7. ЛИНЕЙНЫЕ БЛОКОВЫЕ КОДЫ

7. ЛИНЕЙНЫЕ БЛОКОВЫЕ КОДЫ

7.1. БАЗОВЫЕ ОПРЕДЕЛЕНИЯ

Все канальные коды могут быть разделены на два класса: блоковые коды (block codes) и последовательные / свёрточные коды (convolutional codes).

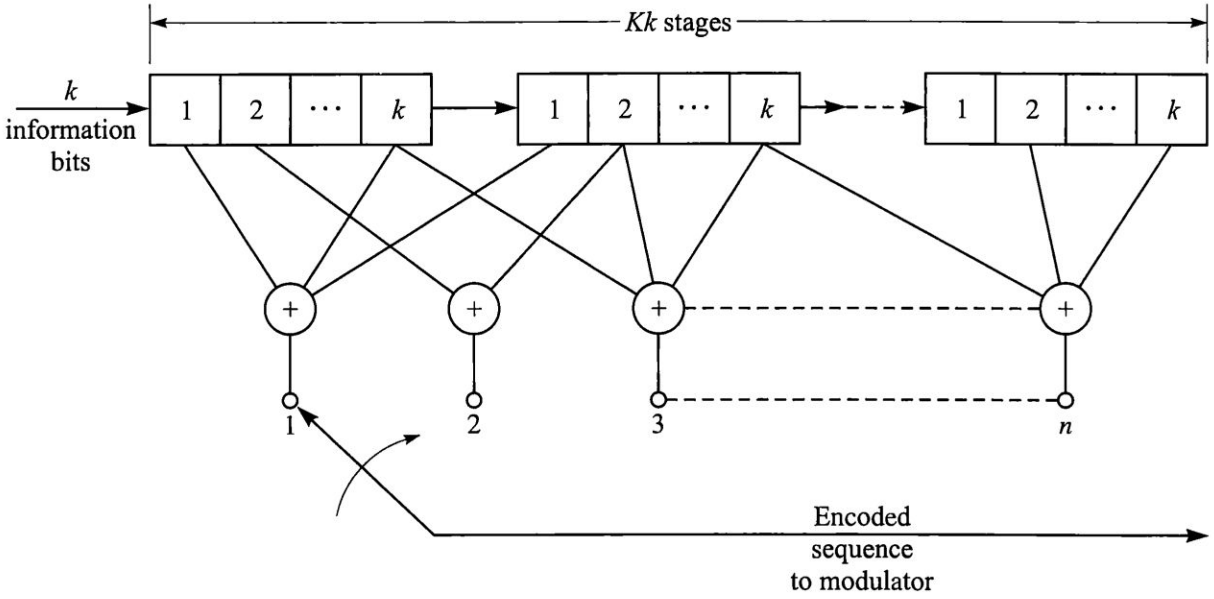
Блочный код выполняет отображение каждого информационного сообщения / блока (information sequence) длиной k бит (всего возможны $M = 2^k$ сообщений) в кодовое слово (codeword) длиной $n > k$ бит.

В блочных кодах нет памяти в том смысле, что результат кодирования текущего информационного блока не зависит от результатов кодирования предыдущих информационных блоков.

7.1. БАЗОВЫЕ ОПРЕДЕЛЕНИЯ

Свёрточные коды определяются с помощью конечных автоматов (finite-state machines). Для этих кодов в каждый i-ый момент времени на вход поступают k информационных бит, а на выходе генерируются n кодовых бит и при этом состояние кодера меняется с σ_{i-1} на σ_i. Количество состояния финитно и равно Σ. Таким образом, значения n выходных кодовых бит и номер нового состояния σ_i зависят от k входных информационных бит и номера текущего состояния σ_{i-1}.

Работу кодера свёрточного кода удобно описывать с помощью сдвигового регистра (shift register) длиной Kk. Состояние кодера определяется (правыми) $(K-1)k$ битами.



7.1. БАЗОВЫЕ ОПРЕДЕЛЕНИЯ

Кодовая скорость (code rate) как блочного, так и свёрточного кодов определяется отношением

$$R_c = \frac{k}{n}$$

Предположим, что кодовое слово длиной n передаётся с помощью N-мерного сигнального созвездия размером M, где M – целая степень 2 и $L = n/\log_2 M$ – количество M-ичных символов, требуемых для передачи кодового слова – также целое.

Если длина тактового интервала T_s, то получается, что для передачи k информационных бит требуется период времени $T = LT_s$ и скорость передачи информации равна

$$R = \frac{k}{LT_s} = \frac{k}{n}\frac{\log_2 M}{T_s} = R_c \frac{\log_2 M}{T_s} \text{ бит/с}$$

Rem: $N = \lfloor 2WT + 1 \rfloor \approx 2WT$

Размерность сигнала равна LN и, согласно теореме о размерности, минимальная полоса

$$W = \frac{N}{2T_s} = \frac{RN}{2R_c \log_2 M} \text{ Гц}$$

Спектральная эффективность

$$r = \frac{R}{W} = \frac{2\log_2 M}{N} R_c$$

$$W = \frac{RN}{2R_c \log_2 M}, \; r = \frac{2\log_2 M}{N} R_c$$

Следовательно:

$$\text{BPSK(Double-Sideband)}: \begin{cases} W = R / R_c \\ r = R_c \end{cases} \quad \text{BFSK}: \begin{cases} W = R / R_c \\ r = R_c \end{cases} \quad \text{QPSK}: \begin{cases} W = R / 2R_c \\ r = 2R_c \end{cases}$$

