ЛАБОРАТОРНАЯ РАБОТА № 1
Способы задания и основные характеристики сверточных кодов

Сверточные коды широко применяются в самых различных областях техники передачи и хранения информации. Наиболее наглядными примерами их эффективного применения являются системы космической связи, системы мобильной связи, модемы для телефонных каналов. В частности, протоколы V.32, V.34, ADSL, HDSL используют для защиты от ошибок сверточные коды в сочетании с декодированием по максимуму правдоподобия по алгоритму Витерби. В данной работе изучаются способы математического описания кодеров сверточных кодов, самих кодов и даются определения характеристик сверточных кодов.

1.1. Представление сверточного кода порождающими многочленами

Схема кодера двоичного сверточного кода в общем виде представлена на рис. 1.1. Кодер содержит k двоичных регистров сдвига длин $m_1,...,m_k$. Входами регистров сдвига являются информационные символы. Выходы ячеек регистров соединены с сумматорами по модулю 2. Всего сумматоров n.

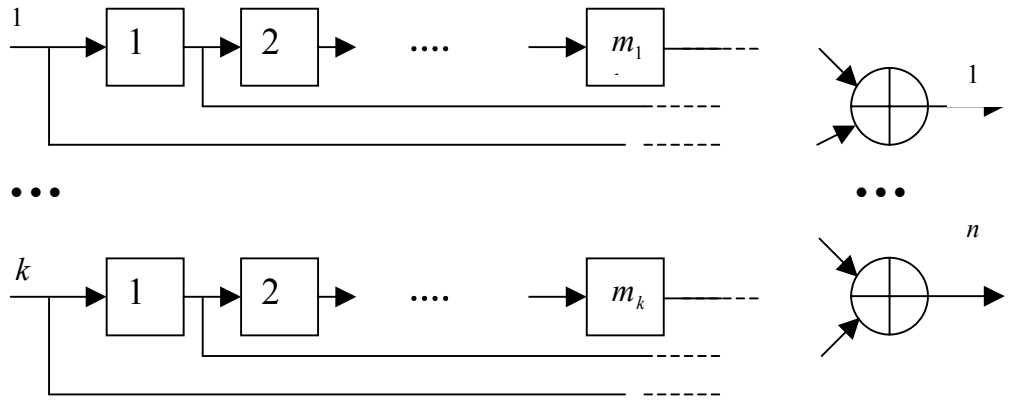


Рис.1.1 Общая структурная схема кодера сверточного кода со скоростью *k/n*

На каждом такте работы кодера на его вход поступает блок из k информационных символов. Эти символы и символы, хранящиеся в данный момент в регистрах кодера, поступают на входы тех сумматоров, которые подключены к соответствующим ячейкам. Результаты сложения по модулю два поступают на выход схемы. После этого в каждом из регистров происходит сдвиг, новые информационные символы записываются в первые ячейки, а содержимое остальных ячеек сдвигается на один разряд.

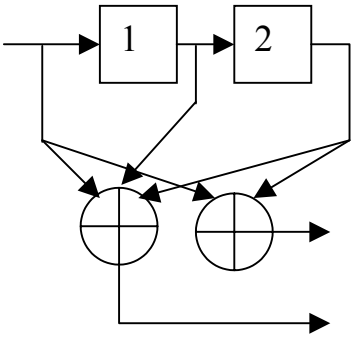
Содержимое $m_1 + m_2 + ... + m_k$ ячеек регистров сдвига в каждый конкретный момент времени называется текущим *состоянием кодера.* Предположим, что в начальный момент времени кодер находится в некотором заранее известном состоянии. Примем для определенности это начальное

состояние нулевым. Рассмотрим процесс кодирования полубесконечной информационной последовательности. Выходы сумматоров на каждом такте работы называются *кодовыми символами* сверточного кода. Полубесконечная последовательность кодовых символов называется *кодовым словом* сверточного кода. Множество всевозможных кодовых слов образует *сверточный код.*

