Лекция 4
Блоковые коды

Помехоустойчивые коды можно разделить на блоковые и древовидные.

При построении **блоковых кодов** передаваемая информационная последовательность разделяется на фрагменты, содержащие определенное количество символов. Из этих информационных и некоторого количества рассчитанных дополнительных (проверочных) символов формируются кодовые слова.

В древовидных кодах информационная последовательность не разделяется на части, а обрабатывается непрерывно по мере поступления. К классу древовидных относятся **сверточные коды**, рассмотренные далее.

Если в блоковом коде количество символов во всех кодовых словах одинаково, то такой код называется **равномерным**. В равномерных блоковых кодах из информационных символов путем выполнения некоторых математических операций формируются проверочные символы, которые, наряду с информационными, содержатся в каждом кодовом слове.

Если информационные символы расположены в начале каждого кодового слова, а проверочные – в конце, и при кодировании значения информационных символов не изменяются, то такие коды называются **систематическими**.

Для равномерных блоковых кодов применяется обозначение (n,k), где n - общее количество символов в блоке, k – количество информационных символов в блоке. Иногда в обозначении таких кодов указывается минимальное кодовое расстояние данного кода $(n,k,d_{мин})$. Количество проверочных символов r определяется из выражения

$$r = n - k . \qquad (7.1)$$

Для блоковых равномерных кодов существует понятие **скорость кода** $\boldsymbol{R}$, которая определяется по формуле

$$R = k/n. \qquad (7.2)$$

Величина R в помехоустойчивом коде всегда меньше единицы, так как при R=1 k=n, все символы в блоке – информационные, а проверочные отсутствуют т. е. $r = n–k$ =0.

Избыточность кода $\boldsymbol{R_{изб}}$ можно определить по формуле

$$\boldsymbol{R_{изб} = r/n} . \qquad (7.3)$$

Рассмотрим систематический код (7,4). Из этого обозначения следует, что общее количество символов в блоке n=7, из них информационных символов k =4, а проверочных – $r = n - k = 3$. Скорость кода в данном случае равна $R = k/n = 4/7$, а избыточность $R_{изб} = r/n = 3/7$. Минимальное кодовое расстояние $d_{мин} = 3$.

Обозначим двоичные информационные символы в блоке как a_1,a_2,a_3,a_4, а проверочные - b_1 ,b_2 ,b_3 .

Проверочные символы можно определить из информационных, используя, например, следующую систему уравнений

$$\begin{cases} a_1 + a_2 + a_3 + b_1 = 0 \\ a_2 + a_3 + a_4 + b_2 = 0 \\ a_1 + a_2 + a_4 + b_3 = 0 \end{cases} \qquad (7.4)$$

Поскольку все символы в этих уравнениях - двоичные, сложение производится по модулю числа 2 (обозначается mod 2). Для двоичных чисел сложение по модулю 2 эквивалентно выполнению логической операции «исключающее ИЛИ», при которой

0 + 0 = 0; 1 + 0 = 1; 0 + 1 = 1; 1 + 1 = 0.

Все четные числа по модулю 2 равны нулю: (0,2,4,6,……) mod 2 = 0, а все нечетные - единице: (1,3,5,7,……) mod 2 = 1, поскольку остатки от деления чисел на 2 в первом случае равны нулю, а во втором – единице.

Существует также понятие сравнимости по модулю заданного числа. При этом сравниваемые числа делятся на число, по модулю которого осуществляется сравнение, и если остатки от деления (так называемые вычеты по модулю заданного числа) равны, то такие числа называются сравнимыми по модулю заданного числа (обозначается знаком ≡). Например 11 ≡ 25 (mod 7), поскольку при делении обоих чисел на 7 остатки равны 4.

Учитывая, что при выполнении операций по модулю 2, 1 = -1, уравнения (7.4) можно переписать в виде

$$\begin{cases} b_1 = a_1 + a_2 + a_3 \\ b_2 = a_2 + a_3 + a_4 \\ b_3 = a_1 + a_2 + a_4 \end{cases} \qquad (7.5)$$

.

Допустим, что $a_1 = a_2 = 1$, $a_3 = a_4 = 0$. Тогда из уравнений (7.5)

$b_1 = 0$, $b_2 = 1$, $b_3 = 0$. Таким же образом можно определить значения проверочных символов для любого набора информационных.

Количество информационных символов k=4. При этом количество различных наборов информационных символов равно $2^k=2^4=16$. Поскольку каждый набор информационных символов задает свои значения проверочных символов, количество кодовых слов $N_к$ в данном коде

$$N_к=2^k=2^4=16.$$

Общее количество различных наборов $N_{общ}$ которые можно образовать из n = 7 двоичных символов равно

$$N_{общ} =2^n = 2^7 = 128.$$

Отсюда количество наборов, не являющихся кодовыми словами

$$N_{нк} = N_{общ} - N_к = 112.$$

Обнаружение и исправление ошибок на приемной стороне с помощью данного кода можно осуществлять путем проверки выполнения уравнений 7.4).

Если все три уравнения справедливы, то можно сделать вывод о том, что принятое кодовое слово не содержит ошибок (или количество ошибок большее, чем может обнаружить данный код). Допустим, что не выполняются все три уравнения. При наличии одной ошибки это может произойти только в том случае, если ошибочным является символ a_2, который входит во все три уравнения. Если не выполняются первые два уравнения, то одиночная ошибка содержится в символе a_3 , входящим в оба эти уравнения. Аналогичным образом, если не выполняются первое и третье уравнения, то ошибочен символ a_1, а если второе и третье – то символ a_4 . Одиночные ошибки в проверочных символах b_1 , b_2 , b_3 приводят к невыполнению только одного из уравнений (5.4), в которое входит соответствующий символ.

