5. Линейные блочные коды и коды Хэмминга

Линейные блочные коды позволяют представить информационные и кодовые слова в виде двоичных векторов, что позволяет описать процессы кодирования и декодирования с помощью аппарата линейной алгебры, с учетом того, что компонентами вводимых векторов и матриц являются символы «0» и «1». Операции над двоичными компонентами производятся при этом по правилам арифметики по модулю 2 [**?**].

Множество 2^k возможных двоичных информационных слов блокового (n,k)-кода взаимно однозначно отображается в множество 2^k кодовых слов длиной n.

Далее рассмотрим механизм исправления и обнаружения ошибок в помехоустойчивом кодировании. Для этого удобно рассмотреть множество двоичных слов (векторов) длиной n в виде точек на плоскости (см. рис. 5.1). Ранее были рассмотрены такие важные параметры, как кодовое расстояние d и минимальное кодовое расстояние $d_{\min}$, а также связанный с минимальным кодовым расстоянием параметр гарантированно исправляемой кратности ошибки t. На рис. 5.1 черными кругами показаны два кодовых слова c_1 и c_2, отличающиеся друг от друга в $d_{\min}$двоичных символов. Вокруг них показаны области, содержащие слова длиной n, отличающиеся от этих кодовых слов не более чем в t позициях. Прочие кодовые слова показаны черными ромбами.

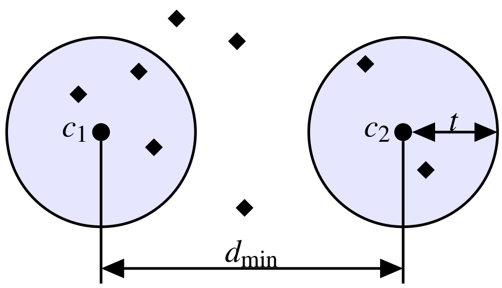

Рис. 5.1. Общий принцип исправления и обнаружения ошибок

В случае, если по каналу было передано кодовое слово c_1, и оно пришло с искажениями, возможны три варианта декодирования с исправлением ошибки.

1. Было получено слово, попадающее в область вокруг вектора c_1. Такое слово будет преобразовано декодером в слово c_1 и декодирование будет осуществлено верно.
2. Если получено слово, не принадлежащее областям ни одного кодового слова, то оно не может быть декодировано и, следовательно, возникает ошибка декодирования.
3. Слово, попадающее в область вокруг c_2, будет преобразовано декодером в c_2. Такая ошибка не может быть обнаружена.

В показанном на рис. 5.1 случае не все слова размерности n принадлежат областям декодирования. Таких кодов большинство. Коды, в которых непересекающиеся области декодирования охватывают все пространство слов размерности n, называются *совершенными* или *плотноупакованными*. При использовании совершенных кодов всегда возможна коррекция ошибок (не всегда правильная). Декодер такого кода не может определить ошибку декодирования. Он работает либо в режиме определения ошибок, либо в режиме исправления ошибок. Основными совершенными кодами являются *коды Хэмминга* и *коды Голея* [**?**].

Код Хэмминга можно построить для любого натурального числа $r \geq 3$. Этот код будет обладать рядом свойств [**?**].

- $n = 2^r - 1$
- $k = 2^r - 1 - r$
- $r = n - k$
- $d_{\min} = 3, t = 1$

Далее будем рассматривать процессы кодирования и декодирования линейных блочных кодов на примере кодов Хэмминга.

5.1. Кодирование линейных блочных кодов

Поскольку между информационными и кодовыми словами существует взаимно однозначное соответствие, процесс кодирования может быть осуществлен с использованием таблицы соответствий, хранящейся в памяти кодера. Однако, для длинных кодов такой метод неприемлем, так как требует большой объем памяти для хранения таблицы.

Вместо этого вводится понятие так называемой *порождающей матрицы* G. Оно основано на том, что подпространство всех кодовых слов линейного блочного (n,k)-кода имеет некоторый *базис* $(v_0, v_1, \ldots, v_{k-1})$, через который может быть выражено любое кодовое слово этого кода [**?**].

$$v = u_0 v_0 + u_1 v_1 + \ldots + u_{k-1} v_{k-1}, \tag{34}$$

где $u_i \in \{0,1\},\ \ 0 \leq i < k$.

Векторы базиса образуют порождающую матрицу G размера $k \times n$

$$G = \begin{bmatrix} v_0 \\ v_1 \\ \ldots \\ v_k - 1 \end{bmatrix} = \begin{bmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,n-1} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,n-1} \\ \ldots & \ldots & \ldots & \ldots \\ v_{k-1,0} & v_{k-1,1} & \cdots & v_{k-1,n-1} \end{bmatrix}. \tag{35}$$

Тогда уравнение (34) принимает вид

$$v = uG, \tag{36}$$

где $u = (u_0, u_1, \ldots, u_{k-1})$ — информационное слово [?].