Пусть средняя энергия сигнального созвездия E_{av}, тогда энергия кодового слова

$$E = LE_{av} = \frac{n}{\log_2 M} E_{av}$$

и энергия на элемент (component) кодового слова

$$E_c = \frac{E}{n} = \frac{E_{av}}{\log_2 M}$$

Энергия на информационный бит:

$$E_b = \frac{E}{k} = \frac{E_{av}}{R_c \log_2 M}$$

Таким образом,

$$E_c = R_c E_b$$

Средняя мощность сигнала

$$P = \frac{E}{LT_s} = \frac{E_{av}}{T_s} = R \frac{E_{av}}{R_c \log_2 M} = RE_b$$

7. ЛИНЕЙНЫЕ БЛОКОВЫЕ КОДЫ

7.2. ОСНОВНЫЕ СВОЙСТВА ЛИНЕЙНЫХ БЛОКОВЫХ КОДОВ

Блоковый код □ называется q-ичным, если символы (элементы) его кодовых слов

$$\mathbf{c}_m = (c_{m1}, c_{m2}, ..., c_{mn}), \quad 1 \le m \le M$$

выбираются из q-ичного алфавита.

Заметим, что если $q = 2^b$ и b – натуральное, то каждый q-ичный символ может быть заменён двоичной последовательностью длиной b, следовательно, недвоичный код с длиной кодового блока N может быть представлен двоичным кодом с длиной кодового блока $n = bN$.

Если для двоичного кода из всех возможных 2^n кодовых блоков длиной n выбираются $M = 2k$ $(k < n)$ разрешённых кодовых блоков, то кратко код обозначается (n, k), его кодовая скорость $R_c = k/n$.

7.2. ОСНОВНЫЕ СВОЙСТВА ЛИНЕЙНЫХ БЛОКОВЫХ КОДОВ

Помимо кодовой скорости важным параметром для кодового слова является его вес (weight) – количество ненулевых элементов. В общем, слова могут иметь разный вес, образуя распределение весов кодовых слов (weight distribution). Если все веса имеют одинаковый вес, код называется кодом с фиксированным/постоянным весом (fixed-weight/constant-weight code).

Блоковый код □ (n, k) (k-мерное подпространство n-мерного пространства) называется линейным, если для любых двух кодовых слов $\mathbf{c}_1, \mathbf{c}_2$ □ □ их сумма также является кодовым словом $\mathbf{c}_1 + \mathbf{c}_2$ □ □. Таким образом, последовательность $\mathbf{0}$ является кодовым словом любого линейного блокового кода.

Линейные блоковые коды (ЛБК) являются наиболее хорошо изученными, так как их проще анализировать. ЛБК проще имплементировать. (Максимальная) эффективность ЛБК близка к (максимальной) эффективности блоковых кодов в целом. В этой связи далее будем изучать только ЛБК.

7.2.1. ПОРОЖДАЮЩАЯ И ПРОВЕРОЧНАЯ МАТРИЦЫ

Для ЛБК (n, k) алгоритм формирования n-битового кодового слова $\mathbf{c}_m$ на основании k-битовой информационной последовательности $\mathbf{u}_m$ может быть описан с применением порождающей матрицы (generator matrix) $\mathbf{G}$ размером $k \times n$:

$$\mathbf{c}_m = \mathbf{u}_m \mathbf{G}, \quad 1 \le m \le 2^k$$

Строки $\mathbf{g}_i$ $1 \le i \le k$, порождающей матрицы является кодовыми словами для последовательностей единичного веса $(1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, …, $(0, \ldots, 0, 1)$. Следовательно,

$$\mathbf{c}_m = \sum_{i=1}^{k} u_{mi} \mathbf{g}_i,$$

где сложения выполняются по модулю 2, т.е. кодовые слова – все возможные линейные комбинации строк порождающей матрицы.

Два ЛБ кода $\square_1$ и $\square_2$ называются эквивалентными (equivalent), если их порождающие матрицы состоят из одинакового набора строк, возможно, с одинаковым перемешиванием.

7.2.1. ПОРОЖДАЮЩАЯ И ПРОВЕРОЧНАЯ МАТРИЦЫ

Если порождающая матрица имеет форму

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}],$$

где $\mathbf{I}_k$ – единичная матрица $k \times k$, а матрица $\mathbf{P}$ имеет размер $k \times (n - k)$, то ЛБК называется систематическим кодом.

Для систематического кода первые k элементов кодового слова совпадают с информационной последовательностью, а последние $(n - k)$ элементов называются проверочными битами (parity check bits).

Можно показать, что любой ЛБК имеет систематический эквивалентный ЛБК, порождающая матрица которого может быть получена элементарными преобразованиями строк и перемешиванием столбцов порождающей матрицы исходного кода.

7.2.1. ПОРОЖДАЮЩАЯ И ПРОВЕРОЧНАЯ МАТРИЦЫ

Учитывая, что ЛБК □ $(n,\ k)$ является k-мерным подпространством n-мерного пространства, его ортогональное дополнение, т.е. все n-мерные двоичные вектора ортогональные кодовым словам кода □, образуют $(n-k)$-мерное подпространство n-мерного пространства, т.е. код $(n,\ n-k)$, обозначаемый □□, и называемый дуальным (dual code) к коду □.

Порождающая матрица $\mathbf{H}$ дуального кода имеет размер $(n-k)\times n$ вместо $k\times n$ для исходного кода, её строки ортогональны строкам матрицы исходного кода и она называется проверочной матрицей (parity check matrix) для исходного кода.

Каждое кодовое слово исходного кода ортогонально каждой строке проверочной матрицы, поэтому

$$\mathbf{c}\mathbf{H}^t = 0,\ \forall \mathbf{c} \in \mathrm{C}$$

Вообще, если для произвольного n-мерного вектора $\mathbf{c}$ выполняется $\mathbf{c}\mathbf{H}^t = \mathbf{0}$, это значит, что $\mathbf{c}$ принадлежит ортогональному дополнению $\mathbf{H}$, т.е. $\mathbf{c}$ □ □ – необходимое и достаточное условие.

