ВЕКТОРНЫЙ СПОСОБ ПОСТРОЕНИЯ ГРУППОВЫХ ДВОИЧНЫХ КОДОВ

И. Н. Бута, И. М. Израйлит

Ленинград

В системах связи целесообразно иметь возможность регулирования параметров путем *слежения за состоянием канала* и выбора способа кодирования. Это выдвигает проблему конструирования кодопреобразователя, способного генерировать различные корректирующие коды с возможностью *изменения скорости телеграфирования и повторения* кодовых комбинаций заданное число раз.

Допустим, что по каналу связи передается сообщение $(\alpha_1, \ldots, \alpha_k)$, закодированное двоичным корректирующим кодом. Для групповых кодов j-ый контрольный символ n-значной комбинации является линейной функцией исходного сообщения

$$a_j = f(\alpha_1, \alpha_2, \ldots, \alpha_k).$$

Это выражение можно представить следующим образом:

$$a_j = \bigoplus \sum_{i=1}^{k} \gamma_{ij} a_i,$$

где $\gamma_{ij} = 0$ *или* 1 *— коэффициент, характеризующий* зависимость j-го символа кодограммы от i-го информационного символа; знак $\bigoplus$ означает посимвольное суммирование произведений $\gamma_{ij}a_i$ по модулю «два».

Множество коэффициентов $\{\gamma_{ij}\}$ *определяет производящую матрицу*

$$G = \begin{pmatrix} \gamma_{11} \cdots \gamma_{1n} \\ \cdots \\ \gamma_{k1} \cdots \gamma_{kn} \end{pmatrix},$$

каждая строка которой представляет собой образующий вектор $Y_i = (\gamma_{i1}, \ldots, \gamma_{in})$ для которого информационный символ a_i является общим множителем. *Результирующая кодограмма определяется произведением* $(\alpha_1, \ldots \alpha_k)\, G$.

Тогда алгоритм построения групповых двоичных кодов можно представить в виде

$$F = \bigoplus \sum_{i=1}^{k} a_i Y_i. \qquad (1)$$

Набор образующих векторов, определяемых алфавитом источника и построенных согласно частным алгоритмам кодов, позволяет генерировать в одном устройстве различные групповые коды (Слепяна, Хемминга, Боуза—Чоудхури и др.). Изменение значности кода обеспечивает изменение его корректирующей способности. Многократность повторения кодо-

грамм достигается задержкой каждой последовательности информационных символов на заданное число временных интервалов, соответствующих длительности кодограмм.

Групповые коды позволяют выделять в явном или неявном виде информационные и контрольные символы. В кодах с неявным разделением символов образующие векторы могут быть заданы непосредственно. Единичные символы вектора указывают на связь данного контрольного символа с информационным. В этом случае коэффициент $\gamma_{ij} = 1$; при отсутствии такой связи $\gamma_{ij} = 0$.

Для кодов с явным разделением символов значения γ_{ij} могут быть найдены следующим образом:

определяются информационные позиции кодограмм

$$a_j \equiv \alpha_i; \tag{2}$$

уравнения проверок, согласно частному алгоритму кода, представляются в виде

$$a_j = f(\alpha_1, \alpha_2, \ldots, \alpha_k), \tag{3}$$

для каждого контрольного символа a_j в матрицу G вписывается значение коэффициента

$$\gamma_{ij} = \begin{cases} 1 — \text{при наличии в уравнениях (2) и (3) символа } \alpha_i; \\ 0 — \text{при отсутствии в уравнениях (2) и (3) символа } \alpha_i. \end{cases}$$

Составленные таким образом двоичные последовательности служат образующими векторами для генерации заданного кода.

Пример. (12,7)-код Слепяна. Алгоритм кода задается уравнениями [4]:

$$a_1 = \alpha_1;\ a_2 = \alpha_2;\ a_3 = \alpha_3;\ a_4 = \alpha_4;\ a_5 = \alpha_5;\ a_6 = \alpha_6;\ a_7 = \alpha_7;$$

$$a_8 = \alpha_1 \oplus \alpha_2 \oplus \alpha_4 \oplus \alpha_6 \oplus \alpha_7;\ a_9 = \alpha_1 \oplus \alpha_2 \oplus \alpha_4 \oplus \alpha_5 \oplus \alpha_6 \oplus \alpha_7;$$

$$a_{10} = \alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_5 \oplus \alpha_6 \oplus \alpha_7;\ a_{11} = \alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_6 \oplus \alpha_7;$$

$$a_{12} = \alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 \oplus \alpha_7.$$

