Э. П. ЗОГРАБЯН

О РЕАЛИЗАЦИИ АЛГОРИТМА СИНТЕЗА ЛИНЕЙНЫХ КОДОВ

1°. Почти во всех теоретических работах, представляющих непосредственный практический интерес, исследуются системы кодирования, допускающие исправления не более $r>0$ случайных и независимых канальных ошибок; отсюда и синтез систем сигналов с заданным кодовым расстоянием. Подобная интерпретация канальных ошибок системы почти никогда не соответствует характеру и интенсивности помех в реальных условиях. Поэтому представляется важным рассмотрение этой проблемы в наиболее общей ее постановке, а именно: для любой дискретной системы связи с произвольной (но известной) статистической структурой источников сообщения и шумов, способной передавать сигналы[*)] из q различных элементарных символов длины n, указать систему линейного кодирования, сколь угодно надежно передающую сообщения, а также скорость, с которой возможно осуществить эту надежную связь.

Пусть статистическая система связи описывается вероятностями $P_c(x)$ и $P_ш(\varepsilon/x)$. Здесь $P_c(x)$ — вероятность появления на выходе передатчика сигнала x, $P_ш(\varepsilon/x)$ — условная вероятность появления помехи ε в момент прохождения по каналу сигнала x. Далее, пусть $P(\varepsilon)=\sum\limits_{x\in G} P_c(x)P_ш(\varepsilon/x)$, где G — множество рабочих сигналов x, $P(\varepsilon)$ — частота появления помехи. Без ограничения общности можно предположить, что $P_ш(\varepsilon/x)=P_ш(\varepsilon)$; тогда $P(\varepsilon)=P_ш(\varepsilon)$.

Пусть U означает множество всех допустимых канальных помех ε ($U\subseteq G_{n,q}$). Расположим все векторы, принадлежащие множеству U, в порядке убывания их вероятностей, т. е. $P(\varepsilon^1)\geqslant P(\varepsilon^2)\geqslant\cdots$ $\cdots\geqslant P(\varepsilon^u)$, где $u\leqslant q^n$ — порядок U. Поскольку $P(U)=\sum\limits_{\varepsilon\in U}P(\varepsilon)=1$, то для любого положительного числа $\sigma<1$ однозначно определится наименьшее целое $k(\sigma)\leqslant u$ такое, что $P(U_{k(\sigma)})=\sum\limits_{i=1}^{k(\sigma)}P(\varepsilon^i)\geqslant\sigma$,

---

*) Условно сигналы, с помощью которых передаются сообщения, рассматриваются как элементы векторного n-мерного пространства над полем Галуа $GF(q)$.

Обозначим через V множество всех возможных векторов $v = \varepsilon^i - \varepsilon^j$ $(i < j \leqslant k(\sigma))$ и через Н — линейное преобразование в $G_{n,q}$, удовлетворяющее условию

$$V \mathrm{H} \overline{\ni} \Theta = (0, 0, \cdots, 0). \tag{1}$$

Рассмотрим более подробно, что из себя представляет преобразование Н. Для этого для любой пары натуральных чисел $j \leqslant n$ и $i < q$ введем в рассмотрение:

а) множество $V_i(j)$ — совокупность всевозможных векторов v из V, у которых $v_j = i$, причем $v_k = 0$ $(k > j)$;

в) множество $V_{i^0}(j) = i^0 V_i(j) + e_j$, где e_j — единичный n-мерный вектор $(e_j^j = 1)$, i^0 — решение сравнения $ix \equiv -1 \pmod q$;

с) множество $V(j) = \bigcup\limits_{i=1}^{q-1} V_{i^0}(j)$. Так как $\bigcup\limits_{i=1}^{q-1} V_i(j) \subseteq V$, то в силу (1) будем иметь $\bigcup\limits_{i=1}^{q-1} i^0 V_i(j) \mathrm{H} \overline{\ni} \Theta$, или, что то же самое, $V(j) \mathrm{H} \overline{\ni} e^j \mathrm{H}$, или, наконец, в матричной форме

$$V(j)\, H \overline{\ni} h^j, \tag{2}$$

где h^j — j-я строка матрицы H.