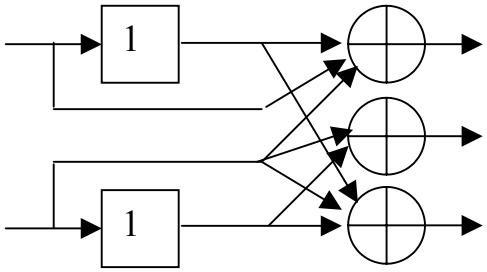
На каждом такте работы кодера k информационных символа используются для формирования n кодовых символов. Отношение

$$R = k / n$$

называется скоростью сверточного кода. Примеры кодеров сверточных кодов приведены на рис. 1.2.



а) Код (5,7). $R = 1/2$,
$\mathbf{g_1} = (101)$, $\mathbf{g}_2 = (111)$,
$\mathbf{G}_0 = [11]$; $\mathbf{G}_1 = [01]$; $\mathbf{G}_2 = [11]$;



б) $R = 2/3$, $m = 1$, $\nu = 2$,
$\mathbf{G}_0 = \begin{bmatrix} 101 \\ 011 \end{bmatrix}$, $\mathbf{G}_1 = \begin{bmatrix} 100 \\ 111 \end{bmatrix}$;

Рис. 1.2. Примеры сверточных кодеров

Суммарная длина регистров сверточного кодера $\nu = \sum_{i=1}^{n} m_i$ называется длиной кодового ограничения кода, а максимальная длина регистров $m = \max_i \{m_i\}$ называется задержкой кодера. Для кодов со скоростью $1/n$ память и кодовое ограничение совпадают.

Рассматриваемые схемы кодеров сверточных кодов полностью описываются связями ячеек регистров сдвига с выходными сумматорами. Существует несколько общепринятых форм представления этих связей. Начнем с кодов со скоростью $1/n$. Связи каждого из n сумматоров с ячейками одного регистра длины m_i записываются в виде двоичного вектора $\mathbf{g}_i = (g_{0i},...,g_{mi})$, $i = 1,...,n$. Ноль означает отсутствие связи, единица - наличие. Векторы $\mathbf{g}_i$ называют порождающими. В таблицах кодов порождающие векторы приводят в восьмеричной форме. Например, генератор $\mathbf{g} = (1010111)$ будет записан как (125). Другие примеры показаны на рис. 1.2.

Порождающие векторы записывают также в виде полиномов формальной переменной D. Например, порождающие полиномы кодера, показанного на рис. 1.2а, имеют вид

$$g_1(D) = 1 + D^2,$$
$$g_2(D) = 1 + D + D^2.$$

Эту совокупность полиномов можно также записать в виде матричного порождающего полинома

$$\mathbf{G}(D) = \mathbf{G}_0 + \mathbf{G}_1 D + \mathbf{G}_2 D^2,$$

где

$$\mathbf{G}_0 = [11]; \quad \mathbf{G}_1 = [01]; \quad \mathbf{G}_2 = [11].$$

Полиномиальное и матричное полиномиальное представление кодера удобно тем, что оно позволяет записать процесс кодирования в виде умножения полиномов формальной переменной D. Предположим, что на вход кодера подается информационная последовательность $\mathbf{u} = [101000...]$. Эта последовательность может быть записана в виде полинома

$$u(D) = 1 + D^2 + ...$$

Нетрудно убедиться в том, что на выходах сумматоров будут наблюдаться кодовые последовательности

$$c_1(D) = u(D)g_1(D) = 1 + D^4 + ... ;$$
$$c_2(D) = u(D)g_2(D) = 1 + D + D^3 + D^4 + ...$$

Матричная запись имеет вид

$$\mathbf{c}(D) = u(D)\mathbf{G}(D) =$$
$$= \mathbf{G}_0 + \mathbf{G}_1 D + (\mathbf{G}_0 + \mathbf{G}_2)D^2 + \mathbf{G}_1 D^3 + \mathbf{G}_2 D^4 + ... =$$
$$= [11] + [01]D + [00]D^2 + [01]D^3 + [11]D^4 + ...$$

Следовательно, на выходе кодера будет сформировано кодовое слово $[1101000111...]$.

