Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный **университет**
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

Институт информационных технологий и радиоэлектроники

Кафедра радиотехники и радиосистем

Полушин Петр Алексеевич

Современные методы кодирования информации

Методические указания к лабораторным работам
по дисциплине «Современные методы кодирования информации» для студентов ВлГУ,
обучающихся по направлению 11.04.01 «Радиотехника»

Владимир – 2018

Оглавление

Правила внутреннего распорядка при проведении лабораторных работ

1. Студенты выполняют лабораторные работы в часы, предусмотренные расписанием, в строгом соответствии с графиком работы лаборатории.
2. Все лабораторные работы выполняются фронтально. Студенты группы разбиваются на рабочие бригады, в составе которых они выполняют все лабораторные работы.
3. Студенты обязаны бережно обращаться с оборудованием и компьютерами, применяемыми при выполнении лабораторных работ, и несут ответственность за порчу или выход из строя приборов и оборудования, происшедших по их вине.
4. Включать компьютеры и приборы можно только с разрешения преподавателя. Неправильное включение может привести к несчастному случаю или порче дорогостоящих измерительных приборов.
5. Обо всех неисправностях студенты обязаны немедленно сообщить преподавателю, предварительно отключив схему.
6. Во время выполнения лабораторной работы необходимо соблюдать тишину, не покидать рабочего места без разрешения преподавателя.
7. Студенты обязаны после окончания работы выключить аппаратуру, привести рабочее место в порядок.
8. Студенты обязаны строго соблюдать указания по технике безопасности для данной лаборатории.

Указания по технике безопасности

К проведению лабораторных работ допускаются студенты, ознакомленные с данным указанием и проинструктированные преподавателем по технике безопасности обращения с приборами, используемыми по теме проводимой работы.

Включение компьютера и работа с ним допускается только после получения от преподавателя разрешения на продолжение работы.

При обнаружении неисправности выключается общий рубильник лабораторного стола.

Первая помощь при поражении током

Необходимо очень быстро освободить пострадавшего от тока, выключив источник питания или отделив пострадавшего от токоведущих частей. При этом (во избежание поражения электрическим током) ни в коем случае не следует браться за тело пострадавшего без резиновых диэлектрических перчаток. Если нельзя выключить ток нормальным путем, нужно разрезать (перерубить) провод или, заземлив его, отвести ток

от пострадавшего. Инструмент, используемый для перерезания провода, должен иметь изолированные ручки. Предмет, набрасываемый на провод, предварительно заземляется. При отсутствии резиновых перчаток и обуви для изоляции может быть использована сухая доска, сухая одежда и т. п.

Если пораженный электрическим током находится высоко над поверхностью пола, необходимо предусмотреть, чтобы после освобождения от действия тока он не упал и не получил повреждений.

В случае потери пострадавшим сознания нужно срочно вызвать скорую помощь и попытаться самим привести его в сознание (брызгать в лицо водой, растирать и согревать его тело). При отсутствии дыхания немедленно начать искусственное дыхание и массаж сердца, не прекращая до прибытия врача. Следует твердо помнить, что первые секунды и минуты являются решающими для возможности вернуть человека к жизни, поэтому искусственное дыхание должны уметь делать все работники лаборатории.

Прежде чем начать делать искусственное дыхание, нужно обеспечить доступ свежего воздуха, освободить пострадавшего от стесняющей одежды, затем положить на спину или живот (в зависимости от выбранного способа искусственного дыхания) и принять меры к тому, чтобы обеспечить прохождение воздуха в легкие. Чтобы язык не западал в дыхательное горло и не закрывал доступ воздуха в легкие, его вытягивают и придерживают с помощью платка или марли.

Существует несколько способов искусственного дыхания: первый способ — пострадавшего положить на живот таким образом, чтобы обеспечить доступ воздуха к его дыхательным путям, встать на колени над его бедрами и попеременно сдавливать ладонями и опускать грудную клетку. Сдавливать грудную клетку нужно в течение трех секунд (по счету «раз, два, три»), постепенно усиливая нажим, подавшись для этого всем туловищем вперед. Затем быстро отнять руку, выждать около трех секунд и снова нажать;

второй способ — скатать из одежды валик и положить под спину пострадавшего, затем встать на колени у его изголовья, захватить обе руки ниже локтей и крепко прижать их на три секунды к бокам. Затем поднять руки пострадавшего и вытянуть их вдоль головы. Такие движения нужно повторять 12—15 раз в минуту;

третий способ — положить пострадавшего на спину, расстегнуть предварительно стесняющую его одежду, запрокинув его голову назад. При таком положении головы воздух свободно проходит в легкие. Надавив на подбородок, раскрыть рот пострадавшего, зажав ему нос рукой. Сделать глубокий вдох и, плотно обхватив своими губами открытый рот пострадавшего, с силой выдохнуть воздух ему в рот. После того как грудная клетка

пострадавшего расширится, отстраниться от него. Выдох у него произойдет произвольно. Вдувание воздуха таким путем производится 16—20 раз в минуту, что соответствует нормальной частоте.

Лабораторное оборудование.

В качестве лабораторного оборудования (макета) используется персональный компьютер.

Методика проведения лабораторных работ

При подготовке к работе студент должен:

1) изучить теоретические положения, на которых данная работа базируется;

2) получить допуск к работе от преподавателя, изложив ему основные теоретические положения, последовательность действий,

3) поставленную программу в среде Matlab,

4) предполагаемые результаты, графики и зависимости.

При удовлетворительном ответе студент получает допуск к работе.

При выполнении работы студент обязан:

1) ввести разработанную программу в компьютер,

2) при необходимости отладить ее,

3) выполнить необходимые исследования, снять графики и зависимости,

4) показать полученные результаты преподавателю.

К следующей лабораторной работе необходимо представить отчет по предыдущей работе. Отчет по работе аккуратно оформляется. Схемы вычерчиваются в соответствии с требованиями ЕСКД, графики выполняются на миллиметровой бумаге. На графике наносятся экспериментальные точки, по ним проводится плавная кривая;

6) полученные зависимости необходимо сравнить со справочными и сделать необходимые выводы и расчеты по проделанной работе.

Содержание отчета по лабораторной работе

1. Наименование и цель работы.

2. Краткие теоретические сведения по тематике работы.

3. Паспортные данные исследуемых приборов (компьютера).
4. Тексты программ.
5. Результаты измерений (в виде таблиц).
6. Полученные графические зависимости.
7. При необходимости сопутствующие измерениям расчеты.
8. Краткие выводы по проделанной работе.

ЛАБОРАТОРНАЯ РАБОТА №1

ИЗУЧЕНИЕ СВЕРТОЧНОГО КОДИРОВАНИЯ

Цель работы

1. Изучить формирование несистематического сверточного кода с кодовой скоростью 1/2.
2. Изучить формирование несистематического сверточного кода с кодовой скоростью 1/3.
3. Изучить формирование систематического сверточного кода.

Краткие теоретические сведения

Сверточные коды в настоящее время нашли широкое применение. Они иногда называются непрерывными кодами, потому что используют непрерывную (последовательную) обработку символов. Кроме того они являются линейными кодами. Кодер обладает памятью в том смысле, что каждый выходной кодовый символ зависит не только от текущего входного информационного символа, но и от нескольких предыдущих входных символов.

Избыточность кода, определяется соотношением количества кодовых символов, требующихся для передачи определенного числа информационных символов. Если для передачи k информационных символов требуется n кодовых символов, то кодовая скорость равна $R=k/n$. Обобщенная структурная схема сверточного кодера приведена на рисунке 1.

На его вход поступает последовательность информационных символов $\mathbf{m}=m_1,m_2,\ldots$ На выходе образуется последовательность кодированных символов $\mathbf{u}=u_1,u_2,\ldots$ Кодер состоит из k сдвиговых регистров, содержащих по K–1 ячеек. Входной коммутатор (Комм.1) распределяет символы входной последовательности по последовательным входам регистров. При каждом такте содержимое регистров сдвигается в соседнюю

ячейку. Сигналы с параллельных выходов регистров подаются на входы n многовходовых сумматоров $n>k$. В сумматорах над входными сигналами производится логическая операция сложения по модулю 2 («исключающее или»). Входы каждого сумматора подключены к своему определенному набору ячеек сдвиговых регистров. Эти наборы различаются у всех сумматоров и определяют структуру конкретного используемого кодера.

Выходная кодовая последовательность образуется последовательным поочередным подключением с помощью коммутатора (Комм.2) выходных сигналов сумматоров на общий выход кодера. Таким образом, при поступлении k исходных информационных символов образуется n кодовых символов. Варьируя параметры k и n, можно регулировать кодовую скорость R. Однако чаще используются коды со скоростью $1/n$, а скорость регулируется применением перфорации (выкалывания).

В формировании каждого кодового символа в данный момент времени участвует входной символ и K–1 предыдущих входных символов. Таким образом кодовое ограничение K определяет память кодовой последовательности. Кодер, приведенный на рисунке 2.1, представляет собой систему с конечным откликом.

