ОБ ОДНОМ СПОСОБЕ ДЕКОДИРОВАНИЯ ДВОИЧНЫХ ГРУППОВЫХ КОДОВ

Н. Н. Буга, И. М. Израйлит

Многие каналы связи, использующие помехоустойчивые коды, характеризуются сравнительно низкой скоростью телеграфирования — от десятков до нескольких тысяч бод. В то же время скорость работы импульсных устройств в счетных схемах кодопреобразователей достигает 10^6 *операций/сек*. Таким образом, имеется резерв времени, который может использоваться для осуществления динамического декодирования, повышающего эффективность групповых кодов.

Действие помех в канале связи на передаваемую кодограмму A_i можно представить в виде двоичного вектора помехи X. При этом принимаемая кодограмма будет представлять собой случайный вектор $Z_i = A_i \oplus X$, образованный посимвольным суммированием по модулю два векторов A_i и X.

Пусть декодирующее устройство содержит генератор исправляющих векторов B_i той же значности, что и кодограммы $\{A_i\}$. Множество $\{B_i\}$ представляет собой совокупность двоичных последовательностей, тождественных векторам ошибок, исправляемых данным кодом. Результирующая кодовая комбинация

$$N_i = Z_i \oplus B_i = A_i \oplus X \oplus B_i.$$

Очевидно, что $N_i = A_i$ только тогда, когда $B_i = X$.

Априорное знание вектора помехи на приемном конце системы исключается. Однако с помощью проверок на соответствие принятой кодограммы можно из всей совокупности исправляющих векторов выбрать один единственный, обеспечивающий правильное декодирование. Действительно, при выполнении проверок на соответствие получается $l = (n - k)$-значный корректор Q, где n — значность кода, k — число информационных символов. Введем функцию обнаружения

$$F = \begin{cases} 0 & \text{при } Q = 0; \\ 1 & \text{при } Q \neq 0. \end{cases} \qquad (1)$$

При смене исправляющих векторов Q и F дискретно изменяют свои значения. Максимально время обработки принятой кодограммы

$$T_0 = \Delta t 2^{n-k}, \qquad (2)$$

где Δt — временной интервал, необходимый для определения значения одного исправляющего вектора.

Функциональная схема устройства, работающего по принципу динамического декодирования, представлена на рис. 1. Генератор исправляющих векторов (ГИВ) формирует последовательно во времени векторы ошибок B_i до тех пор, пока на выходе решающего устройства (РУ) не появится сигнал, указывающий на отсутствие ошибки в кодограмме N_i. Этот сигнал останавливает ГИВ и выдает результат получателю информации (ПИ).

Для коррекции m-кратных ошибок хэммингово расстояние между различными кодограммами должно быть $d \geqslant 2m + 1$. Искажение помехой переданного вектора A_i с трансформацией его в вектор Z_i уменьшает расстояние d до одной из ближайших комбинаций. Однако его величина не может превышать значения m. Генератор ГИВ вырабатывает векторы с весом, также не превышающим m. При суммировании по модулю два векторов Z_i и B_i, когда вектор помехи X и исправляющий вектор B_i имеют максимальный вес (кроме векторов, имеющих единицы на одноименных позициях), расстояние $d = (2m + 1) - 2m = 1$.

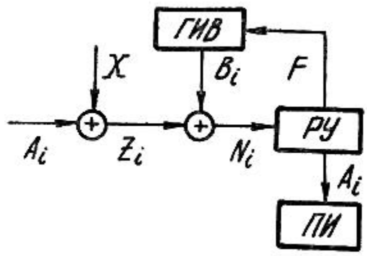

Рис. 1.

Этот наиболее неблагоприятный случай показывает, что даже при ошибке с максимальным весом расстояние d обеспечивает возможность обнаружения ошибки. При $X = B_i$ происходит взаимная компенсация векторов, и принятая кодограмма декодируется как неискаженная.

Проиллюстрируем принцип динамического декодирования на коде с хэмминговым расстоянием $d = 3$. Пусть A_i — кодовая комбинация, соответствующая i-у сообщению алфавита источника 2^k. Любая другая комбинация $A_{i \pm j}$, где $j \leqslant 2^k - 1$, отстоит от нее на расстоянии $d = 3$. Пусть помеха исказила кодограмму A_i так, что она трансформировалась в запрещенную кодограмму. Проверка на соответствие выявит наличие искажений $(F = 1)$, и будет включен ГИВ. Последний методом проб будет изменять положение импульсной последовательности в кодовом пространстве, перемещая ее по точкам запрещенных комбинаций, отстоящим на расстоянии $d > 1$, до тех пор, пока проверка на соответствие не укажет на отсутствие ошибки.