В этом примере был рассмотрен код со скоростью 1/2. В случае кода со скоростью k/n схема кодера содержит k регистров сдвига. Обозначим через m максимальную длину регистра. Связи ячеек, имеющих одинаковый номер i с n выходными сумматорами, по-прежнему описываются матрицами $\mathbf{G}_i$, однако, поскольку таких ячеек теперь k, размерность каждой из матриц будет равна $k \times n$. Кодер будет описан порождающим полиномом

$$\mathbf{G}(D) = \mathbf{G}_0 + \mathbf{G}_1 D + ... + \mathbf{G}_m D^m.$$

Входная последовательность кодера будет представлена векторным полиномом

$$\mathbf{u}(D) = \mathbf{u}_0 + \mathbf{u}_1 D + \mathbf{u}_2 D^2,$$

в котором подпоследовательности $\mathbf{u}_i$, $i = 0,1,...$ имеют размерность $1 \times k$. Кодирование представляется как умножение полиномов

$$\mathbf{c}(D) = \mathbf{u}(D)\mathbf{G}(D). \qquad (1.1)$$

Пример сверточного кода со скоростью 2/3 приведен на рис. 1.2б.

1.2. Порождающая матрица сверточного кода

Из описания кодера или из формулы (1.1) непосредственно следует, что сверточный код является линейным. Это означает, что множество кодовых слов образует линейное пространство полубесконечных двоичных последовательностей.

Удобной формой представления блоковых линейных кодов является описание с помощью порождающей матрицы кода, строками которой, как известно, служат базисные векторы кода.

Хотя размерность пространства кодовых слов сверточного кода бесконечна, его регулярная структура позволяет в явном виде указать базис пространства и выписать порождающую матрицу сверточного кода.

Рассмотрим сначала коды со скоростью $1/n$. Заметим, что информационные последовательности вида

$$\mathbf{u}_0 = [100,...];$$
$$\mathbf{u}_1 = [010...];$$
$$\dots$$

образуют в совокупности базис линейного пространства входных последовательностей кодера, поскольку любая информационная последовательность может быть представлена в виде линейной комбинации $\mathbf{u}_i$, $i = 1,2,...$ Каждой линейной комбинации информационных последовательностей $\mathbf{u}_i$ соответствует кодовая последовательность, равная линейной комбинации кодовых слов, соответствующих $\mathbf{u}_i$. Отсюда следует, что кодовые слова, соответствующие этим информационным последовательностям, образуют базис в пространстве кодовых слов. Из (1.1) следует, что порождающая матрица сверточного кода может быть записана в виде

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \mathbf{G}_2 & \dots & \mathbf{G}_m & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_{m-1} & \mathbf{G}_m & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_0 & \dots & \mathbf{G}_{m-2} & \mathbf{G}_{m-1} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}, \qquad (1.2)$$

где через $\mathbf{0}$ обозначена нулевая матрица размерности $1 \times n$. В случае кодов со скоростью k/n формула (1.2) для порождающей матрицы сверточного кода также верна, но размерность составляющих подматриц будет $k \times n$.

1.3. Кодовое дерево сверточного кода и решетчатая диаграмма

Вернемся к коду (5,7), кодер которого показан на рис.1.2а. Считаем, что начальное состояние кодера было нулевым. Это состояние и все предыдущее мы будем отображать в виде узлов дерева. Возможные переходы из состояния в состояние соответствуют ребрам дерева. Каждому такому ребру соответствует некоторый блок из $n = 2$ кодовых символов. Несколько начальных ярусов дерева, соответствующего коду (5,7), показано на рис. 1.3. Это дерево называют кодовым. В связи с такой графической интерпретацией сверточного кода параметр n называют длиной ребра кода.