Если при k=1 пропускать через кодер исходную последовательность символов, содержащую только одну единицу, а остальные – нули, то формируемая кодовая последовательность будет представлять собой набор импульсных откликов $g_1 \div g_n$ каждого из n сумматоров. Наборы коэффициентов $g_1 \div g_n$ описываются в виде векторов **g**. Размеры векторов составляют максимум K двоичных элементов и полностью определяют структуру кодера. Каждый разряд соответствующего двоичного кода соответствует одному отводу регистра, и единица в нем устанавливает факт наличия связи данного разряда регистра с сумматором. Двоичные последовательности **g** (кодовых генераторов) обычно записывают в десятичной форме или разделяют на группы по три символа и записывают в восьмеричной форме. Также используется запись **g** в виде полинома (порождающего полинома).

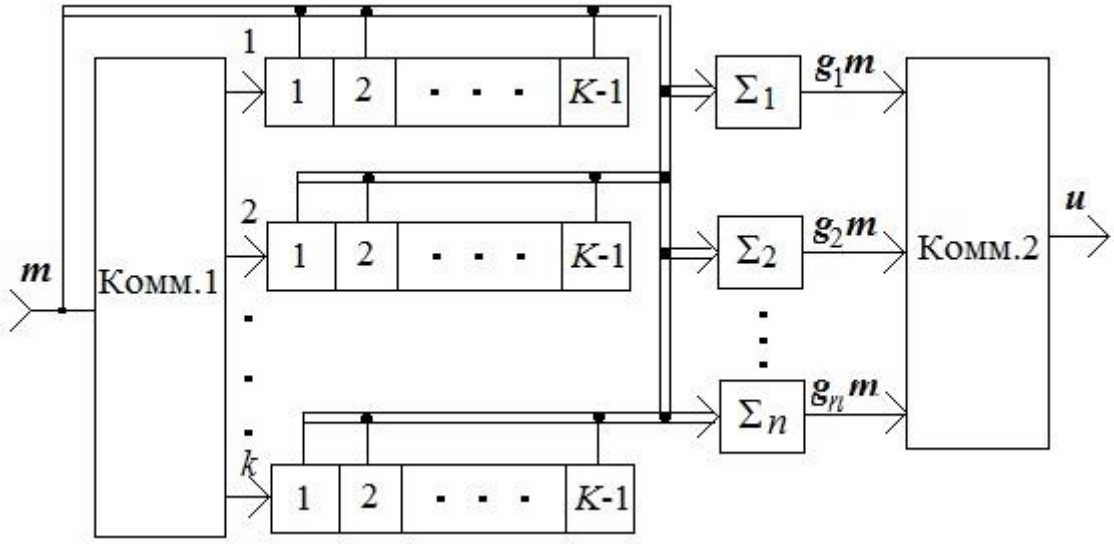


Рисунок 1.

Поскольку выходной сигнал каждого сумматора представляет собой свертку соответствующего порождающего полинома и фрагмента входной информационной последовательности **m**, состоящей и текущего входного символа и K–1 предыдущих входных символов, то формируемые группы по n кодовых символов, соответствующих одному входному символу, можно записать в векторной форме: $\mathbf{u}_0=u_1,\ldots,u_n$, где $u_1=\mathbf{g}_1\mathbf{m}$, $u_2=\mathbf{g}_2\mathbf{m},\ldots,u_n=\mathbf{g}_n\mathbf{m}$ (произведением обозначена свертка векторов). Формируемую кодовую последовательность также можно получить с помощью некоторой матрицы **G** (порождающей матрицы), имеющей ленточную структуру.

Исправляющие свойства сверточных кодов с разными наборами порождающих полиномов различаются. Для удобного графического исследования работы кодера и декодера эффективно использование решетчатых диаграмм. После достижения глубины анализа, равной K шагов, структура диаграммы становится постоянной. Решетчатая диаграмма используется при декодировании. Правильно декодированная кодовая последовательность должна соответствовать на каждом шаге только одному из нескольких разрешенных путей по решетке. Суммарное количество отличий принятой последовательности от разрешенных вариантов, измеренное в определенной метрике, служит указанием для восстановления передаваемого информационного сообщения и освобождения его от ошибок. Для каждого набора исходных данных (кодовой скорости, кодового ограничения) известны наиболее эффективные в смысле исправления ошибок коды, хотя могут использоваться и коды с близкой структурой, несколько уступающие им по эффективности.

Структура систематических сверточных кодеров незначительно отличается от структуры несистематических кодеров, представленной на рисунке 2.1. Отличия заключаются в том, что один из входов выходного коммутатора подключен не к выходу одного из сумматоров по модулю 2, а непосредственно ко входу кодера. В результате одна из компонент формируемого кодовой последовательности совпадает со входной информационной последовательностью, хотя ее символы следуют не один за другим, а разделены n–1 другими кодовыми символами. Систематические сверточные коды используются редко, т.к. их характеристики в среднем хуже, чем у соответствующих несистематических кодов.

Кроме требования высокой эффективности исправления ошибок на вид полиномиальных генераторов накладываются и другие ограничения. Кодовые генераторы удобно представлять в виде полиномов вида:

$$\mathbf{g}(X)=a_0+a_1X+a_2X^2+\ldots+a_nX^n,$$

где переменная X обозначает сдвиг по времени на один такт, X^2 – сдвиг по времени на два такта, и т.д.; коэффициенты $a_0 \div a_n$ могут принимать нулевое или единичное значение. Важное ограничение на вид полиномов связано с возможностью распространения катастрофических ошибок при декодировании, когда конечное число ошибок в кодовых словах вызывает бесконечное число ошибок в декодированных данных.

Возможность для появления катастрофических ошибок появляется, если при разложении используемых порождающих полиномов на более простые полиномы-множители у любых двух из них в разложении будет присутствовать хотя бы один одинаковый простой полином-множитель.

Известны также рекурсивные сверточные коды, которые обычно бывают систематическими. Они отличаются тем, что характеризуются бесконечным импульсным откликом и сумматоры могут находиться не только на выходах, но и на входах регистров. Несистематический кодер и систематический сверточный кодер имеют одно и т же множество кодовых последовательностей, но с другим соответствие между информационными и кодовыми словами.

Методические указания по выполнению

Для исследования выбрать по согласию с преподавателем два сверточных кодера из списка: (5,7), (13,15), (15,17), (25,35), (31,33), (23,35), (71,73). (Обозначения кодеров приведены в восьмеричной системе исчисления).

Составить структурные схемы для формирования выбранных сверточных кодов. После этого, задавая случайную последовательность двоичных символов достаточной длины, аналитически получить вид кодированных последовательностей. Далее в среде Matlab ввести в компьютер программу формирования сверточного кода выбранной структуры и проверить получение каждой кодированной последовательности.

Далее для исследования выбрать по сгласованию с преподавателем один из сверточных кодеров из списка: (5,7,7), (13,15,17), 25,33,37), 47,53,75), (133, 145), (175). Составить структурную схему для формирования выбранного сверточного кода. После этого, задавая случайную последовательность двоичных символов достаточной длины, аналитически получить вид кодированной последовательности. Далее в среде Matlab ввести в компьютер программу формирования сверточного кода выбранной структуры и проверить получение кодированной последовательности.

После этого составить структурную схему систематического сверточного кодера, к качестве дополнительной части кодовой последовательности выбрав любую из использованных восьмеричных цифр и повторить аналогичные действия.

Программа работы

1. Составить и ввести в компьютер программу формирования случайной двоичной последовательности.
2. Составить и ввести в компьютер программу формирования несистематической кодовой последовательности для скорости 1/2.
3. Получить кодированные последовательности для выбранных видов кодера.
4. Составить и ввести в компьютер программу формирования несистематической кодовой последовательности для скорости 1/3.
5. Получить кодированные последовательности для выбранного вида кодера.
6. Составить и ввести в компьютер программу формирования систематической кодовой последовательности.
7. Получить кодированные последовательности для выбранного вида кодера.
8. Повторить пункты 1-7, введя случайную последовательность, которая использовалась при аналитических расчетах и сравнить полученные результаты.
9. Показать полученные результаты преподавателю.

Содержание отчета

1. Наименование и цель работы.
2. Краткие теоретические сведения по тематике работы.

3. Результаты аналитических расчетов.
4. Результаты исследований на компьютере.
5. Выводы по работе.

Контрольные вопросы

1. Что такое сверточный код?
2. В чем при сверточном кодировании проявляется параметрическая избыточность?
3. Как при сверточном кодировании строятся системтические и несистематические коды?
4. Что такое избыточность кода и чему она равна при сверточном кодировании?
5. Какими схемами формируются сверточные коды?
6. В чем отличия схем формирования сверточных кодов со скоростями 1/2 и 1/3.
7. Что такое импульсный отклик сверточного кодера и как его получить?
8. Что такое порождающий полином и что означают его компоненты?
9. Когда возможно появление катастрофических ошибок?
10. Что такое рекурсивные сверточные кодеры?