Отсюда видно, что в качестве j строки матрицы H может служить любой вектор из множества $G_{n,q} | V(j)$ Н. Задачу отыскания матрицы H с условием (2) будем решать путем последовательного выбора элементов h^j один за другим, тогда для полного ее решения, как легко показать, необходимо

$$\log_q k(\sigma) \leqslant r \tag{3}$$

и достаточно

$$\log_q (g+1) \leqslant r, \tag{4}$$

где r — ранг матрицы H, $g = \max\limits_j \{g(j)\}$, $g(j)$ — порядок $V(j)$.

Таким образом, согласно [1], для обеспечения передачи с заданной точностью $Q_{\text{ср}} = \sigma$ при скорости передачи $R \geqslant 1 + n^{-1}[-\log_q(g+1)]$, достаточно указать систему линейного кодирования, осуществляемую преобразованием К, которое переводит пространство $G_{n-r,q}$ в ядро M преобразования Н.

Следует отметить, что изложенным методом можно получить, в частности, ранее известные результаты. Так, например, для кода, исправляющего r случайных и независимых ошибок, можно показать, что условия (3) и (4) совпадают с оценками Хэмминга, соответственно Варшамова — Гилберта [2].

2°. Для реализации описанного алгоритма разработано программно-управляемое устройство. Рассмотрим основные предпосылки, которые учитывались при составлении структурной схемы устройства.

Как уже было отмечено, в качестве j-ой строки матрицы можно выбрать любой вектор из множества $G_{n,2}/V(j)$ H. Однако с целью ограничения числа переборов эту область можно уменьшить, не нарушая условий (3) и (4). При этом также увеличивается вероятность получения оптимальных кодов. С учетом этого фактора предусмотрены два режима работы устройства:

режим направленного поиска строк матрицы Н (НП);

режим случайного поиска (СП).

В режиме направленного поиска при отыскании любой строки (в частности, j-ой) в фиксированной ячейке Я (h), по мере необходимости, последовательно вырабатываются векторы

$$h_0 = 00 \cdots\cdots 000; \; h_1 = 00 \cdots\cdot 001; \; h_2 = 00 \cdots\cdot 010;$$

$$h_3 = 00 \cdots\cdot 011; \; h_4 = 00 \cdots\cdot 0100, \text{ и т. д.}$$

Каждый из них проверяется на условие

$$V(j)\, H \overline{\ni} h_i. \tag{5}$$

Первый же вектор из этой последовательности, удовлетворяющий (5), и определяет j-ю строку матрицы H. Идея такого метода построения матрицы H заключается в том, что прежде чем выбрать в качестве очередной j-ой строки вектор, линейно независимый по отношению к предыдущим строкам, делаются попытки использовать для этой цели вектор, являющийся линейной комбинацией уже выбранных строк. Очевидно, что построенная матрица будет иметь ранг

$$r \leqslant \log_2 (g+1). \tag{6}$$

Для кодов, синтезируемых по описанному методу и исправляющих однократные ошибки, условия (3) и (4) совпадают, т. е. получаем оптимальные коды.

Действительно, в этом случае

$$g = k(\sigma) - 1. \tag{7}$$

Подставляя (7) в (6), имеем

$$r \leqslant \log_2 k(\sigma). \tag{8}$$

Из условий (3) и (8) следует

$$r = \log_2 k(\sigma). \tag{9}$$

Однако режим направленного поиска имеет тот недостаток, что для одной и той же совокупности помех, подлежащих исправлению, невозможно получить класс кодов с тем, чтобы среди них выбрать наилучший. Для этой цели и предусмотрена возможность работы устройства в режиме случайного поиска. Кроме того, метод СП позволит использовать устройство для исследования внутренней структуры и взаимосвязи между помехами и получаемыми кодами.

В качестве генератора псевдослучайных векторов применен регистр сдвига (рис. 1), обратная связь которого соответствует примитивному полиному $\varphi(x) = x^{32} + x^{22} + x^2 + x + 1$. Степень полинома совпадает с числом разрядов регистра. Такие регистры порождают последовательности максимальной длины [2].

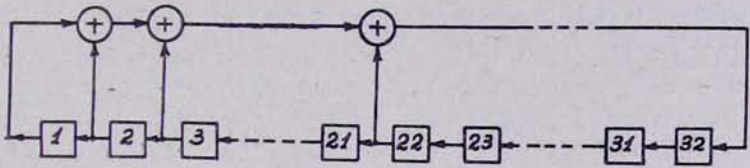

Рис. 1.