Фактически, формула (36) описывает процедуру кодирования линейного блочного кода посредством образующей матрицы.

Для пространства кодовых слов линейного (n,k)-кода существует *дуальное* ему пространство кода $(n, n-k)$, порождаемое матрицей H размера $(n-k) \times n$. Такая матрица получила название *проверочной* для кода (n,k) и обладает следующими свойствами

$$\begin{aligned} GH^T &= 0, \\ vH^T &= 0, \end{aligned} \tag{37}$$

на основе которых реализована операция декодирования линейных блочных кодов [?].

Как правило рассматривают так называемые *систематические* или *каноничные* формы матриц G и H, использующиеся для процедуры *систематического* кодирования. На практике, любая порождающая матрица G линейного блочного (n,k)-кода может быть преобразована к систематическому виду посредством элементарных операций и перестановок столбцов матрицы [?, ?].

Матрица G в систематической форме состоит из двух подматриц: единичной матрицы I_k размера $k \times k$ и проверочной подматрицы P размера $k \times (n-k)$ [?].

$$G_{k \times n} = (P_{k \times (n-k)} | I_k). \tag{38}$$

Соответственно, исходя из свойства (37), следует, что проверочная матрица H состоит из единичной матрицы I_{n-k} и транспонированной проверочной подматрицы P [?].

$$H_{(n-k) \times n} = (I_{n-k} | P^T_{k \times (n-k)}). \tag{39}$$

В качестве примера приведем порождающую (40) и проверочную (41) матрицы для кода Хэмминга $(7,4)$.

$$G_{(7,4)} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{40}$$

$$H_{(7,4)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{41}$$

Для примера также рассмотрим процедуру кодирования с использованием порождающей матрицы G (40). В качестве информационного слова возьмем вектор $u = [1\ 0\ 1\ 1]$.

$$v = u \cdot G_{(7,4)} = [1\ 0\ 1\ 1] \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = [1\ 0\ 0\ 1\ 0\ 1\ 1]. \tag{42}$$

5.2. Декодирование линейных блочных кодов

Как и в случае кодирования, декодирование линейных блочных кодов можно осуществлять посредством таблицы по принципу максимального правдоподобия. В этом случае производится последовательное поразрядное сравнение принятого на вход декодера слова со всеми возможными кодовыми словами. В результате будет выбрано кодовое слово, имеющее наименьшее число отличий от декодируемого. В случае несовершенных кодов возможен вариант, когда есть несколько кодовых слов, отличающихся от принятого в одинаковом числе разрядов. Соответственно, декодер не может принять решение о верности одного из вариантов и выдает сигнал о невозможности декодирования. Недостатки такой схемы те же, что и в случае кодирования — необходим большой объем памяти для хранения всех кодовых слов в случае длинных кодов. Быстродействие для длинных кодов также значительно увеличивается.

В связи с этим используют механизм *синдромного декодирования*, основанный на использовании проверочной матрицы H.

Для понимания принципа декодирования рассмотрим как выражаются проверочные символы кодового слова через информационные на примере систематического кода Хэмминга $(7,4)$.

$$\begin{aligned} v_0 &= v_3 \oplus v_5 \oplus v_6; \\ v_1 &= v_3 \oplus v_4 \oplus v_5; \\ v_2 &= v_4 \oplus v_5 \oplus v_6. \end{aligned}$$

Если в канале произошла ошибка, то для принятого вектора r хотя бы одно из равенств выполняться не будет. Эти проверочные соотношения можно записать для принятого вектора в виде системы уравнений (43).

$$\begin{aligned} r_0 \oplus r_3 \oplus r_5 \oplus r_6 &= s_0; \\ r_1 \oplus r_3 \oplus r_4 \oplus r_5 &= s_1; \\ r_2 \oplus r_4 \oplus r_5 \oplus r_6 &= s_2. \end{aligned} \tag{43}$$

Соответственно, если хотя бы один из компонент вектора $s = \{s_0, s_1, s_2\}$ не равен нулю, то в принятом слове есть ошибка [**?**].

Уравнения (43) можно записать через проверочную матрицу H.

$$s = r \odot H^T. \tag{44}$$

Вектор s принято называть *синдромом*. Таким образом, ошибка в принятом слове будет обнаружена, если хоть один компонент синдрома принятого слова не равен нулю [**?**].