7.2.1. ПОРОЖДАЮЩАЯ И ПРОВЕРОЧНАЯ МАТРИЦЫ

Учитывая ортогональность строк $\mathbf{G}$ и $\mathbf{H}$, имеем

$$\mathbf{G}\mathbf{H}^t = \mathbf{0}$$

Для систематических БЛК $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]$ и, следовательно,

$$\mathbf{H} = [-\mathbf{P}^t \mid \mathbf{I}_{n-k}]$$

в чём легко убедиться, проверив, что $\mathbf{G}\mathbf{H}^t = \mathbf{0}$. Для двоичных кодов (GF(2)) $-\mathbf{P}^t = \mathbf{P}^t$ и

$$\mathbf{H} = [\mathbf{P}^t \mid \mathbf{I}_{n-k}]$$

7.2.2. ПОНЯТИЯ ВЕСА И РАССТОЯНИЯ

Вес кодового слова $\mathbf{c}_2$ □ □ – число ненулевых элементов – обозначается как $w(\mathbf{c})$. Учитывая что $\mathbf{0}$ является кодовым словом всех ЛБК, у каждого ЛБК есть кодовое слово с нулевым весом.

Хеммингово расстояние (Hamming distance) между двумя кодовыми словами $\mathbf{c}_1$, $\mathbf{c}_2$ □ □ обозначается как $d(\mathbf{c}_1, \mathbf{c}_2)$ и равно количеству элементов, в которых отличаются $\mathbf{c}_1$ и $\mathbf{c}_2$. Очевидно, что вес кодового слова равен его расстоянию до нулевого кодового слова $\mathbf{0}$.

Далее, учитывая, что для двух кодовых слов $\mathbf{c}_1$ и $\mathbf{c}_2$ линейного кода слово $\mathbf{c}_1 - \mathbf{c}_2$ также является кодовым словом, очевидно, что $d(\mathbf{c}_1, \mathbf{c}_2) = w(\mathbf{c}_1 - \mathbf{c}_2)$, т.е. для ЛБК имеется соответствие между весом и расстоянием.

Получается, что множество всех расстояний от текущего кодового слова $\mathbf{c}$ □ □ до остальных кодовых слов совпадает с множеством всех весов данного кода и, следовательно, не зависит от выбора кодового слова $\mathbf{c}$.

7.2.2. ПОНЯТИЯ ВЕСА И РАССТОЯНИЯ

Минимальным расстоянием (minimum distance) кода является минимальное расстояние между всеми возможными парами кодовых слов:

$$d_{\min} = \min_{\substack{\mathbf{c}_1, \mathbf{c}_2 \in \mathrm{C} \\ \mathbf{c}_1 \neq \mathbf{c}_2}} d(\mathbf{c}_1, \mathbf{c}_2)$$

Минимальным весом (minimum weight) кода является минимальный вес кодовых слов среди всех кодовых слов, кроме нулевого:

$$w_{\min} = \min_{\mathbf{c} \in \mathrm{C},\ \mathbf{c} \neq \mathbf{0}} w(\mathbf{c})$$

Для ЛБК минимальный вес совпадает с минимальным расстоянием.

Для ЛБК имеется связь между минимальным весом и столбцами проверочной матрицы:

Rem: $\mathbf{c}\mathbf{H}^t = \mathbf{0}$ это необходимое и достаточное условие того, что $\mathbf{c}$ □ □.

Выбрав слово с минимальным весом $w_{\min}$ (или, что то же самое, минимальным расстоянием $d_{\min}$), получим, что, как минимум, $d_{\min}$ столбцов матрицы $\mathbf{H}$ линейно зависимы. Учитывая, что для других слов расстояние будет не меньше, получаем, что минимальное число линейно зависимых столбцов матрицы $\mathbf{H}$ равно $d_{\min}$, т.е. пространство столбцов имеет размерность $(d_{\min} - 1)$.

7.2.2. ПОНЯТИЯ ВЕСА И РАССТОЯНИЯ

Для некоторых типов модуляции возможно установить простое соотношение между Хеммингловым и Евклидовым расстоянием для кодовых слов.

Для случая противоположных сигналов, например, BPSK, значения 0 и 1 кодовых слов отображаются на $-\sqrt{E_c}$ и $\sqrt{E_c}$ соответственно. Иначе говоря, кодовому слову $\mathbf{c}_m$ ставится в соответствие вектор $\mathbf{s}_m$ такой, что

$$s_{mj} = (2\,\mathrm{c}_{mj} - 1)\sqrt{E_c},\ 1 \le j \le n,\ 1 \le m \le M$$

При этом

$$d^2_{\mathbf{s}_m,\mathbf{s}_{m'}} = 4E_c d(\mathbf{c}_m, \mathbf{c}_{m'})$$

где $d_{\mathbf{s}m,\mathbf{s}m'}$ – Евклидово расстояние между модулированными последовательностями, а $d(\mathbf{c}_m,\mathbf{c}_{m'})$ – Хемингово расстояние между соответствующими кодовыми словами. Очевидно,

$$d^2_{\mathrm{E}\min} = 4E_c d_{\min}$$

Rem: $E_c = R_c E_b$

$$d^2_{\mathrm{E}\min} = 4R_c E_b d_{\min}$$

Для двоичных ортогональных сигналов, например, двоичной ЧМ, имеем:

$$d^2_{\mathrm{E}\min} = 2R_c E_b d_{\min}$$

7.2.3. ПОЛИНОМ РАСПРЕДЕЛЕНИЯ ВЕСОВ

Полином распределения весов (функция-счётчик кодовых слов с заданным весом) (weight distribution polynomial, WEP или weight enumeration function, WEF) указывает информацию о количестве кодовых слов A_i для каждого возможного веса i:

$$A(Z) = \sum_{i=0}^{n} A_i Z^i = 1 + \sum_{i=d_{\min}}^{n} A_i Z^i$$

Очевидно, что

$$A(1) = \sum_{i=0}^{n} A_i = 2^k$$

$$A(0) = 1$$

Другой вариант полинома дополнительно указывает количество соответствующих информационных последовательностей с заданным весом (input-output weight enumeration function, IOWEF)

$$B(Y, Z) = \sum_{i=0}^{n} \sum_{j=0}^{k} B_{ij} Y^j Z^i$$

где B_{ij} – количество кодовых слов с весом i, полученных из информационной последовательности с весом j. Очевидно, что

$$A_i = \sum_{j=0}^{k} B_{ij}$$

Для ЛБК $B(0, 0) = B_{00} = 1$.