Функцию регистра сдвига выполняет универсальный регистр (рис. 2). При выполнении каждой команды, предусмотренной в программе, производится автоматический сдвиг регистра, в котором в определенной последовательности образуются все $2^{32} - 1$ векторы. Эффект случайности получается благодаря тому, что обращение к регистру за псевдослучайным вектором в ходе выполнения программы носит случайный характер и зависит от количества векторов-помех, от их вида и порядка расположения в ЗУ, а также от первого случайного заполнения регистра.

В нужный момент из универсального регистра (УР) считываются в качестве псевдослучайного вектора последние $r < N$ символов, содержащегося в нем вектора длины $N = 32$. Таким образом, множество, из которого выбираются псевдослучайные последовательности, сводится к векторному пространству $G_{r,2}$.

Величина r вычисляется по формуле

$$r =]\log_2(g + 1)[, \tag{10}$$

где $g = \max\limits_j \{k_{1j} \cdot k_{2j}\}$, k_{1j} — количество помех, у которых $\varepsilon_j = 1$, а для всех $m > j$ $\varepsilon_m = 0$, k_{2j} — количество помех, у которых $\varepsilon_m = 0$ для всех $m \geqslant j$. Функция $]x[$ — наименьшее целое число, не меньше x.

3°. Необходимо отметить, что программа реализации алгоритма составлена из расчета уменьшения объема памяти за счет некоторого увеличения времени его выполнения. Вся программа, рабочие ячейки, постоянные величины и построенная матрица занимают 90—110 ячеек ЗУ, в остальные ячейки ЗУ могут быть записаны в определенной последовательности заданные векторы-помехи.

Рассмотрим вкратце работу блок-схемы программы на примере построения j-ой строки матрицы H.

1. В рабочей ячейке Я(h) фиксируется очередной псевдослучайный вектор h.

2. Считывается пара помех ε^i, ε^j и определяется $v = \varepsilon^i \oplus \varepsilon_j$. Анализируется принадлежность вектора v множеству $V_1(j)$. Возможны два исхода:

$$\text{I} \quad v \overline{\in} V_1(j) - \text{идти к пункту 3.}$$

$$\text{II} \quad V \in V_1(j) - \text{идти к пункту 4.}$$

3. Индексы i и j при ε подвергаются изменению по определенному закону и анализируются операторами „конец j", „конец i" $(i = \overline{1, (k(\varepsilon) - 1)};\ \ j = \overline{(i+1), k(\varepsilon)})$. Если все комбинации $\varepsilon^i \oplus \varepsilon^j$ рассмотрены — идти к пункту 5, в противном случае — идти к пункту 2.

4. Результат сложения v по mod 2 с единичным вектором e^j умножается на частично построенную матрицу H. Вектор $(v \oplus e^j)\, H$ сравнивается с h. При этом:

$$\text{если } (v \oplus e^j)\, H = h - \text{идти к пункту 1,}$$

$$\text{если } (v \oplus e^j)\, H \neq h - \text{идти к пункту 3.}$$

5. Зафиксированный псевдослучайный вектор записывается в качестве j-ой строки матрицы H и выдается на печать.

6. Оператор „конец реализации алгоритма" анализирует значение параметра, фиксирующего количество найденных строк: если не вся матрица построена — идти к пункту 1 (программа целиком повторяется для нахождения $(j+1)$-ой строки), если вся матрица построена — идти к пункту 7.

7. „Останов".

4°. Отметим некоторые используемые приемы программирования. Вычисление величины $r =]\log_2(g+1)[$ осуществляется следующим образом. Известно, что целое число $g+1>0$ в двоичной системе исчисления представляется последовательностью коэффициентов $\alpha_k, \alpha_{k-1}, \alpha_{k-2} \cdots \alpha_1$, где $\alpha_k = 1$, $\alpha_i = 0$, или 1 $(i<k)$. Будем считать, что индексы при α представляют номера разрядов. Тогда

$$r =]\log_2(g+1)[= \begin{cases} k-1, & \text{если } \alpha_{k-1} = \alpha_{k-2} = \cdots = \alpha_1 = 0 \\ k, & \text{если среди } \alpha_{k-1}, \alpha_{k-2} \cdots, \alpha_1, \\ & \text{имеется хотя бы один } \alpha_i = 1. \end{cases} \tag{11}$$

Последовательным сдвигом на один разряд вправо вектора $e^n = (10 \cdots \cdots 0)$ и логическим умножением на $g+1$ на k-ом шаге выделяется $e^k = (0 \cdots 010 \cdots 0)$. Далее, анализируются разряды числа $g+1$ и, согласно (11), e^k сдвигается еще на один разряд или оставляется без изменения, образуя $e^r = (0 \cdots 010 \cdots 0)$. Заменой всех нулей в e^r правее r-го разряда на единицы получается последовательность $(0 \cdots 011 \cdots 1)$, которая используется для выделения псевдослучайного вектора.

