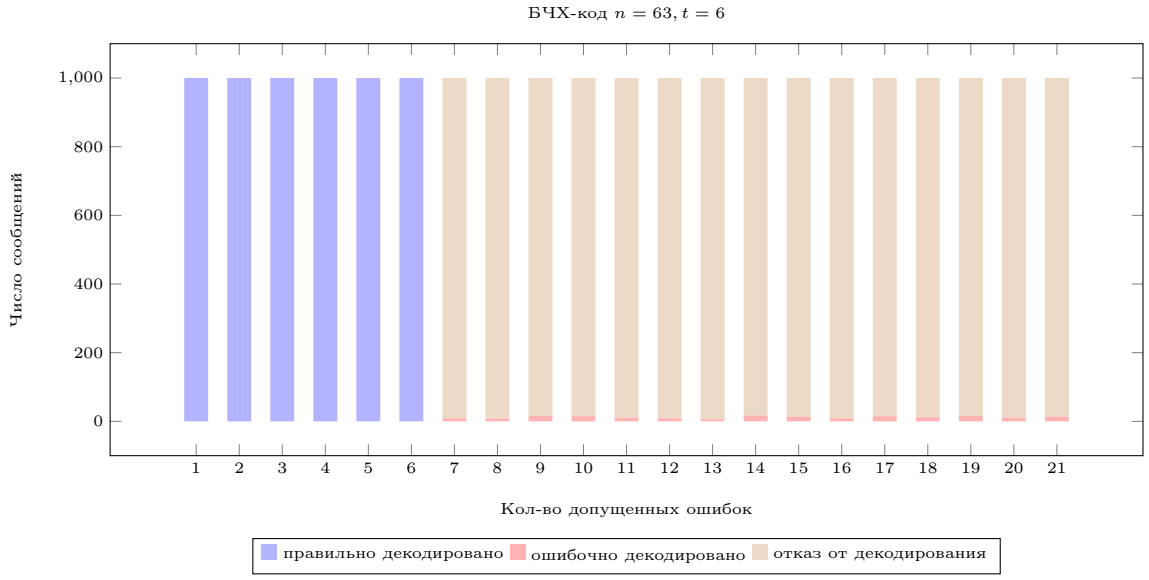
С помощью метода стат. испытаний была реализована процедура оценки доли правильно раскодированных сообщений, доли ошибочно раскодированных сообщений и доли отказов от декодирования для БЧХ-кода.

Производилось заранее предопределенное кол-во ошибок в кодовых словах, но в случайных позициях. С помощью этой процедуры удалось убедиться в том, что БЧХ-код действительно позволяет гарантированно исправить до t ошибок. Результаты экспериментов приведены ниже.


БЧХ-код $n = 15, t = 3$
Число сообщений
1,000
800
600
400
200
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Кол-во допущенных ошибок
правильно декодировано
ошибочно декодировано
отказ от декодирования


БЧХ-код $n = 31, t = 5$
Число сообщений
1,000
800
600
400
200
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Кол-во допущенных ошибок
правильно декодировано
ошибочно декодировано
отказ от декодирования





Смотря на получившиеся статистические данные мы убеждаемся, что БЧХ-код не может исправить больше чем t ошибок. Даже БЧХ-код $n = 31, t = 8$, для которого истинное минимальное расстояние равно $31 > 2 * t + 1 = 17$, не может исправить больше 8 ошибок.

По графикам также видно, что при кол-ве ошибок превыщающем t БЧХ-код сначала в большом проценте случаев дает отказ от декодирования, а потом начинает уверенно раскодировать сообщения неправильно.