7.2.3. ПОЛИНОМ РАСПРЕДЕЛЕНИЯ ВЕСОВ

Ещё один вариант полинома указывает количество кодовых слов с весом i, соответствующих информационным последовательностям с весом j:

$$B_j(Z) = \sum_{i=0}^{n} B_{ij} Z^i$$

7.2.4. ПОМЕХОУСТОЙЧИВОСТЬ ЛИНЕЙНЫХ БЛОКОВЫХ КОДОВ

Вероятность блоковой ошибки (Block Error Probability)

Для ЛБК набор расстояний от текущего кодового слова до всех остальных не зависит от кодового слова, поэтому без потери точности можно рассматривать передачу слова $\mathbf{0}$. Учитывая неравенство из аддитивной границы, имеем:

$$P_e \le \sum_{\substack{\mathbf{c}_m \in \mathrm{C} \\ \mathbf{c}_m \neq \mathbf{0}}} P_{\mathbf{0}\to\mathbf{c}_m}$$

где $P_{\mathbf{0}\to\mathbf{c}m}$ – вероятность перепутать кодовые слова в двоичной системе, зависящая от хеммингова расстояния между $\mathbf{0}$ и $\mathbf{c}_m$, т.е. от веса $w(\mathbf{c}_m)$. По-видимому, при фиксированном весе эта вероятность одинакова, тогда

$$P_e \le \sum_{i=d_{\min}}^{n} A_i P_2(i)$$

где $P_2(i)$ – вероятность перепутать кодовые слова с хемминговым расстоянием i в двоичной системе (pairwise error probability, PER).

Можно показать, что

$$P_e \le \sum_{i=d_{\min}}^{n} A_i \Delta^i = A(\Delta) - 1 \le \left\langle \Delta^i \underset{i \ge d_{\min}}{\le} \Delta^{d_{\min}} \right\rangle \le (2^k - 1)\Delta^{d_{\min}}$$

$$\Delta = \sum_{y \in Y} \sqrt{p(y \mid 0)\, p(y \mid 1)}$$

7.2.4. ПОМЕХОУСТОЙЧИВОСТЬ ЛИНЕЙНЫХ БЛОКОВЫХ КОДОВ

Вероятность битовой ошибки (Bit Error Probability)

Среднее число ожидаемых битовых ошибок:

$$\bar{b} \le \sum_{j=0}^{k} j \sum_{i=d_{\min}}^{n} B_{ij} P_2(i)$$

Учитывая, что для $0 < i < d_{\min}$ имеем $B_{ij} = 0$, можно упростить:

$$\bar{b} \le \sum_{j=0}^{k} j \sum_{i=0}^{n} B_{ij} P_2(i)$$

Средняя битовая ошибка

$$P_b = \bar{b} / k \le \frac{1}{k} \sum_{j=0}^{k} j \sum_{i=0}^{n} B_{ij} P_2(i) \le \frac{1}{k} \sum_{j=0}^{k} j \sum_{i=0}^{n} B_{ij} \Delta^i$$

Rem: $B_j(Z) = \sum_{i=0}^{n} B_{ij} Z^i$

$$P_b \le \frac{1}{k} \sum_{j=0}^{k} j \sum_{i=0}^{n} B_{ij} \Delta^i = \frac{1}{k} \sum_{j=0}^{k} j B_j(\Delta)$$

7. ЛИНЕЙНЫЕ БЛОКОВЫЕ КОДЫ

7.3.1. КОДЫ С ПОВТОРЕНИЕМ

Двоичный код с повторениями (repetition code) $(n,\ 1)$ состоит из двух возможных кодовых слов длиной n: $\mathbf{0}$, $\mathbf{1}$. Кодовая скорость $R_c = 1/n$, минимальное расстояние $d_{\min} = n$.

Дуальным кодом является код $(n,\ n\ -\ 1)$, состоящий из всех двоичных последовательностей длины n с проверкой на чётность, минимальное расстояние $d_{\min} = 2$.

7.3.2. КОДЫ ХЕММИНГА

Коды Хемминга (Hamming codes) – одни из первых предложенных кодов, являются линейными кодами с параметрами: $n = 2^m - 1$, $k = 2^m - m - 1$, $m \geq 3$.

Особенностью кодов Хемминга является вид их проверочной матрицы $\mathbf{H}$ размером $(n-k) \times n = m \times (2^m - 1)$. В её $(2^m - 1)$ столбцах находятся все возможные двоичные числа длины m, кроме нуля.

Таким образом, сумма двух любых столбцов всегда даст один из имеющихся столбцов, т.е. независимо от параметра m в матрице $\mathbf{H}$ всегда есть три линейно зависимых столбца. Значит, $d_{\min} = 3$.

Пример проверочной матрицы систематического кода (7, 4) для $m = 3$:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Можно показать, что

$$A(Z) = \frac{1}{n+1}\left[(1+Z)^n + n(1+Z)^{(n-1)/2}(1-Z)^{(n+1)/2} \right]$$

7.3.3. КОДЫ МАКСИМАЛЬНОЙ ДЛИНЫ

Коды максимальной длины (maximum-length codes) являются дуальными к кодам Хемминга, т.е. это коды $(2^m - 1, m)$, $m \geq 3$.