Рассмотрим процесс умножения вектора $v \in V(j)$ на матрицу H. Пронумеруем каждый символ вектора v слева направо и, соответственно, каждую строку частично построенной матрицы H сверху вниз. Тогда vH можно представить в виде

$$vH = v_1 h_1 \oplus v_2 h_2 \oplus \cdots\cdots \oplus v_{j-1} h_{j-1}, \tag{12}$$

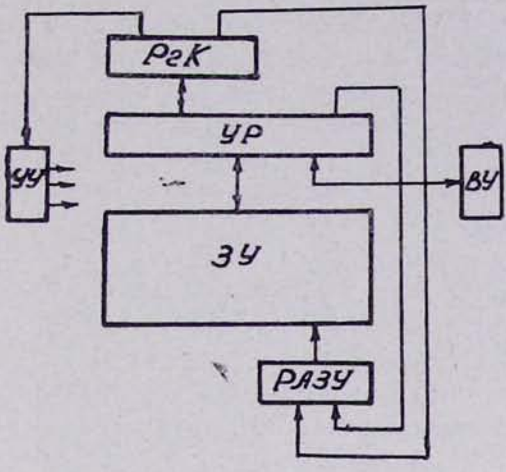

Рис. 2.

где v_i — i-ый символ вектора v, равный 0, или 1, h_i — i-я строка матрицы H. Благодаря (12) в устройстве используется следующий прием. Последовательно анализируется каждый символ вектора v и если $v_i = 1$, то соответствующая ему строка-вектор h_i складывается по mod 2 с вектором, представляющим частичную сумму строк, полученную в результате анализа предыдущих символов. Если $v_i = 0$, то к частичной сумме ничего не прибавляется. Процесс умножения завершается анализом $(j-1)$-го символа, и полученная результирующая сумма-вектор представляет vH.

5°. Перейдем к описанию блок-схемы программно-управляющего устройства (рис. 2). Устройство состоит из следующих узлов.

I. Запоминающее устройство (ЗУ) предназначено для записи, хранения и выдачи команд, векторов, чисел и конечных результатов вычислений. Емкость ЗУ 512 32-разрядных чисел. Первые 3 ячейки (a_1, a_2, a_3) выполняют функции вспомогательных регистров.

II. Регистр адреса ЗУ (РАЗУ) предназначен для приема и дешифрации адресов. Состоит из 9 разрядов.

III. Универсальный регистр (УР) содержит 32 разряда. Выполняет логические операции над числами и векторами, операции сдвига и счета, служит регистром числа для ЗУ, используется в качестве генератора, порождающего последовательности максимальной длины.

IV. Регистр команд (РгК) принимает и хранит код операции и адрес второго операнда. РгК содержит 13 разрядов, разряды с 10 по 13 отведены для кода операции.

V. Устройство управления (УУ) предназначено для выработки управляющих сигналов и для автоматического управления устройства при выполнении заданной программы. В устройстве принят естественный порядок выполнения команд.

Работу УУ для выполнения одной операции типа „сложение“ можно разделить на следующие этапы:

1. Чтение из ЗУ содержимого первой ячейки a_1, в которой хранится адрес очередной команды, передача его в РгК, формирование адреса следующей команды и запись его в a_1.

2. Выборка кода команды из ЗУ и передача из УР кода операции и второго адреса в РгК, расшифровка кода операции. Передача первого адреса из УР в РАЗУ, прием первого числа и запись во вторую ячейку a_2.

3. Передача второго адреса из РгК в РАЗУ, выборка, чтение содержимого второй ячейки с выполнением операции.

4. Запись результата в ЗУ по второму адресу команды и во вторую ячейку a_2.

5. Чтение третьей ячейки a_3, в которой хранится псевдослучайный вектор, образование новой последовательности, запись. (В режиме НП этот этап отсутствует).

6. Выработка сигнала возврата к этапу 1. (При операции „Останов“ этот сигнал не вырабатывается).