ЛАБОРАТОРНАЯ РАБОТА №2

ИЗУЧЕНИЕ БЛОКОВЫХ КОДОВ

Цель работы

1. Изучить формирование блоковых кодов с помощью порождающей матрицы.
2. Изучить формирование блоковых кодов с помощью порождающих полиномов.

Краткие теоретические сведения

Виды известных блоковых кодов характеризуются исключительным разнообразием, связанным с особенностями кодирования и декодирования, сложностью реализации, быстродействием и помехоустойчивостью. Однако для целей диагностики большинство методов их получения можно объединить в две связанные между собой группы. В одной из них используются порождающие матрицы, в другой группе – порождающие полиномы.

Любой блоковый код основан на том, что исходная информационная последовательность символов разбивается на группы, как правило, одинаковой длины k. По определенному правилу на основе значений информационных символов в группе

вычисляется последовательность из b проверочных символов и добавляется к информационной последовательности, формируя выходной кодовый блок длиной n.

Если длина информационной части блока невелика, то наиболее простым методом кодирования является использование таблицы соответствия. Таблица соответствия содержит все варианты информационных последовательностей и соответствующие им варианты кодовых блоков. При кодировании на основе каждой новой информационной части для передачи просто выбирается нужный кодовый блок. Однако размер таблицы пропорционален 2^k, поэтому при большой длине информационных частей величина таблицы и время кодирования могут стать недопустимо большими.

В этом случае задачу можно упростить, если не хранить в памяти кодовые блоки, соответствующие поступающим информационным группам, и не отыскивать их в таблице, а каждый раз генерировать вновь. Пусть **m** – вектор, содержащий k двоичных элементов и составленный из группы исходных информационных символов. Тогда, если имеется произвольный базис из k линейно независимых двоичных векторов, тогда любой информационный вектор можно записать, как линейную комбинацию некоторых векторов этого базиса. (При составлении линейной комбинации под умножением понимается операция «и», под сложением – операция «исключающее или»). Таким образом, если имеется набор k двоичных линейно независимых векторов $\mathbf{V}_1$, $\mathbf{V}_2$,…, $\mathbf{V}_k$, каждый из которых состоит из n элементов, то вектор **u**, описывающий любой кодовый блок, можно записать, как: $\mathbf{u}=m_1\mathbf{V}_1+m_2\mathbf{V}_2+\ldots+m_k\mathbf{V}_k$.

Порождающая матрица **G** имеет размер $k\times n$ и представляет собой набор векторов $\mathbf{V}_1 \div \mathbf{V}_k$, как строк:

$$\mathbf{G} = \begin{Vmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_k \end{Vmatrix} = \begin{Vmatrix} v_{1,1}, v_{1,2}, \ldots, v_{1,n} \\ v_{2,1}, v_{2,2}, \ldots, v_{2,n} \\ \ldots\ldots\ldots \\ v_{k,1}, v_{k,2}, \ldots, v_{k,n} \end{Vmatrix},$$

где $v_{i,j}$ – элементы с индексом j вектора $\mathbf{V}_i$.

Генерация кодового слова в матричной форме описывается уравнением $\mathbf{u}=\mathbf{mG}$. При этом при кодировании в памяти необходимо сохранять только k векторов $\mathbf{V}_1 \div \mathbf{V}_k$, а не 2^k, как при использовании таблицы соответствия.

Матрицу **G** можно представить, как состоящую из двух блоков, первый блок $\mathbf{G}_1$ размером $k\times k$, второй блок $\mathbf{G}_2$ размером $k\times(n–k)$, т.е.: $\mathbf{G} = \|\mathbf{G}_1 \vdots \mathbf{G}_2\|$. Матрица $\mathbf{G}_1$ определяет вид информационной части кодового слова, матрица $\mathbf{G}_2$ определяет вид

проверочной части кодового слова. При использовании несистематического кодирования вид матрицы $\mathbf{G}_1$ может быть произвольным (при сохранении условия линейной независимости строк). При использовании систематического кодирования $\mathbf{G}_1$ является единичной матрицей, $\mathbf{G}_1=\mathbf{E}$. Несистематическое кодирование обеспечивает такие же характеристики помехоустойчивости, как и систематическое, однако при использовании систематического кодирования процесс кодирования существенно упрощается и ускоряется ([ ]), поэтому практически используется систематическое кодирование. К тому же порождающая матрица для несистематического кодирования легко преобразуется в матрицу для систематического кодирования умножением на обратную матрицу $\mathbf{G}_1^{-1}$.

После переобозначения части матрицы, отвечающей за формирование проверочной части кодового слова, $\mathbf{G}_2=\mathbf{P}$, порождающая матрица приобретает вид: $\mathbf{G} = \|\mathbf{E} \vdots \mathbf{P}\|$. При подробном рассмотрении структуры генерируемого кодового слова видно, что каждый проверочный бит имеет свое происхождение и является «индивидуальной» комбинацией каких-либо информационных битов. При этом очевидно, что по сравнению с контролем четности методом дублирования разряда или применением одного бита четности, применяемый метод обеспечивает более широкие возможности для исправления возникающих при передаче ошибок.

Циклические коды представляют класс блоковых кодов, кодирование и декодирование которых основано на использовании полиномиальных представлений, более простых, чем матричные процедуры. Широкое распространение этот класс кодов получил в связи с удобством практической реализации на основе достаточно простой современной цифровой элементной базы. Линейный код называется циклическим, если при любом циклическом сдвиге некоторого кодового слова получается другое кодовое слово, которое может быть получено тем же кодером, но из другой входной информационной группы.

Значения символов кодового слова можно рассматривать, как коэффициенты полинома: $\mathbf{u}(X)=u_0+u_1X+u_2X^2+\ldots+u_{n-1}X^{n-1}$. Наличие или отсутствие каких-либо членов в полиноме означает наличие нулей в соответствующих разрядах кодового слова. Если $u_{n-1}\neq 0$, то порядок полинома равен $n-1$.

Циклический код создается с помощью полиномиального генератора $\mathbf{g}(X)$. Для заданного циклического кода (n, k) вид полиномиального генератора является единственным:

$$\mathbf{g}(X)=g_0+g_1X+g_2X^2+\ldots+g_bX^b\ .$$

Предполагается, что коэффициенты g_0 и g_b – ненулевые, $b=n–k$. Каждый полином, описывающий какое-либо кодовое слово, имеет вид $\mathbf{u}(X)=\mathbf{m}(X)\mathbf{g}(X)$, т.е. кодовое последовательность двоичных символов действительно является кодовым словом, сформированным данным кодером, если оно делится без остатка на $\mathbf{g}(X)$. Полиномиальный генератор является одним простым множителем или произведением нескольких простых множителей полинома X^n+1. Использование циклического кодирования в описанном виде формирует несистематические коды, т.е. в кодовом слове $\mathbf{u}(X)$ нет фрагмента, все символы которого совпадали бы с вектором $\mathbf{m}(X)=m_0+m_1X+m_2X^2+m_3X^3+\ldots+m_{k-1}X^{k-1}$.

Однако чаще используется систематическая форма кодов, упрощающая процедуры обработки. При систематическом кодировании вектор $\mathbf{m}(X)$ совпадает с фрагментом выходного кодового слова $\mathbf{u}(X)$ и используется как его часть. Для этого символы информационного сообщения первоначально сдвигаются в сторону бóльших степеней X на $n–k$ позиций с помощью умножения на X^{n-k}. При этом формируется полином:

$$X^{n-k}\,\mathbf{m}(X)= m_0X^{n-k}+ m_1X^{n-k+1}+m_2X^{n-k+2}+m_3X^{n-k+3}+\ldots+m_{k-1}X^{n-1}.$$

Далее он делится на выбранный порождающий полином $\mathbf{g}(X)$. Результат деления можно записать в виде:

$$X^{n-k}\,\mathbf{m}(X)=\mathbf{q}(X)\mathbf{m}(X)+\mathbf{p}(X), \qquad (3.1)$$

где $\mathbf{q}(X)$ – частное от деления; $\mathbf{p}(X)$– остаток от деления, равный:

$$\mathbf{p}(X)=p0+p1X+p2X^2+\ldots+p^{n-k-1}X^{n-k-1}.$$

Остаток от деления суммируется по модулю 2 с обеими частями уравнения (3.1), в результате получается полином:

$$\mathbf{u}(X)=X^{n-k}\,\mathbf{m}(X)+\mathbf{p}(X)=\mathbf{q}(X)\mathbf{m}(X).$$

Полученный таким образом полином $\mathbf{u}(X)$ является кодовым словом, поскольку делится на $\mathbf{g}(X)$, но его фрагмент из k символов, стоящий в области больших степеней, теперь уже совпадает с информационным вектором $\mathbf{m}(X)$, а фрагмент, стоящий в области $n–k$ меньших степеней, является проверочной частью кодового слова, т.е.:

$$\mathbf{u}(X)=p_0+p_1X+p_2X^2+\ldots+p_{n-k-1}X^{n-k-1}+m_0X^{n-k}+m_1X^{n-k+1}+\ldots+m_{k-1}X^{n-1}.$$

Процедура кодирования иллюстрируется структурной схемой, приведенной на рисунке 2. Схема состоит из последовательно включенных элементов памяти $Э_1 \div Э_{n-k-1}$. Фактически она представляет собой регистр сдвига с обратными связями. В момент прихода тактового импульса каждый элемент памяти запоминает значение символа на своем входе и хранит его до поступления следующего тактового импульса. Между ними находятся сумматоры по модулю 2. На их входы поступают выходные сигналы с элементов памяти и с элементов $g_1 \div g_{n-k-1}$. Если коэффициент $g_1 \div g_{n-k-1}$. равен единице, то выходной сигнал ключа (Кл.) подается на соответствующий сумматор, если равен нулю, то суммирование не производится.