По кодовому дереву легко определить кодовое слово, соответствующее заданной информационной последовательности. На рис. 1.3 движение вверх соответствует информационному символу 0, движение вниз - символу 1. Например, информационной последовательности 010110… соответствует кодовое слово 00111000…

Обращает на себя внимание то, что число узлов дерева быстро растет от яруса к ярусу. В то же время, каждому узлу соответствует некоторое состояние кодера. Число различных состояний кода со скоростью $1/n$ и с длиной кодового ограничения m в точности равно 2^m. Следовательно, начиная с яруса

с номером $m+1$, найдутся узлы соответствующие одинаковым состояниям. Поддеревья, начинающиеся в этих узлах, полностью идентичны. Чтобы сделать графическое представление кода более компактным, имеет смысл объединить узлы, соответствующие одинаковым состояниям и отображать их в виде одного узла.

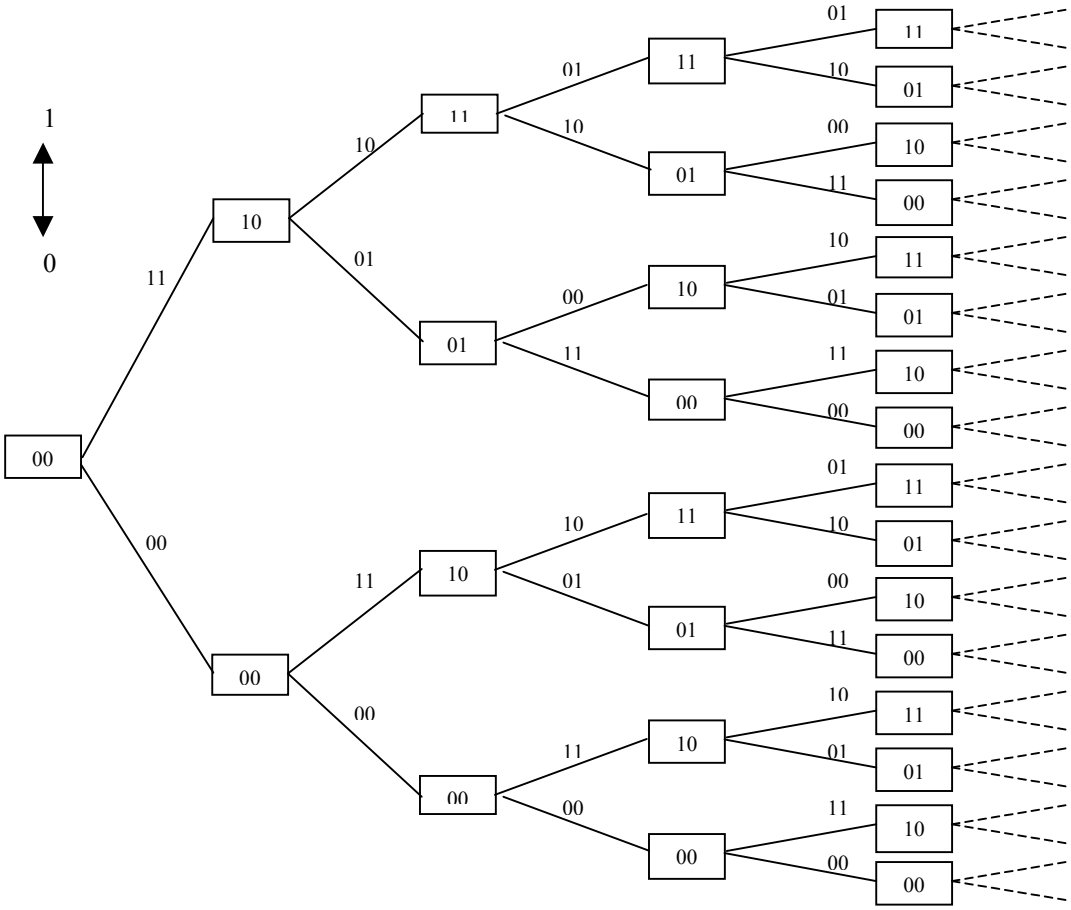


Рис. 1.3. Кодовое дерево кода (5,7)