Для исправления ошибки используется тот факт, что каждый синдром соответствует своей позиции одиночной ошибки (мы говорим о кодах Хэмминга). Таким образом, перебрав все возможные варианты одиночной ошибки можно получить таблицу соответствия *синдром-ошибка*. В табл. 5.1 приведены соответствия позиций ошибки и синдромов для кода Хэмминга $(7,4)$ [**?**].

Таблица 5.1

Таблица соответствия синдром-ошибка для кода Хэмминга $(7,4)$

Позиция ошибки	r_0	r_1	r_2	r_3	r_4	r_5	r_6
Синдром	$[1\ 0\ 0]$	$[0\ 1\ 0]$	$[0\ 0\ 1]$	$[1\ 1\ 0]$	$[0\ 1\ 1]$	$[1\ 1\ 1]$	$[1\ 0\ 1]$

Если сравнить табл. 5.1 и проверочную матрицу (41), то можно увидеть, что ошибке в i-й позиции кодового слова соответствует синдром, образованный i-м столбцом матрицы H [**?**].

Для примера рассмотрим декодирование полученной ранее кодовой комбинации $v = [1\ 0\ 0\ 1\ 0\ 1\ 1]$ без ошибок и с ошибкой в позиции v_4.

При отсутствии ошибки синдром будет равен

$$s = v \odot H^T = [1\ 0\ 0\ 1\ 0\ 1\ 1] \odot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = [0\ 0\ 0],$$

что доказывает отсутствие ошибки

Если наложить на вектор v ошибку в позиции v_4 будет получен вектор $r = [1\ 0\ 0\ 1\ 1\ 1\ 1]$. Теперь синдром будет равен

$$s = r \odot H^T = [1\ 0\ 0\ 1\ 1\ 1\ 1] \odot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = [0\ 1\ 1],$$

что, во-первых, показывает наличие ошибки, а во-вторых, согласно табл. 5.1, указывает, что она произошла в позиции r_4. Таким образом, ошибка может быть исправлена.

5.3. Расширенные коды Хэмминга

Расширение кода Хэмминга заключается в дополнении кодового слова дополнительным двоичным разрядом так, чтобы оно содержало четное число единиц. Такое расширение дает ряд преимуществ [**?**].

1. Длина кода увеличивается до $n = 2^r$, что удобнее для хранения и передачи информации.
2. Минимальное расстояние $d_{\min} = 4$, следовательно $t_{\text{обн}} = 3$.

Также, дополнительный разряд позволяет использовать декодер в гибридном режиме обнаружения и коррекции ошибок.

Для примера рассмотрим расширение кода Хэмминга $(7,4)$ — расширенный код Хэмминга $(8,4)$. Кодовый вектор

$$\tilde{v} = (\tilde{v}_0, \tilde{v}_1, \ldots, \tilde{v}_7)$$

расширенного кода $(8,4)$ получается из вектора

$$v = (v_0, v_1, \ldots, v_6)$$

кода $(7,4)$ путем добавления разряда проверки на четность, то есть

$$\tilde{v} = (\tilde{v}_0, \tilde{v}_1, \ldots, \tilde{v}_7) = (\tilde{v}_0, v_0, v_1, \ldots, v_6),$$

где

$$\tilde{v}_0 = \sum_{i=0}^{6} v_i.$$

Проверочная матрица кода $(8,4)$ получается из проверочной матрицы кода $(7,4)$ в два приема [**?**].

1. Слева к матрице $H_{(7,4)}$ дописывается нулевой столбец.
2. Полученная матрица дополняется сверху строкой из единиц.

$$H_{(8,4)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{45}$$

При синдромном декодировании

$$\tilde{s} = \tilde{v} \odot H_{(8,4)}^T \tag{46}$$

вектор синдрома имеет вид

$$\tilde{s} = (\tilde{s}_0, s_0, s_1, s_2) = (\tilde{s}_0, s), \tag{47}$$

где компонента $\tilde{s}_0$ равна сумме всех элементов кодового слова $\tilde{v}$ и, следовательно, равна нулю.

Далее рассмотрим процесс коррекции и обнаружения ошибок.

Процедура исправления одиночных ошибок совпадает с таковой для обычных кодов Хэмминга. Компонента $\tilde{s}_0$ при этом всегда равна единице, а синдром s соответствует синдрому обычного кода Хэмминга. Если же ошибка в дополнительном разряде $\tilde{v}_0$, то $\tilde{s}_0$ будет равно 1, а $s = (000)$. При двукратной же ошибке компонента $\tilde{s}_0$ всегда будет равна нулю. Таким образом можно представить гибридный алгоритм коррекции ошибок.

1. Если $\tilde{s}_0 = 1$, то исправление одиночной ошибки.
2. Если $\tilde{s}_0 = 0$ и $s \neq 0$, то обнаружена неисправляемая ошибка.