Операция		Разряды 1 2 3 4 5 6 7	Q	F
$Z_i \oplus$	B_1	1 0 1 0 0 1 0	101	1
	B_2	0 1 1 0 0 1 0	101	1
	B_3	0 0 0 0 0 1 0	011	1
	B_4	0 0 1 1 0 1 0	101	1
	B_5	0 0 1 0 1 1 0	000	0
	B_6	0 0 1 0 0 0 0	110	1
	B_7	0 0 1 0 0 1 1	010	1

Метод динамического декодирования исключает возможность трансформации одной кодограммы в другую, если веса векторов помехи и ошибки не превышают d.

Рассмотрим следующий пример. Пусть в канале применен (7,4) — код Хэмминга с расстоянием $d = 3$; очевидно, что алфавит источника может содержать 16 сообщений. Уравнения проверок на соответствие имеют вид $Z_1 = Z_3 \oplus Z_5 \oplus Z_7$; $Z_2 = Z_3 \oplus Z_6 \oplus Z_7$; $Z_4 = Z_5 \oplus Z_6 \oplus Z_7$. Пусть передается кодограмма $A_r = 0010110$, а вектор помехи $X = 0000100$. Тогда принятая кодограмма $Z_i = 0010010$, корректор $q_1 q_2 q_3 = 101$, а функция $F = 1$. Генератор ГИВ формирует исправляющие векторы с весом $w = 1$, например, в следующем порядке: $B_1 = 1000000$, $B_2 = 0100000$, $B_3 = 0010000$, $B_4 = 0001000$, $B_5 = 0000100$, $B_6 = 0000010$, $B_7 = 0000001$, которые поочередно посимвольно складываются по модулю два с принятой кодограммой Z_i, и производится проверка на соответствие. Возможные варианты сумм $Z_i \oplus B_i$ представлены в таблице, где измененный с помощью вектора B_i

кодовый символ подчеркнут. Анализ таблицы показывает, что искаженная кодограмма $Z_i = 0010010$ в процессе своего изменения по два раза приближается к кодограммам 1011010 и 0110011 на расстояние $d = 1$; дважды она оказывается удаленной от всех комбинаций алфавита на расстояние $d = 2$ и только один раз совпадает с переданной кодограммой. В последнем случае $F = 0$, что обусловливает однозначность декодирования.

Динамическое декодирование выгодно отличается от известных методов статического декодирования тем, что позволяет учитывать различные законы распределения ошибок в канале связи, изменяя программу работы ГИВ. Так, например, при биноминальном законе распределения ошибок ГИВ должен генерировать сначала все векторы с весом $w = 1$, затем с весом $w = 2$ и т. д. Если наиболее вероятными оказываются пакеты ошибок, то ГИВ перестраивается на формирование векторов ошибок в порядке убывания их наиболее вероятного веса.

Рассмотрим в качестве примера (11,7) — код Слепяна. Кроме коррекции ошибок при биноминальном законе распределения, когда код исправляет все 11 одиночных и 4 из 55 двойных ошибок (вариант I), его можно использовать при других законах распределения ошибок в канале связи. В частности, при преобладании парных (стоящих рядом) ошибок метод динамического декодирования обеспечивает коррекцию 8 из 10 парных ошибок, 2 двойных разнесенных и 5 одиночных ошибок (вариант II). Если же условия в канале таковы, что после одной ошибки искажение двух последующих символов маловероятно, то код можно использовать для исправления только двойных ошибок (вариант III). Во всех рассмотренных случаях используется полный объем корректора, допускающий исправление 15 ошибок. Бо́льших возможностей при (11,7) — коде достигнуть невозможно. Заметим, что несмотря на различные исправляющие векторы при рассмотренных трех вариантах декодирования, алгоритм кодирования остается неизменным и определяется образующими векторами $Y_1 = 10000001111$, $Y_2 = 01000000111$, $Y_3 = 00100001011$, $Y_4 = 00010001101$, $Y_5 = 00001001110$, $Y_6 = 00000100011$, $Y_7 = 00000010101$.

Рассмотрим возможность построения универсального алгоритма декодирования, позволяющего осуществлять обработку любого группового кода в одном кодопреобразователе. Естественно, что должна быть обеспечена возможность перестройки программы в соответствии с заданием.

Выше было показано, что при динамическом декодировании корректор Q получается согласно $l = (n - k)$ проверкам на соответствие и реализуется посредством функции обнаружения F, учитывающей наличие или отсутствие ошибок в принятой кодограмме. Таким образом, изменение алгоритма декодирования при различных кодах обеспечивается суммированием по модулю два различных сочетаний символов принятой кодограммы.

Корректор для (n, k)-кода представляет собой $(n - k)$-мерный вектор, каждая реализация которого является функцией принятой кодограммы

$$Q = f(Z_1, Z_2, \ldots, Z_n),$$

где $Z_i = 0$ или 1. Так как групповые коды линейные, то

$$Q_j = \bigoplus \sum_{i=1}^{n} \psi_{ji} Z_i,$$

где $\psi_{ji} = 0$ или 1 — коэффициент, характеризующий зависимость j-й компоненты корректора от i-го символа принятой кодограммы.