Полученный таким образом граф называют решетчатой диаграммой сверточного кода. Решетчатая диаграмм кода (5,7) приведена на рис. 1.4. В общем случае кода со скоростью $1/n$ и кодовым ограничением m на ярусах с номерами $t = 1, 2, \ldots, m-1$ располагается 2^t узлов, и из каждого исходит $q = 2$ ребра. На ярусах с номерами $t \geq m$ число узлов равно 2^m, в каждый узел входит 2 ребра и из каждого узла исходит 2 ребра. В случае кода со скоростью k/n и кодовым ограничением ν максимальное число состояний равно 2^ν, число ребер, исходящих из каждого, узла равно 2^k.

По заданной информационной последовательности легко проследить кодовое слово. Поступление на вход кодера информационного символа 0 соответствует переходу в то из двух доступных из данного узла состояний, номер которого меньше. Поступление единицы приводит к переходу в состояние с большим номером. На рис. 1.4 видно, что в первом случае мы двигаемся по решетке "вниз", во втором - "вверх".

Представление кода решетчатой диаграммой чрезвычайно удобно для анализа декодирования сверточных кодов. Для изучения характеристик сверточных кодов удобно представление кодера в виде конечного автомата. Для кода (5,7) конечный автомат показан на рис. 1.5. Конечный автомат описывается графом, узлы которого соответствуют состояниям кодера, а ребра - переходам из состояние в состояние. Видно, что из каждого состояния выходят ровно два ребра, соответствующие двум возможным значениям входных символов. Каждому ребру приписаны кодовые символы, порождаемы кодером при данном переходе.

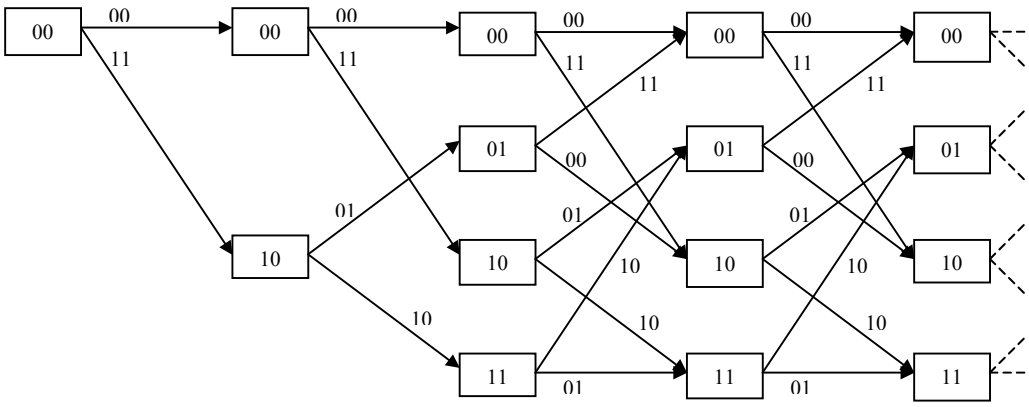


Рис. 1.4. Решетчатая диаграмма кода (5,7)

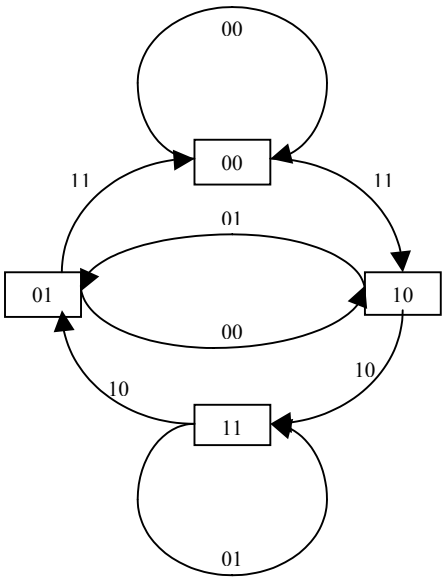


Рис. 1.5. Конечный автомат, описывающий работу кодера кода (5,7)