Порождающая матрица этих кодов совпадает с проверочной матрицей кодов Хемминга.

Отличительной особенностью этих кодов является то, что это коды с фиксированным весом, т.е. вес всех кодовых слов, кроме нулевого, одинаковый и равен 2^{m-1}.

Следовательно,

$$A(Z) = 1 + (2^m - 1) Z^{2^{m-1}}$$

7.3.4. КОДЫ РИДА-МАЛЛЕРА

Коды Рида-Маллера (Reed-Muller codes) известны благодаря существованию простого алгоритма их декодирования.

Для длины кода $n = 2^m$ и порядка $r < m$ имеем

$$n = 2^m$$

$$k = \sum_{i=0}^{r} \binom{m}{i}$$

$$d_{\min} = 2^{m-r}$$

7.3.5. КОДЫ АДАМАРА

Кодовые слова кода Адамара (Hadamard code) – это строки матрицы Адамара. Свойством строк матрицы Адамара $n \times n$ является то, что каждая пара строк отличается ровно в $n/2$ позициях. Одна из строк нулевая. Остальные состоят из $n/2$ нулей и $n/2$ единиц.

Пример:

$$\mathbf{M}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \ \mathbf{M}_{2n} = \begin{bmatrix} \mathbf{M}_n & \mathbf{M}_n \\ \mathbf{M}_n & \overline{\mathbf{M}}_n \end{bmatrix}, \ \mathbf{M}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \ \overline{\mathbf{M}}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Строки $\mathbf{M}_4$ и комплиментарной ей матрицы образуют код с $n = 4$ и $2n = 8$ кодовыми словами, минимальное расстояние $d_{\min} = n/2 = 2$.

В общем случае: $n = 2^m$, $k = \log_2 2n = \log_2 2^{m+1} = m + 1$, $d_{\min} = n/2 = 2^{m-1}$, m – натуральное число.

7.3.6. КОДЫ ГОЛЕЯ

(Совершенный) код Голея (the Golay code) – двоичный линейный код (23, 12) с $d_{\min} = 7$.

Расширенный код Голея (the extended Golay code) (24, 12) получается путём добавления одной проверки на чётность, для него $d_{\min} = 8$.

Для кодов Голея известны полиномы распределения весов:

$$A_G(Z) = 1 + 253Z^7 + 506Z^8 + 1288Z^{11} + 1288Z^{12} + 506Z^{15} + 253Z^{16} + Z^{23}$$

$$A_{EG}(Z) = 1 + 759Z^8 + 2576Z^{12} + 759Z^{16} + Z^{24}$$

7. ЛИНЕЙНЫЕ БЛОКОВЫЕ КОДЫ

7.12. КОДИРОВАНИЕ ДЛЯ КАНАЛОВ С ПАКЕТНЫМИ ОШИБКАМИ

Большинство хорошо изученных кодов эффективно работают в условиях каналов без памяти, т.е. когда появления ошибок независимы. Это верно, например, для АБГШ.

Однако, для каналов, характеризующихся многолучёвостью (multipath) и замираниями (fading), а также, например, для каналов магнитной записи информации, характерно появление пакетных ошибок.

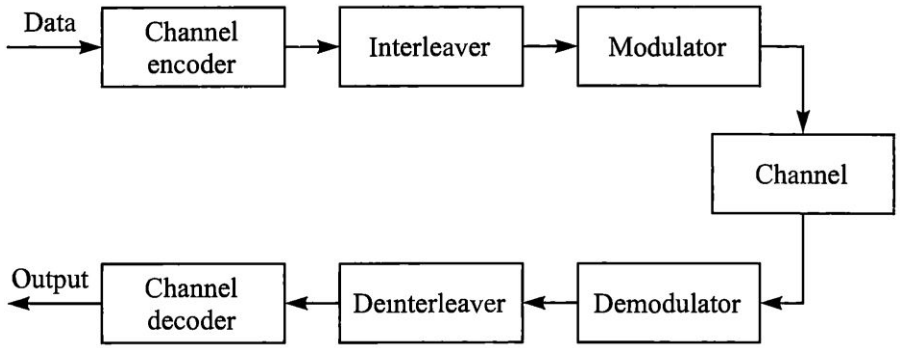
Отметим, что некоторые коды, направленные на борьбу с независимыми ошибками, тем не менее, могут справляться с пакетными ошибками. Яркий пример – коды Рида-Соломона, так как для них пакетная ошибка может приводить к ошибке всего лишь в нескольких символах.
Известный пример кодов, направленных на борьбу с пакетными ошибками – коды Файра (Fire codes).

Можно показать, что систематический (n, k) код, содержащий $(n-k)$ проверочных бит может исправлять пакетные ошибки длиной

$$b \le \left\lfloor \frac{1}{2}(n-k) \right\rfloor$$

однако возможно существенно улучшить этот результат!