Составим матрицу для всех возможных значений корректора

$$M_Q = \begin{pmatrix} \psi_{11}Z_1 & \ldots & \psi_{1n}Z_n \\ \cdot & \cdot\ \cdot\ \cdot & \cdot \\ \psi_{n-k,\,1}Z_1 & \ldots & \psi_{n-k,\,n}Z_n \end{pmatrix},$$

где для каждого разряда корректора в соответствующей j-й строке производится суммирование по модулю два посимвольных произведений компонент ψ_{ji} и Z_i. Так как совокупность символов $\{Z_i\}$ составляет принятую кодограмму Z_i, то совокупность элементов $\{\psi_{ji}\}$ для каждой строки матрицы может быть представлена в виде управляющего вектора $C_j = (\psi_{j1}, \psi_{j2}, \ldots, \psi_{jn})$. Таким образом, принятая кодограмма посимвольно умножается на векторы C_j всех строк. Символы результирующей двоичной последовательности суммируются по модулю два, образуя реализацию корректора. Совокупность компонент корректора образует $(n-k)$-значное двоичное число, преобразуемое затем в одноразрядную двоичную функцию обнаружения F.

Управляющие векторы C_j для каждого частного алгоритма кода формируются по следующим правилам:

выявляется соответствие между компонентами корректора и контрольными позициями кодограммы ($Q_j \leftrightarrow Z_j$);

уравнения проверок на соответствие представляются в виде функциональной зависимости от информационных символов

$$Q_j = f(Z_1, Z_2, \ldots, Z_n) = f(\alpha_1, \alpha_2, \ldots, \alpha_k);$$

для каждого значения Z_i в матрицу M_Q вписывается значение

$$\psi_{ji} = \begin{cases} 1 \text{ — при наличии зависимости от соответствующего разряда } Z_i; \\ 0 \text{ — при отсутствии такой зависимости.} \end{cases}$$

Рассмотрим примеры построения управляющих векторов.

(12,8) — код Слепяна. Уравнения проверок кода имеют вид [3]

$$\alpha_1 = Z_1; \quad \alpha_2 = Z_2; \quad \alpha_3 = Z_3; \quad \alpha_4 = Z_4; \quad \alpha_5 = Z_5;$$
$$\alpha_6 = Z_6; \quad \alpha_7 = Z_7; \quad \alpha_8 = Z_8;$$
$$Z_9 = Z_1 \oplus Z_2 \oplus Z_3 \oplus Z_5 \oplus Z_6 \oplus Z_7 \oplus Z_8;$$
$$Z_{10} = Z_1 \oplus Z_2 \oplus Z_3 \oplus Z_4 \oplus Z_6;$$
$$Z_{11} = Z_1 \oplus Z_2 \oplus Z_4 \oplus Z_5 \oplus Z_7;$$
$$Z_{12} = Z_1 \oplus Z_3 \oplus Z_4 \oplus Z_5 \oplus Z_8;$$

Управляющие векторы, построенные по описанной методике, следующие: $C_1 = 111011111000$, $C_2 = 111101000100$, $C_3 = 110110100010$, $C_4 = 101110010001$.

(12,8) — код Хэмминга. Уравнения проверок кода имеют вид [3]

$$\alpha_1 = Z_3; \quad \alpha_2 = Z_5; \quad \alpha_3 = Z_6; \quad \alpha_4 = Z_7; \quad \alpha_5 = Z_9;$$
$$\alpha_6 = Z_{10}; \quad \alpha_7 = Z_{11}; \quad \alpha_8 = Z_{12};$$
$$Z_1 = Z_3 \oplus Z_5 \oplus Z_7 \oplus Z_9 \oplus Z_{11};$$
$$Z_2 = Z_3 \oplus Z_6 \oplus Z_7 \oplus Z_{10} \oplus Z_{11};$$
$$Z_4 = Z_5 \oplus Z_6 \oplus Z_7 \oplus Z_{12};$$
$$Z_8 = Z_9 \oplus Z_{10} \oplus Z_{11} \oplus Z_{12}.$$

Управляющие векторы будут: $C_1 = 101010101010$, $C_2 = 011001100110$, $C_3 = 000111100001$, $C_4 = 000000011111$.

Для других кодов управляющие векторы строятся аналогичным образом.

ЛИТЕРАТУРА

1. Н. Н. Буга, И. М. Израйлит. Устройство кодирования двоичной информации в самокорректирующийся групповой код, авт. свидетельство № 166260 от 29. 08. 1964.

2. Н. Н. Буга, И. М. Израйлит. Устройство динамического декодирования групповых кодов. Описание к авторскому свидетельству с приоритетом от 9. 02. 1965.

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