Схема работает следующим образом. Моменты появления тактовых импульсов совпадают с появлением информационных символов. При первых символах последовательности $\mathbf{m}(X)$ ключ закрыт, а коммутатор (Комм.) подключает на выход сигнал со своего первого входа, т.е. непосредственно исходную информационную последовательность. После передачи k-го информационного символа ключ открывается, а коммутатор на свой выход подключает сигнал со второго входа. В следующих $n–k$ тактах происходит формирование проверочных символов. После окончания n тактов обработки в выходном регистре (Р) оказываются записаны все сформированные символы кодового слова, которые далее поступают в передатчик. Для повышения скорости работы кодера иногда используют комбинацию табличного метода и регистра с обратными связями.

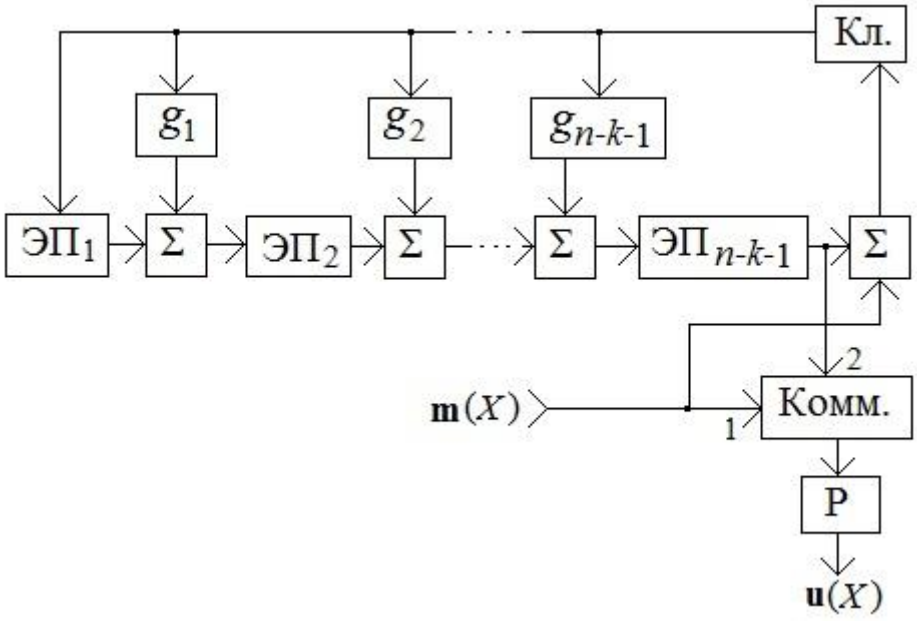


Рисунок 2.

Методические указания по выполнению

Выбрать два варианта порождающей матрицы с параметрами k не менее, чем 5, и b не менее, чем 2, и до проведения лабораторного занятия согласовать их вид с преподавателем. Выбрав вид случайной двоичной последовательности, аналитически произвести ее кодирование с использованием выбранных порождающих матриц.

Также заранее выбрать два варианта порождающего полинома с параметром k не меньше, чем 5. Выбрав вид случайной двоичной последовательности, аналитически произвести ее кодирование с использованием выбранных порождающих полиномов.

Составить программы в среде Matlab для формирования блоковых кодов с помощью порождающих матриц и с помощью порождающих полиномов и с помощью компьютера сформировать кодовые последовательности.

Программа работы

1.Составить и ввести в компьютер программу формирования случайной двоичной последовательности.

2.Составить и ввести в компьютер программу формирования блокового кода с помощью порождающей матрицы

3. Получить кодированные последовательности для выбранных видов матрицы.

4. Составить и ввести в компьютер программу формирования блокового кода с помощью порождающего полинома.
5. Получить кодированные последовательности для выбранных видов порождающего полинома.
6. Повторить пункты 1-5, введя случайную последовательность, которая использовалась при аналитических расчетах и сравнить полученные результаты.
7. Показать полученные результаты преподавателю.

Содержание отчета

1. Наименование и цель работы.
2. Краткие теоретические сведения по тематике работы.
3. Результаты аналитических расчетов.
4. Результаты исследований на компьютере.
5. Выводы по работе.

Контрольные вопросы

1. Что такое блоковые коды?
2. Основные отличия блоковых кодов от циклических кодов.
3. В чем проявляется избыточность при блоковом кодировании и чем она измеряется?
4. Какими способами можно сформировать блоковый код.
5. Описать структуру порождающей матрицы.
6. Разница в порождающих матрицах для систематического и для несистематического кодирования.
7. В чем преимущества циклических блоковых кодов?
8. Как формируется циклический блоковый код с помощью пороэждающего полинома?
9. В чем упрощение алгоритма получения систематического циклического блокового кода?
10. Описать принцип работы структурной схемы для формирования систематического циклического кода.

ЛАБОРАТОРНАЯ РАБОТА №3

ИЗУЧЕНИЕ МОДИФИЦИРОВАННЫХ КОДОВ

Цель работы

1. Изучить формирование укороченных кодов с помощью порождающего полинома.
2. Изучить формирование укороченных кодов с помощью порождающей матрицы.
3. Изучить формирование расширенных кодов.

Краткие теоретические сведения

Модифицированные коды строятся на основе некоторого исходного кода, к которому либо добавляют дополнительные символы (расширенный код), либо сокращают часть информационных символов без изменения расстояния между кодовыми символами (укороченный код), либо выбрасывают часть символов (перфорация или выкалывание). Таким образом, достигается определенная гибкость в получении нужных параметров передачи информации. Например, одной из ситуаций, когда эффективно укорочение, является потребность применить код Хэмминга, но при этом требуется число кодовых символов не равное 2^m, m – целое.

Укорочение кодов. Укорочение производится отбрасыванием определенного числа *s* позиций в информационной части кодового слова (*s* – глубина укорочения). В слове на передачу информации теперь отводится только *k*–*s* символов, но длина проверочной части *n*–*k* остается прежней. Укороченное кодовое слово формируется посредством того, что в некоторых фиксированных позициях информационной части исходного (неукороченного) кодового слова устанавливаются нули. Остальные позиции могут принимать произвольные значения в зависимости от информации, передаваемой в этом кодовом слове. Положение нулевых позиций принципиального значения не имеет, однако для практического удобства кодирования и декодирования обычно выбирается *s* старших позиций. Таким образом, информационная часть кодового слова приобретает вид:

$$\mathbf{m}(X)=m_0+ m_1X+m_2X^2+m_3X^3+\ldots+m_{k-1}X^{k-1-s}.$$

Далее оно также систематическим кодером сдвигается на *n*–*k* разрядов, делится на порождающий полином и остаток от деления записывается в младшие пустые разряды. Образуется укороченный линейный код, степень которого не превышает *n*–*s*–1. В общем случае такой код не остается циклическим.

Если для кодирования используется порождающая матрица $\mathbf{G} = \|\mathbf{E} \vdots \mathbf{P}\|$, то она также должна быть модифицирована. Для этого из нее удаляются s выбранных столбцов и одновременно s строк, соответствующих ненулевым элементам удаленных столбцов.