В теории помехоустойчивого кодирования блоковым кодам приписываются свойства линейного векторного пространства, и каждое кодовое слово рассматривается как кодовый вектор, принадлежащий этому пространству.

Для того чтобы однозначно определить групповой код достаточно задать k линейно независимых уравнений. Из этих уравнений (базисных векторов) можно образовать все остальные кодовые слова (кодовые векторы).

Под термином «линейно независимые» понимается следующее.

Пусть V - векторное пространство и S - произвольное множество векторов из V. Непустое множество $S=\{e_1,e_2,\dots,e_j\}$ попарно различных векторов векторного пространства V линейно зависимо, если существуют действительные числа $\lambda_1,\lambda_2,\dots,\lambda_j$, не все равные нулю, такие, что

$$\lambda_1 e_1+\lambda_2 e_2+\dots+\lambda_j e_j=0.$$

Если, напротив, это соотношение имеет место только при

$$\lambda_1=\lambda_2=\dots=\lambda_j=0 \quad ,$$

то множество векторов S - линейно независимо.

Для того, чтобы определить рассмотренный выше код (7,4) достаточно задать $k = 4$ линейно независимых уравнения. Это можно сделать, в частности, если в каждом из четырех уравнений, не повторяясь, считать один из информационных символов равным единице, а остальные - нулю. Значения проверочных символов следует определять из уравнений (7.5). Тогда получим систему уравнений

$$\begin{cases} a_1+0+0+0+b_1+0+b_3=1 \\ 0+a_2+0+0+b_1+b_2+b_3=0 \\ 0+0+a_3+0+b_1+b_2+0=1 \\ 0+0+0+a_4+0+b_2+b_3=1 \end{cases} \qquad (7.6)$$

Справа от знака равенства указан результат сложения ненулевых символов по модулю 2.

Из символов, входящих в уравнения (7.6), можно образовать матрицу следующего вида.

$$G=\begin{bmatrix} 1000101 \\ 0100111 \\ 0010110 \\ 0001011 \end{bmatrix} \qquad (7.7)$$

Матрица G называется порождающей. С её помощью можно образовать все 16 кодовых слов данного кода. Выполняется это следующим образом. Во-первых, все четыре строки порождающей матрицы являются кодовыми словами. Кроме того, строки можно складывать посимвольно по модулю 2 в любых сочетаниях: по две, по три, по четыре, получая каждый раз кодовые слова. Произведя однократное суммирование любой строки с собой же, можно получить нулевое кодовое слово. При сложении всех четырех строк получится кодовое слово 1111111.

Для группового кода можно образовать так называемую проверочную матрицу. Если переписать уравнения (7.4) таким образом, что в каждом уравнении вместо отсутствующих символов, относящихся к a_1, a_2, a_3, a_4, b_1, b_2, b_3, поставить нули, то получим

$$\begin{cases} a_1 + a_2 + a_3 + 0 + b_1 + 0 + 0 = 0 \\ 0 + a_2 + a_3 + a_4 + 0 + b_2 + 0 = 0 \\ a_1 + a_2 + 0 + a_4 + 0 + 0 + b_3 = 0 \end{cases} \qquad (7.8)$$

Этим уравнениям соответствует матрица *H*, называемая проверочной

$$H = \begin{bmatrix} 1110100 \\ 0111010 \\ 1101001 \end{bmatrix} \qquad (7.9)$$

.

Произведение проверочной матрицы *H* на кодовое слово a , не содержащее ошибок, равно нулю, то есть

$$H \times a = 0 \qquad (7.10)$$

Перемножение матриц имеет смысл только в том случае, если количество столбцов первой матрицы M1 равно количеству строк второй матрицы M2. Такие матрицы называются сцепленными. Результирующая матрица M3 будет иметь количество строк такое же, как у первой матрицы M1, а количество столбцов - как у второй M2.

Для того, чтобы можно было осуществлять перемножение проверочной матрицы *H* на матрицу, соответствующую кодовому вектору, необходимо произвести транспонирование либо первой либо второй матрицы. При транспонировании строки и столбцы матрицы меняются местами (будем обозначать транспонированную матрицу индексом *T*).

Для примера перемножим проверочную матрицу *H* на кодовое слово, образованное сложением первых двух строк порождающей матрицы *G* - 1100010. Транспонируем матрицу, соответствующую этому кодовому слову. В результате получим так называемый вектор-столбец:

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Произведем перемножение

$$H \times a_T = \begin{bmatrix} 1110100 \\ 0111010 \\ 1101001 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+1+0+0+0+0+0 \\ 0+1+0+0+0+1+0 \\ 1+1+0+0+0+0+0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \bmod 2$$

В результате перемножения получили матрицу, содержащую только нулевые элементы.

Введем нарочно ошибку в кодовое слово. Для этого заменим, например, значение третьего от начала символа с 0 на 1 и произведем перемножение матриц

$$H \times a_T = \begin{bmatrix} 1110100 \\ 0111010 \\ 1101001 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+1+1+0+0+0+0 \\ 0+1+1+0+0+1+0 \\ 1+1+0+0+0+0+0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \bmod 2$$

Результирующая матрица содержит два ненулевых элемента, что указывает на наличие ошибки.

По расположению ненулевых элементов в результирующей матрице можно определить ошибочный символ. При наличии одиночной ошибки неравенство нулю первых двух элементов результирующей матрицы эквивалентно невыполнению первого и второго уравнений из системы (7.6). Это возможно при наличии ошибки в информационном символе a_3 (третий символ от начала кодового слова), который входит в оба эти уравнения. Если в результирующей матрице все три элемента не равны нулю, то не выполняются все три уравнения и ошибочен символ a_2 и т. д. аналогично изложенному выше.