7.12. КОДИРОВАНИЕ ДЛЯ КАНАЛОВ С ПАКЕТНЫМИ ОШИБКАМИ

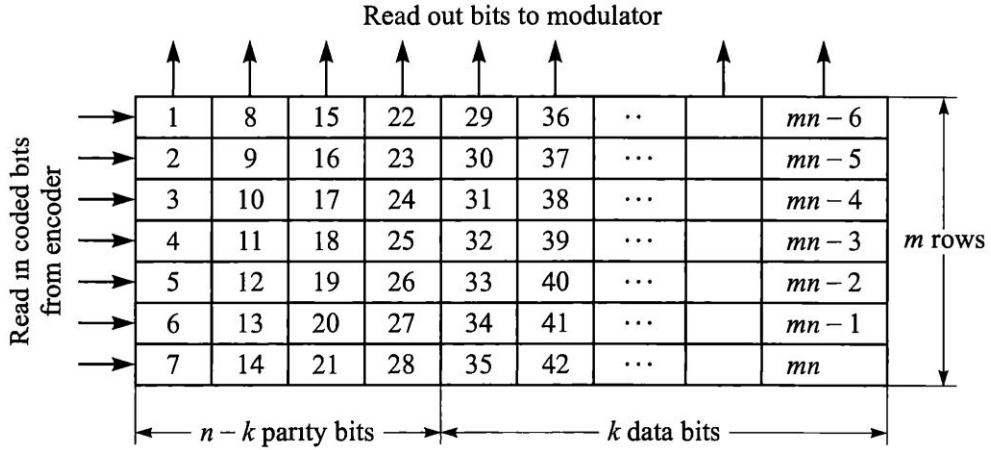


Перед подачей на модулятор можно перемешивать модуляционные символы так, чтобы рассеять / декоррелировать ошибки. В приёмнике перед декодированием нужно выполнять обратную процедуру.

При таком подходе применение кодов, рассчитанных на независимые ошибки, снова становится эффективным.

Перемежители бывают блочными (block) и последовательными (свёрточными) (convolutional).

7.12. КОДИРОВАНИЕ ДЛЯ КАНАЛОВ С ПАКЕТНЫМИ ОШИБКАМИ



Блочные перемежители записывают поступающие биты по строкам в таблицу размером m строк на n столбцов. При этом n – длина кодового слова, m – порядок (degree) перемежителя, т.е. число помещающихся в него кодовых слов.

Биты считываются по столбцам и подаются на модулятор.

Как результат, пакетные ошибки длиной $l = mb$ разбиваются на m пакетов длиной b. С каждым из этих пакетов может справиться декодер.

Принцип работы свёрточного перемежителя (Ramsey и Forney) такой же, но его удобнее использовать в паре со свёрточным кодом.

7. ЛИНЕЙНЫЕ БЛОКОВЫЕ КОДЫ

7.13. КОМБИНИРОВАНИЕ КОДОВ

Эффективность блокового кода определяется его исправляющей способностью, т.е. количеством ошибок, которые он может исправлять, а значит, его минимальным расстоянием. Для фиксированной кодовой скорости R_c можно предложить множество кодов с кодовыми блоками разной длины. Обычно большей длине блока соответствует большее минимальное расстояние, а значит, большая исправляющая способность (в этом можно убедиться исследуя выражения для границ минимальных расстояний блоковых кодов).

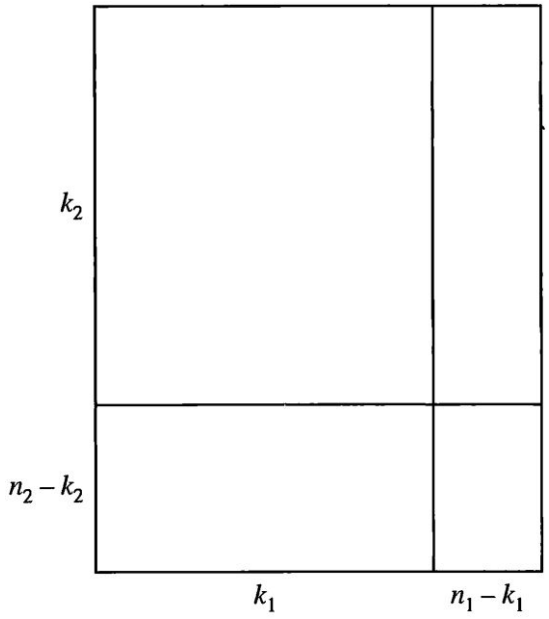
Недостатком увеличения длины блока является экспоненциальное увеличение вычислительной сложности декодирования.

Возможен альтернативный подход: комбинирование двух относительно простых кодов с короткими блоками таким образом, чтобы получать коды большей длины и с лучшими дистантными характеристиками. При этом для декодирования можно применять подоптимальное вычислительно-эффективное декодирование, основанное на декодировании составляющих кодов.

7.13.1. КОДЫ ПРОИЗВЕДЕНИЯ (PRODUCT CODES)

Пусть имеются два систематических линейных блоковых $\square_i$ кода (n_i, k_i) с минимальным расстоянием $d_{\min,i}$, $i = 1, 2$. Код произведения – это систематический линейный блоковый код $(n_1 n_2, k_1 k_2)$ со структурой, представленной на рисунке.

Строки кодируются кодом $\square_1$, столбцы кодом $\square_2$. Биты снизу справа могут быть получены кодированием либо строк, либо столбцов. Можно показать, что разницы нет.



Скорость кода произведения – произведение скоростей составляющих кодов.

Можно показать, что минимальное расстояние кода произведения равно произведению минимальных расстояний составляющих кодов

$$d_{\min} = d_{\min,1} d_{\min,2}$$

Следовательно исправляющая способность кода произведения равна

$$t = \left\lfloor \frac{d_{\min,1} d_{\min,2} - 1}{2} \right\rfloor$$

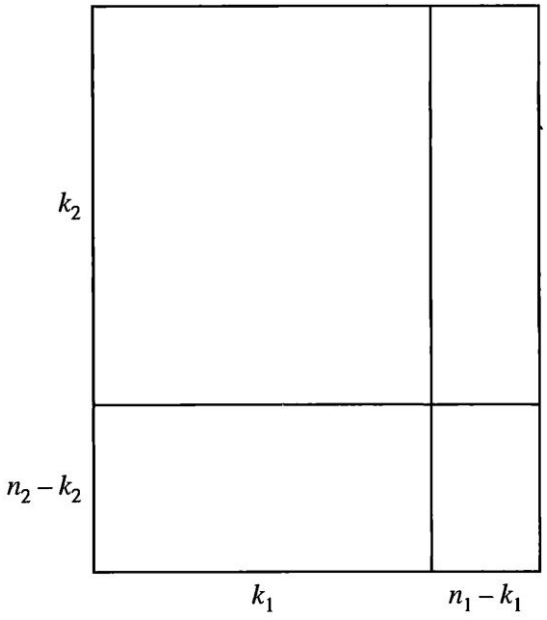
в случае использования вычислительно сложной оптимальной схемы декодирования.