Пусть исходную матрицу можно записать в виде:

$$\mathbf{G} = \begin{Vmatrix} 1,0,0,\ldots..,0, g_{1,k+1}, g_{1,k+2},\ldots.., g_{1,n} \\ 0,1,0,\ldots..,0, g_{2,k+1}, g_{2,k+2},\ldots.., g_{2,n} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ 0,0,0,\ldots..,1, g_{k,k+1}, g_{k,k+2},\ldots..g_{k,n} \end{Vmatrix} = \begin{Vmatrix} 1,0,0,\ldots..,0, p_{1,k+1}, p_{1,k+2},\ldots.., p_{1,n} \\ 0,1,0,\ldots..,0, p_{2,k+1}, p_{2,k+2},\ldots.., p_{2,n} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ 0,0,0,\ldots..,1, p_{k,k+1}, p_{k,k+2},\ldots..p_{k,n} \end{Vmatrix}.$$

Если при укорочении удаляется *s* первых столбцов (и соответственно, *s* первых строк), то размер новой порождающей матрицы становится равным (*k*–*s*)×(*n*–*s*). Единичная информационная матрица становится размером (*k*–*s*)×(*k*–*s*). Элементы новой укороченной проверочной матрицы: $g_{i,j} = p_{i-s,j-s}$. Операция укорочения уменьшает длину информационной части, в то время как число проверочных символов остается прежним. В результате может быть исправлено большее число комбинаций ошибок, и помехоустойчивость передачи информации возрастает. Таким образом, укороченные коды более эффективны, чем исходные неукороченные коды.

Важное свойство укороченных кодов состоит в том, что для работы с ними могут быть использованы те же самые кодеры и декодеры, что и для неукороченных кодов. Однако для них обычно не выполняется правило циклического сдвига. При практической реализации слова обычно дополняются нулями в откинутых старших разрядах и используются те же самые алгоритмы кодирования и декодирования. А при передаче сообщений включенные нули не передаются.

Расширение кодов. Расширение кодов означает добавление в кодовое слово проверочных символов при сохранении в нем количества информационных символов. В результате получается код (n+s,k). Расширенная порождающая матрица может быть получена из исходной матрицы добавлением к ней соответствующих s столбцов.

Ее размер становится равным $k\times(n+s)$. Если вводимые новые элементы обозначить через $q_{i,j}$, то расширенная матрица приобретает вид:

$$\mathbf{G} = \begin{Vmatrix} 1,0,0,\ldots,0, g_{1,k+1}, g_{1,k+2},\ldots\ldots, g_{1,n}, q_{1,n+1},\ldots, q_{1,n+s} \\ 0,1,0,\ldots,0, g_{2,k+1}, g_{2,k+2},\ldots\ldots, g_{2,n}, q_{2,n+1},\ldots, q_{2,n+s} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0,0,0,\ldots,1, g_{k,k+1}, g_{k,k+2},\ldots\ldots g_{k,n}, q_{k,n+1},\ldots, q_{k,n+s} \end{Vmatrix}.$$

Расширение кодов также увеличивает помехоустойчивость и изменяет кодовую скорость.

Иногда рассматривается объединенная операция расширения, модифицирующая исходный код и состоящая в добавлении в порождающую матрицу и новых строк, и новых столбцов. Ее размер становится равным $(k+s)\times(n+s)$. При этом ее часть, имеющая вид единичной матрицы, занимает не одну из половин, а меньшую ее часть. Матрица приобретает вид:

$$\mathbf{G} = \begin{Vmatrix} 1,0,0,\ldots,0, g_{1,k+1}, g_{1,k+2},\ldots\ldots, g_{1,n}, q_{1,n+1},\ldots, q_{1,n+s} \\ 0,1,0,\ldots,0, g_{2,k+1}, g_{2,k+2},\ldots\ldots, g_{2,n}, q_{2,n+1},\ldots, q_{2,n+s} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ 0,0,0,\ldots,1, g_{k,k+1}, g_{k,k+2},\ldots\ldots g_{k,n}, q_{k,n+1},\ldots, q_{k,n+s} \\ q_{k+1,1}, q_{k+1,2}, q_{k+1,3},\ldots\ldots\ldots\ldots\ldots\ldots\ldots, q_{k+1,n+s} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ q_{k+s,1}, q_{k+s,2}, q_{k+s,3},\ldots\ldots\ldots\ldots\ldots\ldots\ldots, q_{k+s,n+s} \end{Vmatrix}.$$

Кроме рассмотренных вариантов модификации известны также варианты, когда к множеству всех возможных кодовых слов добавляются новые кодовые слова (операция пополнения) или из этого множества удаляются некоторые кодовые слова (операция выбрасывания). Для осуществления операции пополнения при сохранении линейности кода в порождающую матрицу добавляются новые линейно независимые строки.

Операция выбрасывания в общем случае приводит к нелинейному коду. Для сохранения линейности возможно удалить часть строк исходной порождающей матрицы. Для систематических кодов операции выбрасывания и укорочения совпадают.

Методические указания по выполнению

Заранее выбрать два варианта порождающего полинома с параметром k не меньше, чем 6, и до проведения лабораторного занятия согласовать их вид с преподавателем.. Выбрав вид случайной двоичной последовательности, аналитически произвести ее кодирование с использованием выбранных порождающих полиномов. После этого сформировать алгоритм укорочения с величиной s=2 и повторить кодирование случайной двоичной последовательности этим укороченным кодом.

Также выбрать два варианта порождающей матрицы с параметрами k не менее, чем 6, и b не менее, чем 2, и с величиной s=2. Выбрав вид случайной двоичной последовательности, аналитически произвести ее кодирование с использованием выбранных исходных и неукороченных порождающих матриц.

На основе исходных порождающих матриц сформировать расширенные матрицы и повторить аналитическим способом кодирование случайной двоичной последовательности.

Составить программы в среде Matlab для формирования расширенных и укороченных кодов с помощью порождающих полиномов и с помощью порождающих матриц и с помощью компьютера сформировать кодовые последовательности.

Программа работы

1. Составить и ввести в компьютер программу формирования случайной двоичной последовательности.
2. Составить и ввести в компьютер программу формирования блокового кода с помощью порождающего полинома и порождающей матрицы.
3. Получить кодированные последовательности для выбранных видов матрицы.
4. Составить и ввести в компьютер программу формирования расширенного блокового кода.

5. Получить кодированные последовательности для выбранных видов порождающего полинома.

6. Повторить пункты 1-5, введя случайную последовательность, которая использовалась при аналитических расчетах и сравнить полученные результаты.

7. Показать полученные результаты преподавателю.

Содержание отчета

1.Наименование и цель работы.

2.Краткие теоретические сведения по тематике работы.

3.Результаты аналитических расчетов.

4.Результаты исследований на компьютере.

5.Выводы по работе.

Контрольные вопросы

1. С какой целью производится модификация кодов?
2. Как формируются укороченные коды с использованием порождающего полинома?
3. Как формируются укороченные коды с использованием порождающей матрицы?
4. Как декодируются укороченные коды?
5. В чем заключается расширение кодов?
6. Как реализуются расширенные коды?
7. Какие еще варианты существуют для получения расширенных кодов?
8. Как изменяются характеристики модифицированных кодов (цикличность, линейность)?

ЛАБОРАТОРНАЯ РАБОТА №4

ИЗУЧЕНИЕ ПЕРФОРИРОВАННЫХ КОДОВ

Цель работы

1. Изучить метод перфорации сверточных кодов.
2. Изучить метод перфорации блоковых кодов

Краткие теоретические сведения

Перфорация сверточных кодов применяется в основном для сверточных кодов и состоит в том, что из процесса передачи в канал удаляются некоторые символы с выхода кодера. Поскольку при этом структура собственно кодера не меняется, то и количество информационных символов не меняется. В результате формируется перфорированный код с более высокой кодовой скоростью.

Правило удаления символов задается, как правило, матрицей (маской) перфорации. Матрица перфорации P задает правило удаления выходных символов. Обычно процесс перфорации является периодическим и сходный (неперфорированный) код имеет скорость $R=1/n$, n – целое. В этом случае количество N_p кодовых символов, через которое процесс повторяется, должно быть кратно n.

Элементы матрицы P равны нулю или единице. Ноль указывает, что соответствующий двоичный символ будет удален, единица – что он будет оставлен. Размеры матрицы P равны $n \times N_p$. Если она содержит q нулей, то кодовая скорость после перфорации станет равной $R=N_p/(nN_p-q)$.

Формирование перфорированного кода производится следующим образом. После прихода на исходный кодер N_p информационных символов он вырабатывает N_p групп кодовых символов, каждая из которых соответствует одному из исходных символов. Эти группы можно записать в виде векторов размера n. Далее символы каждого из векторов соотносятся с соответствующим столбцом маски перфорации. Если элемент маски равен нулю, то символ с таким номером из вектора удаляется. (После этого размеры векторов становятся неравными.) Всего будет удалено q символов. Далее из оставшихся элементов всех векторов последовательно составляется перфорированная последовательность длиной nN_p-q. После этого аналогично производится следующий период обработки.

При записи структуры перфорированного кода правило получения каждого кодового символа как обычно записывается в форме восьмеричного кода, а место перфорированных символов обозначается буквой X. Перфорация используется достаточно широко, поскольку для различных вариантов перфорированного кода, полученных из одного и того же исходного кода, можно использовать один и тот же декодер, предназначенный для исходного кода.

Перфорация применяется также и к линейным блоковым кодам. Она состоит в удалении из кодового слова p проверочных символов. Код при этом остается линейным блоковым, количество информационных символов k в кодовом слове не изменяется, а общее количество кодовых символов уменьшается до $n-p$. Избыточность уменьшается, а скорость кода возрастает.