Образующими векторами будут: $Y_1 = 100000011111$, $Y_2 = 010000001111$, $Y_3 = 001000010111$, $Y_4 = 000100011101$, $Y_5 = 000010011101$, $Y_6 = 000001011110$, $Y_7 = 000000111111$. Тогда, например, для сообщения 1010011 получаем кодограмму 101001101001. Нетрудно убедиться, что такой же результат получим при обычном посимвольном кодировании.

Пример. (12,8)-код Хэмминга. Алгоритм кода описывается уравнениями [4]

$$a_1 = \alpha_1 \oplus \alpha_2 \oplus \alpha_4 \oplus \alpha_5 \oplus \alpha_7;\ a_2 = \alpha_1 \oplus \alpha_3 + \alpha_4 \oplus \alpha_6 \oplus \alpha_7;$$

$$a_3 = \alpha_1;\ a_4 = \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_8;\ a_5 = \alpha_2;\ a_6 = \alpha_3;\ a_7 = \alpha_4;$$

$$a_8 = \alpha_5 \oplus \alpha_6 \oplus \alpha_7 \oplus \alpha_8;\ a_9 = \alpha_5;\ a_{10} = \alpha_6;\ a_{11} = \alpha_7;\ a_{12} = \alpha_8.$$

Образующие векторы имеют следующий вид:

$Y_1 = 111000000000$; $Y_2 = 100110000000$; $Y_3 = 010101000000$;
$Y_4 = 110100100000$; $Y_5 = 100000011000$; $Y_6 = 010000010100$;
$Y_7 = 110000010010$; $Y_8 = 000100010001$.

Кодопреобразователь, реализующий алгоритм (1), содержит формирователь коммутационных импульсов (ФКИ), матричный формирователь образующих векторов (МФ), блок управления (БУ) и сумматор по модулю два (С). Формирователь ФКИ имеет n выходов, на которых последовательно во времени выдается коммутирующий импульс; ФКИ может быть выполнен на регистрах сдвига, триггерных счетчиках и т. д. Импульсы ФКИ поступают на МФ, воспроизводящий импульсные последовательности

$\{Y_i\}$. Матричный формирователь может быть выполнен на диодах или ферритовых элементах, установленных в точках пересечения вертикальных и горизонтальных шин матрицы G и коммутируемых механически или электронным способом. Импульсные последовательности $\{Y_i\}$ и информационные символы $\{\alpha_i\}$ подаются на БУ, который содержит k схем информационного управления последовательностями $\{Y_i\}$, представляющих собой схемы «И» или ключевые каскады. Информационное управление осуществляется символами $\{\alpha_i\}$ в течение всего цикла формирования кодограммы. Сумматор С на k входов служит для поразрядного суммирования векторов $\{Y_i\}$, поступающих с БУ; для его реализации требуется $k-1$ элементов на два входа. Наиболее простыми сумматорами такого типа являются биполярные [1].

Кодопреобразователи указанного типа наиболее целесообразно использовать в каналах связи, когда в зависимости от изменяющихся условий требуются различные коды. Векторный способ кодирования удобен также и для каналов с постоянными параметрами. Например, для некоторых кодов Хемминга этот способ с точки зрения расхода радиодеталей эквивалентен посимвольному кодированию, либо даже экономичнее его. Для кодов Слепяна векторное кодирование наиболее эффективно, когда $n \approx 2k$. Так, при посимвольном генерировании (12,7)-кода Слепяна требуется 24 феррит-транзисторных ячейки (ФТЯ), 23 сумматора по модулю два и семь каскадов управления. Формирование этого же кода векторным способом требует 12 ФТЯ, шесть сумматоров по модулю два, семь каскадов управления и 37 диодов.

Кодопреобразователи при векторном способе кодирования можно существенно упростить получая образующие векторы последовательным делением опорной импульсной последовательности на два раза. В этом случае ФКИ и МФ превращаются в простую цепочку триггеров. При этом вместо n-каскадных регистров необходимы k-каскадные регистры. Выигрыш в количестве радиодеталей наиболее заметен, когда $n \gg k$. Принцип деления импульсных последовательностей на два раза применяется в коде Рида—Мюллера. Однако этот код при посимвольном генерировании корректирует ошибки лишь кратности $m = 2^i - 1$ $(i = 1, 2, \ldots)$.