7.13.1. КОДЫ ПРОИЗВЕДЕНИЯ (PRODUCT CODES)

Исправляющая способность каждого кода:

$$t_i = \left\lfloor \frac{d_{\min,i} - 1}{2} \right\rfloor$$

Предположим, что при передаче кодового блока из ($n_1 n_2$ бит) произошло менее, чем $(t_1 + 1)(t_2 + 1)$ ошибок.



Очевидно, что число строк, в которых количество ошибок больше t_1, меньше либо равно t_2. В противном случае общее количество ошибок получилось бы не меньше $(t_1 + 1)(t_2 + 1)$.

Получается, что после декодирования кода $\square_1$ ошибки останутся максимум в t_2 строках. Очевидно, декодирование кода $\square_2$ в столбцах справится с этими ошибками.

Таким образом, при использовании простого последовательного двух-шагового декодирования, кодом произведения можно исправить до

$$\tau = (t_1 + 1)(t_2 + 1) - 1 = t_1 t_2 + t_1 + t_2$$

ошибок.

Пример

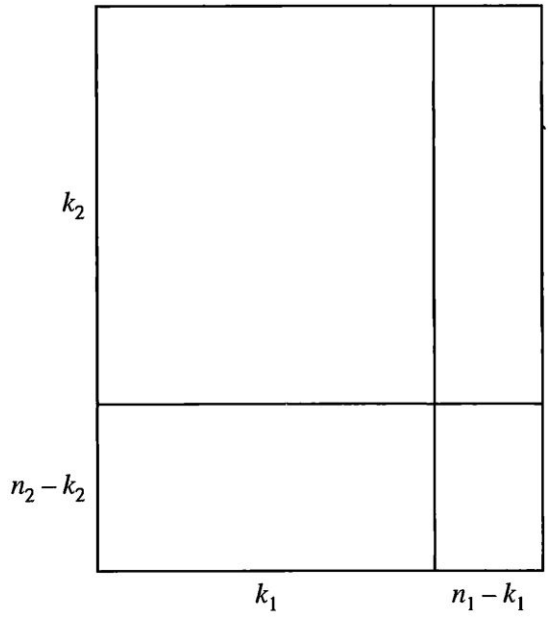
Код БЧХ (255, 123), для которого $d_{\min,1} = 39$, $t_1 = 19$ и Код БЧХ (15, 7), для которого $d_{\min,2} = 5$, $t_2 = 2$. Код произведения имеет минимальное расстояние $39 \times 5 = 195$ и, значит, может исправлять до 97 ошибок при использовании вычислительно сложного оптимального алгоритма декодирования.

При использовании простого последовательного двух-шагового декодирования исправляющая способность равна $(19 + 1)(2 + 1) - 1 = 59$ ошибок, что, конечно, заметно меньше, чем 97, но достигается многократно меньшими вычислительными затратами.

Другим подходом для декодирования кодов произведения является итеративное декодирование (iterative decoding) (crossword puzzle solving).

Идея заключается в том, чтобы выполняя декодирование как строк, так и столбцов, не выносить жёсткие решения а лишь выдавать меру уверенности в том или ином значении, т.е. выдавать мягкие решения (например, ЛОП).

Тогда каждый следующий шаг будет улучшать



7.13.2. КАСКАДНЫЕ КОДЫ (CONCATENATED CODES)

Обычно при каскадном кодировании на основе двух кодов используются двоичный и недвоичный коды, при этом кодовые слова двоичных кодов трактуются как символы недвоичного кода.

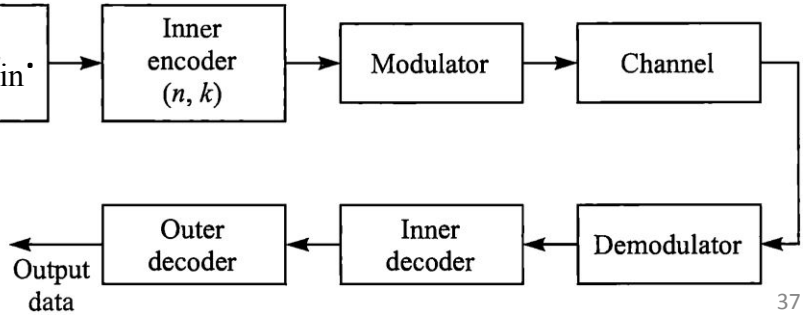
Ближе к каналу обычно находится двоичный код, который называется внутренним (inner code). Расположенный дальше от канала недвоичный код называется внешним (outer code).

Пусть внешний код (N, K), а внутренний (n, k).

- Блоки из kK бит разделяются на K групп – символов. Каждый символ состоит из k бит.
- K k-ичных символов кодируются внешним недвоичным кодом в N k-ичных символов.
- Каждый из N символов кодируется внутренним двоичным кодом из k бит в n бит.

Таким образом, длина кодового блока каскадного кода равна Nn и содержит Kk информационных бит, т.е. имеем (Nn, Kk) код.

Минимальное расстояние – произведение $d_{\min} D_{\min}$.
Кодовая скорость – произведение Kk/Nn.