VI. Внешнее устройство (ВУ) предназначено для ввода информации в устройство и вывода результатов вычислений. ВУ состоит из аппарата типа СТА-2М и местного управления. При кодировании информации на перфоленте используется 16-ичная система счисления (5-ый разряд на перфоленте не используется).

Местное управление внешнего устройства осуществляет формирование вводимой и выводимой информации, а также анализ символа конца ввода (вывода) информации.

В таблице приведена система операций и команд, используемых в устройстве.

6°. Оценим математическое ожидание $E_n(t)$ времени получения матрицы H для случая работы устройства в режиме случайного поиска.

Обозначим число всех элементов множеств: V, $V(j)\,H$ и $G_{r,2}$ соответственно через m, $m(j)$ и $m(r)$ (где $G_{r,2}$ — векторное пространство, из которого выбираются псевдослучайные векторы h_i); число различных элементов множества $V(j)\,H$ через $\overline{m}(j)$. Будем считать, что все $h_i\ (i=\overline{1, 2^r})$ выбираются независимо и равновероятно.

Тогда вероятность выбора каждого псевдослучайного вектора равна $\frac{1}{m(r)}$, а вероятность совпадения h_i с вектором из $V(j)\,H$ — $\frac{\overline{m}(j)}{m(r)}$.

Код операций	Наименование операции и вид команды	Краткое содержание команды	Пояснение
1	Сложение $1\ A_1A_2$	$(A_2)+(A_1)\to A_2,\ a_2$	Выражение (A_1) означает содержимое первого адреса A_1
2	Вычитание $2\ A_1A_2$	$(A_2)-(A_1)\to A_2,\ a_2.$	Знак $\to$ означает запись
3	Передача числа $3\ A_1A_2$	$(A_1)\to A_2$	
4	Передача числа $4\ A_2$	$(a_2)\to A_2$	
5	Сложение по mod 2 $5\ A_1A_2$	$(A_2)\oplus(A_1)\to a_2$	
6	Логическое умножение $6\ A_1A_2$	$(A_2)\wedge(A_1)\to a_2$	
7	Условный переход $7\ A_2$	$A_2\to a_1$, если $(a_2)=0$	Анализируется содержимое ячейки a_2, если оно равно 0, в ячейку a_1 записывается A_2
8	Условный переход $8\ A_2$	$A_2\to a_1$, если $(a_2)>0$	
9	Условный переход $9\ A_2$	$A_2\to a_1$, если $(a_2)<0$	
$\overline{0}$	Безусловный переход $\overline{0}\ A_2$	$A_2\to a_1$	
$\overline{1}$	Правый сдвиг $\overline{1}\ A_2$	$(A_2)\,2^{-1}\to A_2,\ a_2$	
$\overline{2}$	Левый сдвиг $\overline{2}\ A_2$	$(A_2)\,2\to A_2,\ a_2$	
$\overline{3}$	Печать $\overline{3}\ A_2$	$(A_2)\to$ на печать	
0	Останов $0\ A_2$	$A_2\to a_1$, останов	A_2 записывается в ячейку a_1 и устройство останавливается

Пусть $A_k^{(j)}$ — событие, состоящее в том, что при всех k выборах (испытаниях) псевдослучайный вектор совпадает с одним из векторов, принадлежащих $V(j)H$, а в $k+1$-ом испытании — не совпадает. Тогда событие $A_k^{(j)}$ будет иметь место с вероятностью

$$P\left(A_k^{(j)}\right)=\left(1-\frac{\overline{m}(j)}{m(r)}\right)\left(\frac{\overline{m}(j)}{m(r)}\right)^k. \tag{13}$$

Обозначим через $S_i^{(j)}$ число сравнений псевдослучайного вектора с векторами из $V(j)H$ до их совпадения в i-ом испытании. $S_i^{(j)}$ $(i=\overline{1,\ k-1})$ являются одинаково распределенными независимыми случайными величинами, принимающими с равными вероятностями значения

$1, 2, \cdots, m(j)$. Поэтому событие $A_k^{(j)}$ будет иметь место при общем числе сравнений, равном

$$\overline{S}_k^{(j)} = S_1^{(j)} + S_2^{(j)} + \cdots + S_k^{(j)} + m(j), \tag{14}$$

где $m(j)$ — число сравнений в $k+1$-ом испытании.