Рассмотрим способ построения корректирующих кодов при мажоритарном декодировании, в котором простота технической реализации сочетается с возможностью исправления заданного числа ошибок m и обнаружения $V = m + 1$ ошибок. Алгоритм кода будем искать в виде

$$F_0 = F^{\nu} + \lambda^{\nu}, \tag{5}$$

где $\nu = k - 1$;

$F^{\nu} = \bigoplus \sum_{i=0}^{\nu} \alpha_i X_i$ — базисный кодовый полином, использующий периодические опорные векторы X_i;

$n_0 < n$ — значность базисного полинома;

n — значность кода;

λ^{ν} — дополнительный полином, определяющий недостающие уравнения проверок для коррекции заданного числа ошибок.

В зависимости от числа недостающих уравнений проверок вид функции $\lambda^{\nu}(\alpha_i)$ может быть различным. Ниже приводятся три способа построения корректирующих кодов, отличающихся методом формирования полинома λ^{ν}.

Первый способ основан на дополнении каждого опорного вектора X_i некоторым числом k_j кодовых символов, где j — число пар недоста

ющих уравнений проверок. Алгоритм построения полинома λ^* опишем функцией

$$\lambda^* = f[a_{n_0+1}(\alpha_1), a_{n_0+2}(\alpha_2), \ldots, a_n(\alpha_1 \oplus \alpha_2 \oplus \ldots \oplus \alpha_\nu)]. \quad (6)$$

При этом справедливы следующие соотношения:

число исправляемых ошибок

$$m = 2^{\nu-2}, \ (\nu = 2, 3, 4, \ldots);$$

значность кода

$$n = 2^{2+\log_2 m} + \log_2 m + 3;$$

число информационных символов

$$k = 3 + \log_2 m.$$

Второй способ непосредственно использует совокупность опорных векторов $X_1, X_2, \ldots, X_{\nu-1}$, обеспечивающую необходимое число уравнений проверок для информационных символов $\alpha_1, \alpha_2, \ldots, \alpha_{\nu-1}$. Управление опорным вектором X_0 в пределах полинома F^* включительно по n_0-ую позицию осуществляется символом α_0, а в пределах полинома λ^* — символом α_ν. При этом для всех символов $\alpha_1, \alpha_2, \ldots, \alpha_\nu$ число уравнений проверок оказывается достаточным, а полином

$$\lambda^* = \alpha_1 X_1 \oplus \alpha_2 X_2 \oplus \ldots \oplus \alpha_{\nu-1} X_{\nu-1} \oplus \alpha_\nu X_0. \quad (7)$$

Этим способом могут быть построены две группы кодов, обладающих различными значениями m при одинаковом числе информационных символов.

1. Группа кодов, для которых

$$5 \cdot 2^{k-5} - 1 \leqslant m \leqslant 3 \cdot 2^{k-4} - 1 \quad (8)$$

$$(k > m).$$

Число N_k различных кодов для данного k определяется выражением

$$N_k = 2^{k-5} + 1$$

$$(k > 5),$$

а значность кода

$$n = 3 \cdot 2^{k-2} - 2(3 \cdot 2^{k-4} - m - 1).$$

2. Группа кодов, для которых

$$3 \cdot 2^{k-2} - 1 < m < 2^{k-2} - 1 \quad (9)$$

$$(k > 4).$$

Число N_k различных кодов для данного k определяется выражением

$$N_k = 2^{k-4} - 1$$

$$(k > 4),$$

а значность кода

$$n = 2^k - 2(2^{k-2} - m - 1).$$

Третий способ заключается в представлении полинома λ^* в виде совокупности векторов, использующих различные варианты информационного управления образующим вектором X_0

$$\lambda^* = \alpha_1 X_1 \oplus \alpha_2 X_2 \oplus \ldots \oplus \left.\begin{matrix} \alpha_{\nu-2} \\ \alpha_{\nu-1} \\ \alpha_\nu \end{matrix}\right\} X_0. \quad (10)$$

При этом вектор X_0 управляется только одним информационным символом. Этот способ позволяет получать коды, обеспечивающие наибольшую скорость передачи информации и характеризуемые соотношениями:

число исправляемых ошибок

$$m = (2^j - 1)\,2^{k-j-r} - 1$$

$$(j = 0, 1, 2, \ldots)$$

(при $j = 1$ это выражение дает значение m, совпадающее с кодом Рида—Мюллера при $r = 1$);

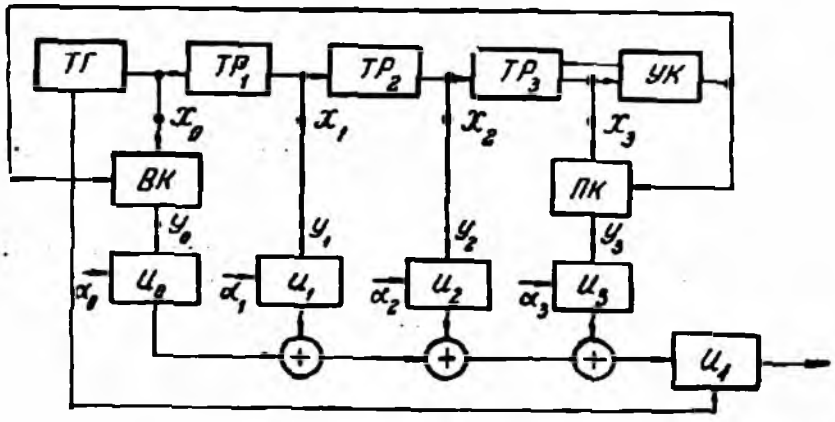


Схема кодопреобразователя.

число информационных символов

$$k = \log_2 4\,(m + 1) - \log_2 (1 - 2^{-j});$$

значность кода

$$n = 4\,(m + 1).$$

Предположим, что необходимо построить кодопреобразователь для кода с параметрами $m = 2$ и $k = 4$. Такой код можно построить и по первому и по второму способам, причем в обоих случаях значность кода $n = 12$. Однако технически более простым оказывается кодопреобразователь, построенный по второму способу. В этом случае базисный полином

$$F^3 = a_0 X_0 \oplus a_1 X_1 \oplus a_2 X_2 \oplus a_3 X_3,$$

а дополнительный полином

$$\lambda^3 = a_1 X_1 \oplus a_2 X_2 \oplus a_3 X_0.$$

Алгоритм кодирования

$$F_0 = F^3 + \lambda^3 = a_0 X_0^{(1-8)} \oplus a_1 X_1^{(1-12)} \oplus a_2 X_2^{(1-12)} \oplus a_3 \left[X_3^{(1-8)} + X_0^{(9-12)}\right],$$

а результирующие векторы $Y_0 = 111111110000$, $Y_1 = 010101010101$, $Y_2 = 001100110011$, $Y_3 = 000011111111$. Нетрудно заметить, что векторы Y_1 и Y_2 тождественны исходным опорным векторам X_1 и X_2, а векторы Y_0 и Y_3 отличаются от векторов X_0 и X_3 тем, что в них применено чередование структуры. Это обеспечивает при декодировании для каждого информационного символа не менее шести уравнений проверок и позволяет корректировать все одиночные и двойные ошибки, а также обнаруживать тройные.

Схема кодопреобразователя представлена на рисунке, где *ТГ* — тактовый генератор, *ТР* — триггер в счетном режиме, *И* — схема логического умножения, *С* — сумматор по модулю два, *УК* — управляющий каскад, *ВК* — включающий каскад, *ПК* — переключающий каскад.

ЛИТЕРАТУРА

1. Н. Н. Буга, И. М. Израйлит. Сумматор по модулю два. Авт. свидет. № 155045 от 25/III 1963. Бюллетень изобретений, 1963, 11.

2. Н. Н. Буга, И. М. Израйлит. Способ генерации помехоустойчивого двоичного группового кода. Авт. свидет. № 166168 от 29/III 1964. Бюллетень изобретений, 1964, № 21.

3. Н. Н. Буга, И. М. Израйлит. Устройство кодирования двоичной информации в самокорректирующийся код. Авт. свидет. № 166260 от 29/VIII 1964. Бюллетень изобретений, 1964, № 21.

4. У. Питерсон. Коды, исправляющие ошибки. Перев. с англ. под ред. Р. Л. Добрушина. Изд-во «Мир», 1964.