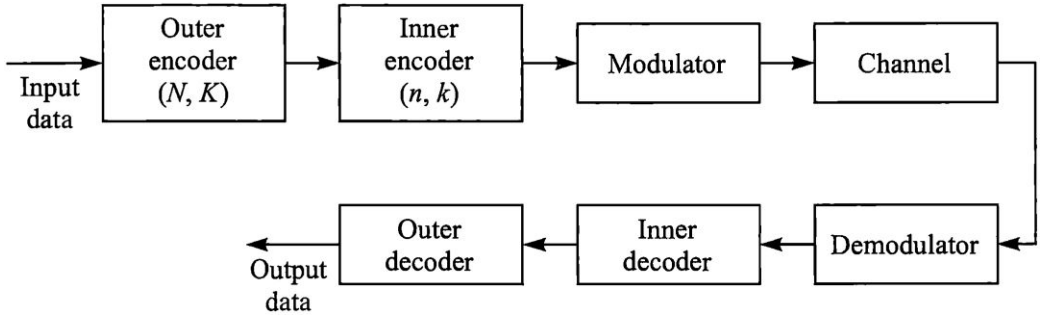
7.13.2. КАСКАДНЫЕ КОДЫ (CONCATENATED CODES)

Декодирование с жёсткими решениями:
Внутренний декодер выполняет МП декодирование с жёсткими решениями для каждого кодового слова.
Как только будут получены жёсткие решения для N слов, внешний декодер выполнит МП декодирование кодового слова.

Если для внутреннего кода возможно выполнять декодирование с мягкими решениями, то это улучшит результаты (путём повышения вычислительной сложности).

Внешнее декодирование обычно выполняется с жёсткими решениями, однако в случае многолучевых каналов применение декодирования с мягкими решениями для внешнего кода также может повысить эффективность декодирования.

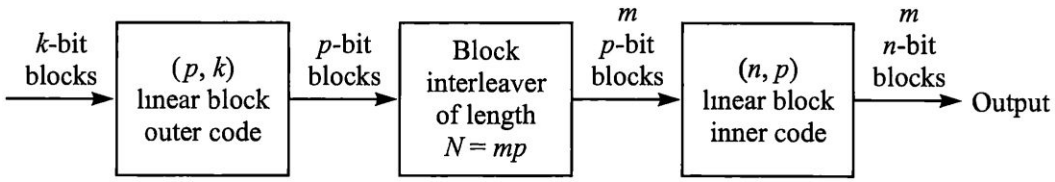
Распространённый вариант – внешний код Рида-Соломона и внутренний свёрточный код.



Последовательное и параллельное соединение с перемежителями (Serial and Parallel Concatenation with Interleavers)

Для построения каскадных кодов с экстремально длинными кодовыми блоками используются перемежители и двоичные систематические коды.

При последовательном соединении (Serially Concatenated Block Code, SCBC) перемежитель вставляется между внутренним и внешним кодом.



Обычно m – большое натуральное число.

Кодирование:

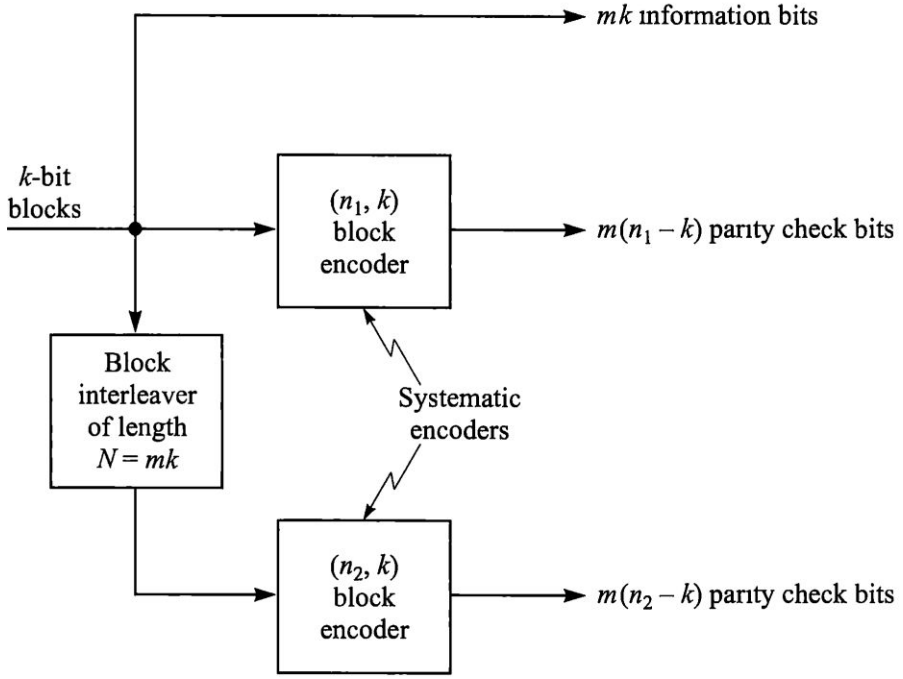
- mk информационных бит кодируются внешним кодом, обеспечивая mp кодовых бит.
- mp перемешанных бит разбиваются на блоки по p штук и поступают на внутренний кодер. На выходе получается mn кодовых бит.

Скорость кода $R_c^s = k/n$ – по-прежнему, произведение скоростей составляющих кодов.

Последовательное и параллельное соединение с перемежителями (Serial and Parallel Concatenation with Interleavers)

При параллельном соединении (Parallel Concatenated Block Code, PCBC) перемежению подвергаются информационные биты, поступающие на один из двух систематических кодов.

Длина итогового кодового блока:
$m(n_1 + n_2 - k)$.
Кодовая скорость:
$R_c^p = k / (n_1 + n_2 - k)$.



Обычно декодирование как SCBC, так и PCBC выполняется итеративно с применением декодеров с мягкими решениями.