Методические указания по выполнению

Заранее выбрать два варианта порождающего полинома для формирования сверточного кода и до проведения лабораторного занятия согласовать их вид с преподавателем. Выбрав вид случайной двоичной последовательности, аналитически произвести ее кодирование с использованием выбранных порождающих полиномов.

После этого сформировать алгоритм перфорации и повторить кодирование случайной двоичной последовательности этим укороченным кодом.

Возможные структуры перфорации могут иметь вид:

$\begin{pmatrix}7,7\\5,X\end{pmatrix}$, $\begin{pmatrix}17,X\\15,15\end{pmatrix}$, $\begin{pmatrix}37,37\\25,X\end{pmatrix}$, $\begin{pmatrix}33,X,X\\31,31,31\end{pmatrix}$, $\begin{pmatrix}37,37,37\\25,X,X\end{pmatrix}$.

Также выбрать два варианта порождающей матрицы и произвести перфорацию. Выбрав вид случайной двоичной последовательности, аналитически произвести ее кодирование с использованием выбранных исходных и перфорированных порождающих полиномов и матриц.

Составить программы в среде Matlab для формирования перфорированных кодов и с помощью компьютера сформировать кодовые последовательности.

Программа работы

1. Составить и ввести в компьютер программу формирования случайной двоичной последовательности.
2. Составить и ввести в компьютер программу формирования перфорированного сверточного кода.
3. Получить кодированные последовательности.
4. Составить и ввести в компьютер программу формирования перфорированного блокового кода.
5. Получить кодированные последовательности.
6. Повторить пункты 1-5, введя случайную последовательность, которая использовалась при аналитических расчетах и сравнить полученные результаты.
7. Показать полученные результаты преподавателю.

Содержание отчета

1.Наименование и цель работы.

2.Краткие теоретические сведения по тематике работы.

3.Результаты аналитических расчетов.

4.Результаты исследований на компьютере.

5.Выводы по работе

Контрольные вопросы

1. В чем сущность алгоритмов перфорации?
2. Какая цель достигается с помощью перфорации кодов?

3. Что такое «маска перфорации»? Привести примеры.
4. Как осуществляется перфорация сверточных кодов?
5. Как осуществляется декодирование перфорированных кодов?
6. Как осуществляется перфорация блоковых кодов?

ЛАБОРАТОРНАЯ РАБОТА №5

ИЗУЧЕНИЕ КОМБИНИРОВАННЫХ КОДОВ

Цель работы

1. Изучить формирование последовательного комбинирования кодов.
2. Изучить формирование кодов произведения.
3. Изучить формирование каскадных кодов.

Краткие теоретические сведения

Комбинированные коды. Известны различные способы комбинирования кодов. Рассмотрим метод последовательного соединения (concatenation – конкатенация). Если используются блоковые коды C_1 и C_2, то их последовательное соединение эквивалентно поочередной передаче кодовых слов, полученных с помощью первого кода и второго кода. В результате последовательного соединения $i=1\div m$ кодов образуется комбинированный код с параметрами n_0 и k_0:

$$n_0 = \sum_{i=1}^{m} n_i \ , \ \ k_0 = \sum_{i=1}^{m} k_i \ ,$$

где n_i и k_i – параметры i-го кода. Если $\mathbf{G}_i$ – исходные порождающие матрицы кодов, составляющих последовательное соединение, то его порождающая матрица $\mathbf{G}_0$ последовательного соединения кодов будет равна:

$$\mathbf{G}_0 = \begin{Vmatrix} \mathbf{G}_1,0,0,....,0 \\ 0,\mathbf{G}_2,0,....,0 \\ ..... \\ o,o,o,.....,\mathbf{G}_m \end{Vmatrix}.$$

Величины n_i и k_i у объединяемых кодов чаще не совпадают, т.к. последовательно соединенные коды используют в случае необходимости реализации разного уровня защиты от ошибок. Если же все n_i и k_i одинаковы, это эквивалентно m-кратной повторной передаче одного кодового слова.

Другим типом комбинированных кодов являются *произведения кодов*. В простейшем виде они представляют собой последовательное (повторное) кодирование. При этом выход первого кодера является входом второго кодера, и т.д. (рисунок 7.1.). Несмотря на простоту, метод формирует достаточно помехоустойчивые коды, особенно для кодов с низкой кодовой скоростью. Если используются два кодера, то код первого из них обычно называется внешним кодом, второй – внутренним кодом.

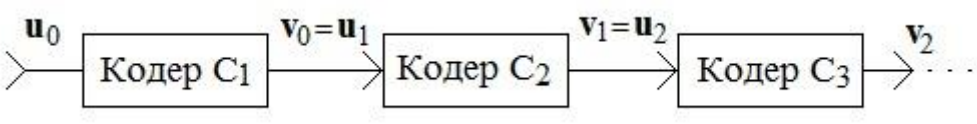


Рисунок 7.1.

Если **G** и **H** – порождающие матрицы двух кодов с размерами $m \times n$ и $p \times r$, то порождающая **Q** матрица произведения этих кодов есть кронекерово произведение этих матриц, т.е.:

$$\mathbf{Q} = \begin{Vmatrix} g_{1,1}\mathbf{H}, g_{1,2}\mathbf{H}, \ldots, g_{1,n}\mathbf{H} \\ g_{2,1}\mathbf{H}, g_{2,2}\mathbf{H}, \ldots, g_{2,n}\mathbf{H} \\ \ldots\ldots\ldots\ldots\ldots \\ g_{m,1}\mathbf{H}, g_{m,2}\mathbf{H}, \ldots, g_{m,n}\mathbf{H} \end{Vmatrix},$$

где gij– элементы матрицы **G**. Матрица **Q** имеет размер $mp \times nr$. Если параметры исходных кодов равны (n_1,k_1) и (n_2,k_2), то при генерации такого кода кодовые слова длиной n_1 символов, уже закодированные внешним кодом, записывают по строкам прямоугольного массива с n_1 столбцами, размещая один символ в один элемент массива. Так заполняют k_2 строк, а оставшиеся $n_2–k_2$ строк заполняются проверочными символами внутреннего кода. При этом k_2 элементов каждого столбца считаются информационными символами внутреннего кода. Общее кодовое слово полученного кода передается из массива по столбцам.

Каскадные коды. В каскадных кодах чаще всего используются в качестве внешнего кода недвоичные блоковые коды Рида-Соломона. Символы полученных кодовых слов подвергаются перемежению, в результате чего порядок следования символов изменяется. Далее они подвергаются кодированию с помощью внутреннего кодера, который может быть как блоковым, так и сверточным. Если внешний, и внутренний коды C_1 и C_2 имеют

параметры (n_1, k_1) и (n_2, k_2), то параметры сформированного каскадного кода равны (n_1n_2, k_1k_2) . Когда внутренний код – двоичный линейный блоковый, тогда и полученный каскадный код также является двоичным линейным блоковым.

Были предложены обобщенные каскадные коды, как усложнение каскадных кодов. Они способны исправлять как случайные ошибки, так и случайные пакеты ошибок. Обобщение заключается в том, что вводится разбиение внутреннего кода на подкоды со своей иерархией, а также используется несколько внешних кодов в соответствии с иерархией. При этом если необходимо, то с помощью выбора параметров внутренних и внешних кодов легко получается блоковый или сверточный код с неравной защитой от ошибок. Сообщения, закодированные на верхнем уровне иерархии, имеют усиленную корректирующую способность по сравнению с нижними уровнями.

Методические указания по выполнению

Заранее выбрать вариант сверточного кода и вариант блокового кода. (можно использовать варианты, примененные в предыдущих лабораторных работах.) и до проведения лабораторного занятия согласовать их вид с преподавателем.

Сформировать алгоритм последовательного комбинирования кодов, алгоритм произведения кодов и алгоритм каскадирования кодов. (при этом можно использовать любой вид перемежающего алгоритма). Выбрав вид случайной двоичной последовательности, аналитически произвести ее кодирование с использованием сформированных алгоритмов.

Составить программы в среде Matlab для формирования всех трех видов комбинированных кодов и с помощью компьютера сформировать кодовые последовательности.

Программа работы

1. Составить и ввести в компьютер программу формирования случайной двоичной последовательности.
2. Составить и ввести в компьютер программу формирования алгоритма последовательного комбинирования кодов.
3. Получить кодированные последовательности.
4. Составить и ввести в компьютер программу формирования алгоритма произведения кодов.
5. Получить кодированные последовательности.

6. Составить и ввести в компьютер программу формирования алгоритма произведения кодов.

7. Получить кодированные последовательности.

8. Составить и ввести в компьютер программу формирования алгоритма каскадирования кодов.

9. Получить кодированные последовательности.