1.4. Свободное расстояние. Спектр

Как и в случае блоковых кодов, способность сверточного кода исправлять ошибки в канале связи определяется тем, насколько "далеко" друг от друга располагаются его кодовые слова. Для многих практических задач хорошей мерой близости служит расстояние Хэмминга между кодовыми словами.

Минимальным расстоянием блокового кода называют минимальное из попарных расстояний между кодовыми словами. Особенность сверточного кода заключается в том, что его слова имеют бесконечную длину. Тем не менее минимальное расстояние между парами кодовых слов определено однозначно и эту величину называют *свободным расстоянием* сверточного кода. Обозначим ее как d_f.

Мы определили сверточный код как линейный код. Для блоковых линейных кодов минимальное расстояние может быть вычислено как минимальный вес ненулевого кодового слова. Точно также для сверточного кода свободное расстояние находится как путь в его решетке, имеющий наименьший вес. Этот путь будет иметь бесконечную длину, но почти на всей длине будет совпадать с нулевым путем. При поиске пути минимального веса достаточно рассматривать только пути, ответвляющиеся от нулевого на начальном ярусе и затем сливающиеся с нулевым путем. Это множество путей называют *первым неправильным поддеревом*. Легко установить с помощью рис. 1.4, что свободное расстояние кода (5,7) равно $d_f = 5$. Слово минимального веса соответствует пути, который выходит из узла 00, затем проходит через узлы 10, 01, 00, 00, … Этот же путь легко проследить и по диаграмме конечного автомата, представленной на рис. 1.5.

При анализе помехоустойчивости сверточных кодов выясняется, что на величину вероятности ошибки декодирования влияет не только свободное расстояние кода, но и то, как много слов малого веса содержит код. Обозначим через N_d число слов веса d, соответствующих путям, ответвляющимся от нулевого пути на нулевом ярусе (путям первого неправильного поддерева). Набор чисел $\{N_d\}$, $d = d_f, d_f + 1, \ldots$ называют *спектром сверточного кода.* Способы вычисления спектров и связь спектров с вероятностью ошибки декодирования будет исследоваться в следующих лабораторных работах.

1.5. Катастрофические кодеры

Рассмотрим влияние ошибок декодирования сверточного кода на точность восстановления передаваемого сообщения. Предположим, что кодирование выполнялось кодером (5,7) с кодовым ограничением 2 и передавалась информационная последовательность из бесконечного числа подряд идущих нулей. Пусть в результате ошибок в канале связи декодер принял решение в пользу кодового слова минимального веса, т. е. кодового слова 11 01 11 00 00 … Этому кодовому слову соответствует информационная последовательность 100…. Это означает, что данная ошибка декодирования привела к искажению одного бита информации. Если же ошибки в канале привели к декодированию в пользу слова 11 10 10 11 00 00… веса 6, то с выхода декодера получателю поступит информационная последовательность 1100…, т.е. уже 2 бита будут воспроизведены неправильно.

Таким образом, если важна вероятность ошибки на бит, то при анализе свойств кода нужно учитывать не только его спектральные характеристики, но и то, как ошибки в канале могут отразиться на передаваемом сообщении. Рассмотрим, например, кодер, показанный на рис. 1.6.

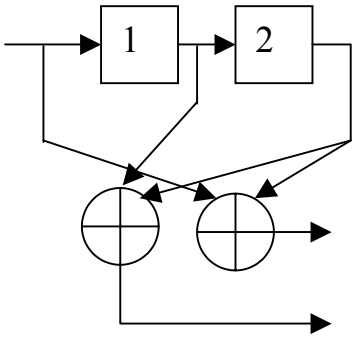


Рис. 1.6. Пример катастрофического кодера

Этому кодеру соответствует конечный автомат, приведенный на рис. 1.7.