Пусть $A^{(j)}$ — событие, состоящее в том, что найдена j-я строка матрицы H; K и $S^{(j)}$ — соответственно число испытаний и число сравнений, необходимых для наступления события $A^{(j)}$. Таким образом, имеются две случайные величины K и $S^{(j)}$, для которых справедлива формула

$$E(S^{(j)}) = \sum_{k=0}^{\infty} P(A_k^{(j)})\, E(\overline{S}_k^{(j)}). \tag{15}$$

Каждое $S_i^{(j)}$ в (14) имеет математическое ожидание, равное

$$E(S_i^{(j)}) = \frac{1 + 2 + \cdots + m(j)}{m(j)} = \frac{m(j) + 1}{2}.$$

Поэтому

$$E(\overline{S}_k^{(j)}) = \sum_{i=1}^{k} E(S_i^{(j)}) + m(j) = \frac{(k-1)(m(j)+1)}{2} + m(j). \tag{16}$$

Подставляя (13) и (16) в (15), получим

$$E(S^{(j)}) = \frac{m(j)(m(j)+1)}{2(m(r) - m(j))} + m(j). \tag{17}$$

Перед тем как сравнить псевдослучайный вектор h_i с вектором, принадлежащим множеству $V(j)H$, производится поиск и выделение вектора $v \in V(j)$ путем последовательного перебора и анализа векторов, принадлежащих множеству V. Затем производится умножение вектора v на матрицу H и сравнение с h_i. Вероятность того, что выделенный вектор $v \in V(j)$, равна $\frac{m(j)}{m}$. Вероятность того, что после $X^{(j)}$ переборов найдется вектор $v \in V(j)$, равна $\frac{m(j)}{m}\left(1 - \frac{m(j)}{m}\right)^x$. $X^{(j)}$ является случайной величиной, принимающей с равными вероятностями значения $x = 1, 2, \cdots, m$. Математическое ожидание числа переборов для нахождения вектора $v \in V(j)$ равно

$$E(X^{(j)}) = \sum_{x=1}^{m} \frac{m(j)}{m}\left(1 - \frac{m(j)}{m}\right)^{x-1} \cdot x =$$

$$= \frac{m}{m(j)}\left[1 - \left(1 - \frac{m(j)}{m}\right)^m \left(1 + m(j)\right)\right]. \tag{18}$$

Пусть N_1 — среднее число операций, предусмотренных по программе для получения и анализа каждого вектора $v \in V$; N_2 — число операций, необходимых для перемножения вектора $v \in V(j)$ на матрицу H и сравнения его с псевдослучайным вектором; C — средняя скорость вычислений на устройстве (т. е. число операций в секунду). Тогда среднее время $E_j(t)$, необходимое для получения j-ой строки матрицы H, равно

$$E_j(t) = \frac{1}{C} E(S^{(j)}) [N_1 E(X^{(j)}) + N_2]. \tag{19}$$

Суммируя (19) по всем $j = \overline{1, n}$, получим среднее время $E_H(t)$ построения матрицы H.

При $C = 1000$, $N_1 = 10$, $N_2 = 3n$, $k(\sigma) = 100$, $n = 20$, получаем $E_H(t) = 2$ ч., что совпадает с экспериментальным результатом.

В заключение автор выражает благодарность Р. В. Варшамову за интерес и внимание к работе.

Է. Պ. ՋՈՀՐԱԲՅԱՆ

ԳԾԱՅԻՆ ԿՈԴԵՐԻ ՍԻՆԹԵԶՄԱՆ ԱԼԳՈՐԻԹՄԻ ԻՐԱԿԱՆԱՑՄԱՆ ՄԱՍԻՆ

Ամփոփում

Նկարագրվում է աղմուկների կամայական ստատիստիկական կառուցվածք ունեցող աղբյուրով դիսկրետ կապի սիստեմի համար գծային կոդերի սինթեզման մի ալգորիթմ։ Դիտարկվում է ծրագրով ղեկավարվող մասնագիտացված սարքի օգնությամբ այդ ալգորիթմի իրականացման երկու մեթոդ։

Մեթոդներից մեկի համար հաշվված է իրականացման ժամանակի մաթեմատիկական սպասողականությունը։

ЛИТЕРАТУРА

1. *Р. Р. Варшамов.* Математические методы повышения надежности реальных систем связи. Изв. АН СССР, серия тех. кибернетика, № 4, 1964, 53—58.
2. *У. Питерсон.* Коды, исправляющие ошибки. Изд-во „Мир“, 1964.