10. Повторить пункты 1-9, введя случайную последовательность, которая использовалась при аналитических расчетах, и сравнить полученные результаты.

11. Показать полученные результаты преподавателю.

Содержание отчета

1. Наименование и цель работы.
2. Краткие теоретические сведения по тематике работы.
3. Результаты аналитических расчетов.
4. Результаты исследований на компьютере.
5. Выводы по работе

Контрольные вопросы

1. С какой целью производится комбинирование кодов?
2. В чем заключается метод последовательного комбинирования кодов?
3. К каким видам кодов может применяться метод последовательного комбинирования кодов?
4. В чем заключатся метод произведения кодов?
5. Что такое внешний код и внутренний код?
6. Какой вид принимает порождающая матрица после произведения кодов?
7. Что такое каскадные коды?
8. Что дает использование каскадных кодов?

ЛАБОРАТОРНАЯ РАБОТА №6

ИЗУЧЕНИЕ МЕТОДОВ ДЕКОДИРОВАНИЯ СВЕРТОЧНЫХ КОДОВ

Цель работы

1. Изучение алгоритма Витерби для декодирования сверточных кодов.

Краткие теоретические сведения

Существует несколько способов декодирования сверточных кодов, среди которых наиболее часто используется алгоритм декодирования Витерби, который работает с решетчатым представлением и отбрасывает пути, заранее не соответствующие критерию максимального правдоподобия. Работа ведется только с выжившими путями. При этом данный алгоритм работает в соответствии с принципом максимального правдоподобия.

Алгоритм декодирования Витерби был разработан в 1967 г. с целью уменьшения объема вычислений по сравнению с алгоритмом последовательного декодирования. В 1969 г. было показано, что данный алгоритм основывается на оценке максимального правдоподобия.

Главное достоинство алгоритма Витерби заключается в том, что в нем не рассматриваются пути, которые согласно принципу максимального правдоподобия не могут быть оптимальными. Алгоритм включает в себя операции вычисления расстояния между принятым сигналом в момент времени t_1, и всеми путями решетки, которые входят в каждое состояние в момент времени t_i. Если в одно состояние входят два пути, то выбирается выживающий путь с наименьшей метрикой. В результате работы декодер постепенно проходит решетку и исключает наименее вероятные пути.

Рассмотрим работу алгоритма Витерби на примере. В качестве меры расстояния используем метрику Хэмминга. Воспользуемся кодером, изображенным на рисунке 11.1 и соответствующей ему решетчатой диаграммой, изображенной на рисунке 11.3.

Предположим, что мы имеем входную информационную последовательность m=10010. После кодирования ее сверточным кодером получаем последовательность U=1110111110, которая передается по каналу связи. В результате воздействия шума принятая последовательность будет иметь вид Z=1110011110, то есть имеет место искажение одного символа, а именно пятого бита.

На рисунке 3 представлена решетчатая диаграмма, в которой над каждой ветвью обозначено расстояние Хэмминга между принятым кодовым символом и кодовым словом, соответствующем данной ветви.

Рассмотрим решетку в момент времени t_1. Переход между состояниями 00→00 приводит к появлению на выходе кодового слова 00, но получено 11, следовательно, Хэммингово расстояние равно 2. Переход между состояниями 00→10 приводит к появлению на выходе кодового слова 11, что полностью совпадает с полученной последовательностью, и, следовательно, Хэммингово расстояние равно 0. Таким образом помечается вся решетка в последующие моменты времени.

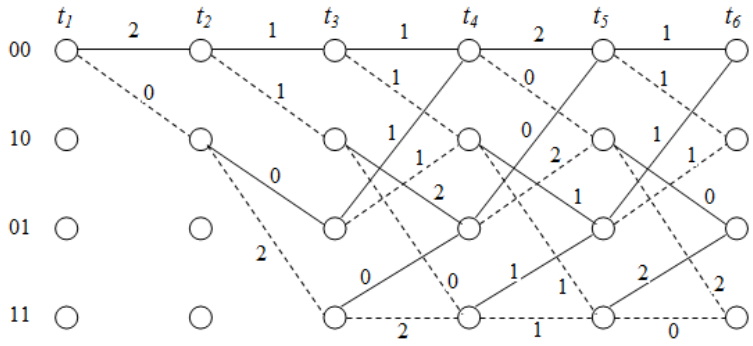
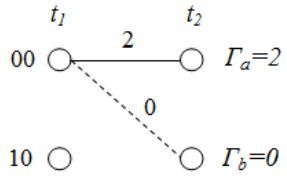


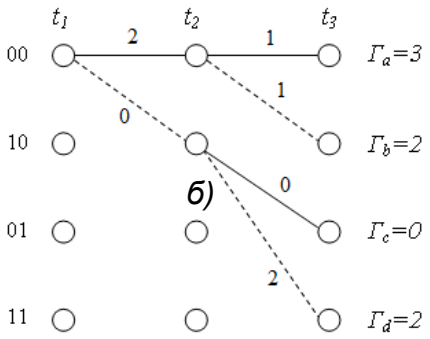
Рисунок 3. Решетчатая диаграмма декодера (7,5)

Теперь рассмотрим детально работу алгоритма декодирования Витерби на примере данной последовательности и кодера. Основной смысл алгоритма Витерби заключается в том, что если два пути в решетке сходятся в одной точке, то один из них при поиске оптимального пути исключается, то есть исключается путь, имеющий большую суммарную метрику пути. В случае совпадения суммарных метрик путей, путь выбирается произвольно.

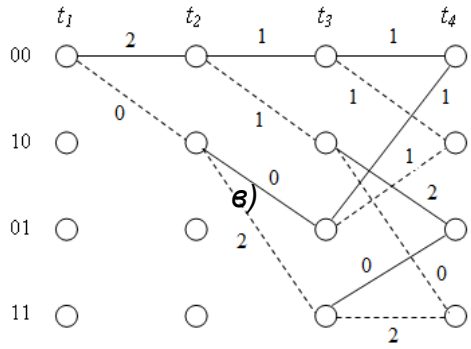
В нашем случае в момент времени t_1 приняты кодовые символы 11. Переходу между состояниями 00→00 соответствует метрика ветви 2, а между состояниями 00→10 метрика ветви 0. (рисунок 4.)



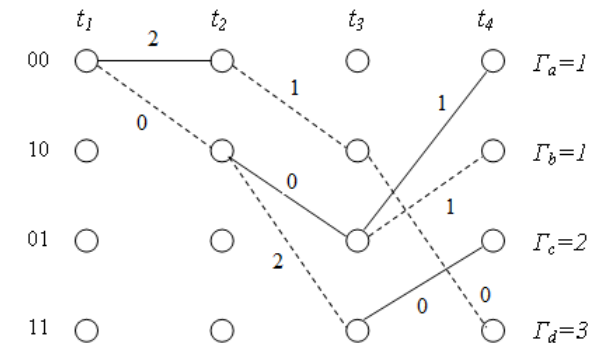
а)



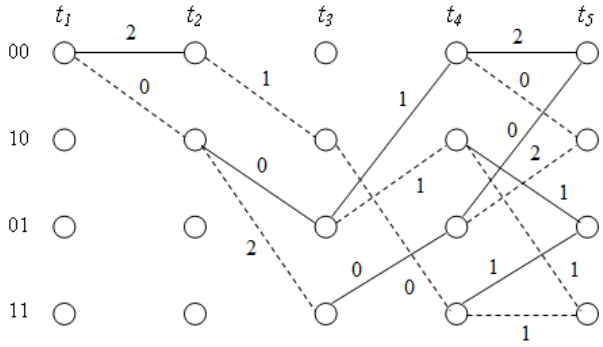
б)



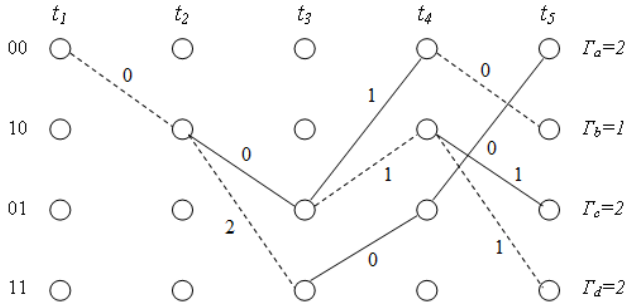
в)



г)



д)



е)

Рисунок 4. Пример работы алгоритма Витерби

В следующий момент времени t_2 из каждого предыдущего состояния выходят еще 2 ветви (рис.4, б). Через Γ_a, Γ_b, Γ_c и Γ_d обозначены суммарные метрики путей. В момент времени t_3 опять происходит разветвление путей (рис 4, в) и в момент времени t_4 имеется по 2 пути, входящие в каждое состояние. Путь, имеющий наибольшую суммарную метрику, может быть исключен. В результате на рисунке 4, г показаны выжившие пути. Та же самая процедура происходит в момент времени t_5 (рисунок 4, д). На рис. 4, е показаны выжившие пути в данный момент времени. Здесь между моментами времени t_1 и t_2

остался только один выживший путь, который называется полной ветвью. При этом декодер, исходя из свойств решетки, решает, что сделан переход 00→10, которому соответствует единичный входной бит. Далее на каждом следующем шаге происходит исключение одного из путей, ведущих в каждое состояние. Таким образом, декодер будет постепенно проходить вглубь решетки, и устранять все пути кроме одного. Следует заметить, что первый бит декодируется только тогда, когда декодер углубится внутрь решетки на некоторое расстояние. Обычно оно равно 4-5 длинам кодового ограничения.