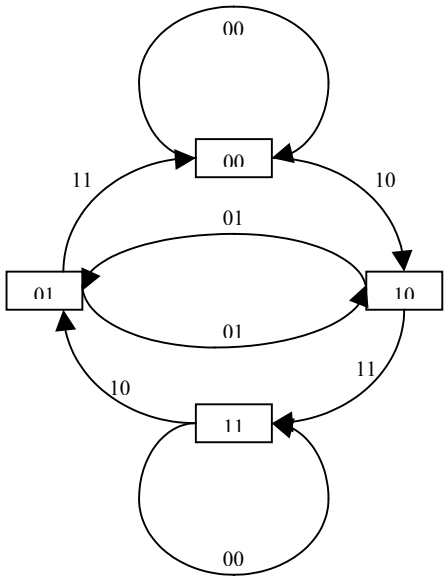


Рис. 1.7. Конечный автомат, описывающий работу кодера кода (5,3)

Воспользовавшись схемой кодера либо диаграммой, легко убедиться, что информационной последовательности бесконечного веса 1111…11100 соответствует кодовая последовательность 10 11 00 00 … 00 10 11 веса 6. При передаче длинной последовательности нулевых информационных символов всего 6 ошибок в канале связи достаточно для того, чтобы вместо нулевого кодового слова было принято записанное выше кодовое слово, содержащее длинную последовательность единиц. Этот пример показывает, что для некоторых сверточных кодеров небольшое число ошибок в канале связи может стать причиной большого числа ошибок декодирования.

Сверточные кодеры, содержащие кодовые слова бесконечного веса, соответствующие информационным последовательностям конечного веса, называются катастрофическими.

Сверточный кодер со скоростью $R = 1/n$ является катастрофическим, если его порождающие полиномы имеют общий делитель отличный от 1.

1.6. Порядок выполнения работы

1. В соответствии с вариантом задания выбрать сверточный кодер и для него представить схему кодера, и представления кода в виде кодового дерева, решетчатой диаграммы, конечного автомата, порождающей матрицы.
2. Написать программу, моделирующую работу кодера и с ее помощью проверить корректность различных способов представления кода.
3. Определить свободное расстояние кода и первые два коэффициента его спектра.

1.7. Контрольные вопросы

1. Пояснить, почему для линейных кодов минимальное расстояние может быть вычислено как минимальный вес кодовых слов.
2. Пусть задано свободное расстояние двоичного сверточного кода. Что можно сказать о гарантированном числе исправляемых и обнаруживаемых кодом ошибок?
3. Может ли свободное расстояние сверточного кода быть больше, чем суммарный вес Хэмминга порождающих векторов?
4. Предположим, что все порождающие многочлены сверточного кода имеют четный вес. Будет ли такой код катастрофическим? Будет ли катастрофическим код, если вес всех порождающих многочленов нечетный?
5. Пусть задана длина кодового ограничения сверточного кода. Каково число узлов на каждом ярусе решетчатой диаграммы? Каково число состояний конечного автомата кодера?
6. Предложите алгоритм для подсчета свободного расстояния кода с помощью компьютера. Как соотносятся сложность вычислений и длина кодового ограничения кода?
7. Рассмотрим код с кодовым ограничением 4 и скоростью ½. Сколько различных кодовых слов может быть получено при кодировании последовательности из 10 информационных символов? Каково суммарное количество узлов в решетке кода, описывающей это множество кодовых слов? Каково общее количество путей в этой решетке? Сколько операций нужно выполнить, чтобы определить, является ли некоторая двоичная последовательность длины 20 кодовым словом? Какова была бы сложность решения этой же задачи для блокового кода?
8. Постройте порождающую матрицу, кодовое дерево, решетку, конечный автомат для кодера на Рис. 1.2б. Определите свободное расстояние кода.