Методические указания по выполнению

Заранее выбрать один из простых сверточных кодов и согласовать его параметры с преподавателем. (Можно использовать дин из блоковых кодов, примененных в предыдущих лабораторных работах.) Выбрать исходную случайную двоичную последовательность символов достаточной длины и произвести ее сверточное кодирование с помощь выбранного кода.

После этого аналитически произвести декодирование и сравнить результат с исходной последовательностью.

Декодирование производить пошаговым образом с подробным построением соответствующих фрагментов кодовой решетки, указанием текущих метрик узлов решетки и переходов. При этом выделять выжившие и отбрасываемые пути по решетке.

Составить программы в среде Matlab для осуществления декодирования и с помощью компьютера осуществить декодирование сверточного кода.

Программа работы

1. Составить и ввести в компьютер программу формирования случайной двоичной последовательности.
2. Закодировать данную случайную последовательность с помощью сверточного кода.
3. Составить и ввести в компьютер программу декодирования кодовой последовательности.
4. Повторить пункты 1-5, введя случайную последовательность, которая использовалась при аналитических расчетах и сравнить полученные результаты.
5. Показать полученные результаты преподавателю.

Содержание отчета

1.Наименование и цель работы.

2.Краткие теоретические сведения по тематике работы.

3.Результаты аналитических расчетов.

4.Результаты исследований на компьютере.

5.Выводы по работе.

Контрольные вопросы

1. Какие алгоритмы могут быть использованы для декодирования сверточных кодов?
2. Что такое решетчатая диаграмма?
3. По каким принципам стоится решетчатая диаграмма?
4. Какой принцип реализует алгоритм Витерби?
5. Что такое метрика переходов и метрика состояний?
6. Как реализуется декодирование сверточных кодов алгоритмом Витерби с использованием решетчатой диаграммы?
7. В чем разница действий при использовании «жесткого» и «мягкого» алгоритмов декодирования?
8. Что дает использование «мягкого» алгоритмов декодирования?

ЛАБОРАТРНАЯ РАБОТА №7

ИЗУЧЕНИЕ МЕТОДОВ ДЕКОДИРОВАНИЯ БЛОКОВЫХ КОДОВ

Цель работы

1. Изучение метода декодирования блоковых кодов.

Краткие теоретические сведения

Рассмотрим декодирование блоковых циклических кодов, как самого распространенного вида используемых блоковых кодов. При «жестком» декодировании на уровне демодулятора все принимаемые символы считаются взаимно независимыми, поэтому каждый из них независимо декодируется по принципу максимального правдоподобия. А уже после этого декодер, использую определенные связи между всей совокупностью символов блока, которая была введена при кодировании за счет использования избыточности, исправляет возможные ошибки в информационной части бока.

В случае «мягкого» декодирования реализация принципа максимального правдоподобия тоже потребует для каждого информационного символа выбора на основе

принимаемой реализации его наиболее вероятного значения. Однако при этом проверочные символы несут дополнительную информацию о значении информационных символов и могут быть использованы для *коррекции* тех исходных вероятностей информационных символов, которые ранее определил демодулятор. В результате эти вероятности изменятся, и наиболее вероятными могут оказаться другие значения символов.

Такими образом, «мягкое» декодирование можно трактовать, как коррекцию исходных величин вероятности принимаемых символов, определенных демодулятором, с последующим выбором их наиболее вероятных значений на основе откорректированных величин вероятностей. Подобный подход позволяет значительно уменьшить требуемый объем вычислений.

Рассмотрим реализацию предлагаемого алгоритма.

Как известно, при «жестком» декодировании для обнаружения и исправления ошибок с использованием циклического кода наиболее общий алгоритм состоит из следующей последовательности операций:

- Принятый блок – кодовая комбинация делится на образующий многочлен **g**(x). Если остаток от деления **R**(x)≠0, то определяется вес остатка w (количество единиц в остатке). Если вес остатка равен или меньше числа исправляемых ошибок t (w≤t), то принятую комбинацию можно сложить по модулю 2 с остатком и получить исправленную комбинацию. Однако можно этого и не делать, т.к. при этом будут откорректированы только проверочные символы, которые после детектирования ценности обычно не представляют. Информационную же часть блока при этом можно считать свободной от ошибок.

- Если w>t , то производится циклический сдвиг на один символ влево. Полученная после такого сдвига комбинация снова делится на образующий многочлен **g**(x). Если при этом вес остатка получился w≤t, то эту циклически сдвинутую комбинацию складывают по mod2 с полученным остатком. Затем после этого сложения его результат циклически сдвигают обратно (вправо) на один символ, т.е. восстанавливают ее исходный вид. При этом также восстанавливается и исправленная информационная часть.

- Если после предыдущего циклического сдвига на один символ после деления на образующий полином **g**(x) вновь сохраняется неравенство w>t, производят следующие посимвольные циклические сдвиги влево. При этом после каждого сдвига осуществляется деление сдвинутой комбинации на **g**(x) и анализируется вес остатка. Когда после какого-либо сдвига начинает выполняться условие w≤t, то сдвинутую

комбинацию складывают с полученным остатком, после чего производят обратных циклических сдвигов вправо столько, сколько их было сделано влево.

Методические указания по выполнению

Заранее выбрать циклический блоковый код и согласовать его параметры с преподавателем. Выбрать исходную случайную двоичную последовательность символов достаточной длины и произвести ее блоковое кодирование с помощь выбранного кода.

После этого аналитически произвести декодирование и сравнить результат с исходной последовательностью. Декодирование производить пошаговым образом.

Составить программы в среде Matlab для осуществления декодирования и с помощью компьютера осуществить декодирование блокового кода.

Программа работы

1. Составить и ввести в компьютер программу формирования случайной двоичной последовательности.
2. Закодировать данную случайную последовательность с помощью блокового кода.
3. Составить и ввести в компьютер программу декодирования кодовой последовательности.
6. Повторить пункты 1-5, введя случайную последовательность, которая использовалась при аналитических расчетах, и сравнить полученные результаты.
7. Показать полученные результаты преподавателю.

Содержание отчета

1.Наименование и цель работы.

2.Краткие теоретические сведения по тематике работы.

3.Результаты аналитических расчетов.

4.Результаты исследований на компьютере.

5.Выводы по работе

Контрольные вопросы

1. Какие методы декодирования блоковых кодов используются и в каких ситуациях?
2. В чем заключается метод декодировании я циклических кодов?
3. Охарактеризовать операции декодирования циклических кодов
4. Прокомментировать возможности осуществления «жесткого» и «мягкого» декодирования блоковых кодов?

5. Что такое «синдром» и как они используются при декодировании блоковых кодов?
6. Что изменяется в операциях декодирования пари переходе к недвоичным блоковым кодам?

Список литературы

1. Цифровые видеоинформационные системы (теория и практика) [Электронный ресурс] / Дворкович В.П., Дворкович А.В. - М. : Техносфера, 2012. - http://www.studentlibrary.ru/book/ISBN9785948363363.html
2. Основы формирования, передачи и приема цифровой информации [Электронный ресурс] : учебное пособие / В.И. Лузин, Н.П. Никитин, В.И. Гадзиковский. - М. : СОЛОН-ПРЕСС, 2014. - http://www.studentlibrary.ru/book/ISBN9785321019610.html
3. Конечные поля в телекоммуникационных приложениях. Теория и применение FEC, CRC, М-последовательностей/Власов Е.Г. - М.: НИЦ ИНФРА-М, 2016. - 280 с.: ISBN 978-5-16-009437-3 http://znanium.com
4. Кодирование и передача речи в цифровых системах подвижной радиосвязи / Рихтер С.Г. - М.:Гор. линия-Телеком, 2009. - 302 с.: ISBN 978-5-9912-0066-0 – http://znanium.com
5. Сидельников В.М. Теория кодирования. – Физматлит, 2008. – 324с.– ISBN 978-5-9221-0943-7 – www.e.lanbook.com
6. Аверченков, В. И. Системы защиты информации в ведущих зарубежных странах [электронный ресурс] : учеб. пособие для вузов / В. И. Аверченков, М. Ю. Рытов, Г. В. Кондрашин, М. В. Рудановский. – 3-е изд., стереотип. – М. : ФЛИНТА, 2011. – 224 с. – ISBN 978-5-9765-1274-0 http://znanium.com/