Санкт-Петербургский политехнический университет Петра Великого
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе

Дисциплина: Телекоммуникационные технологии

Тема: Помехоустойчивое кодирование.

Выполнил студент гр. 33501/3 Федосеенков Н.Ю.
Преподаватель Богач Н.В.

Санкт-Петербург
2018 г.

1 Цель работы

Изучение методов помехоустойчивого кодирования и сравнение их свойств.

2 Постановка задачи

1. Провести кодирование/декодирование сигнала, полученного с помощью функции randerr кодом Хэмминга 2-мя способами: с помощью встроенных функций encode/decode, а также через создание проверочной и генераторной матриц и вычисление синдрома. Оценить корректирующую способность кода.

2. Выполнить кодирование/декодирование циклическим кодом, кодом БЧХ, кодом Рида-Соломона. Оценить корректирующую способность кода.

3 Теоретический раздел

Помехоустойчивое кодирование Существенно сократить избыточность в передаваемых сообщения позволяет помехоустойчивое кодирование. В общем плане помехоустойчивое кодирование можно понимать как такое кодирование сообщений, при котором элементы связаны определенной зависимостью, позволяющей при ее нарушении указать ошибки и восстановить информацию. Помехоустойчивые коды рассчитаны на определенные ошибки. Это значит, что при других ошибках они могут оказаться недостаточно эффективными.

Сообщения по каналу связи передаются в виде кодограммы. Кодограммой, или кодовой комбинацией, называется упорядоченный набор k элементов, каждый из которых может принимать m значений. Множество всех кодовых комбинаций, поставленных в соответствие сообщениям, называется k-разрядным кодом с основанием m. Таким образом, код - множество кодовых комбинаций, а не одна комбинация.

Коды могут быть:

1. Безызбыточными - комбинации ставятся в соответствие каким-то сообщениям. В безызбыточном коде искажение любого элемента приводит к перерождению кодовой комбинации, т.е. изменению смыслового содержания сообщения.

2. Избыточными - использованы не все возможные комбинации. В этих кодах искажение элемента не всегда приводит к искажению сообщения. (Так, при использовании двоичного кода в случае приема комбинации 1100 неизвестно, какая цифра передана, но и ошибки в распознавании не происходит, так как такой комбинации в коде нет. Ошибка в этом

случае обнаруживается. Скорее всего переданы комбинации 1000 или 0100, поскольку они "ближе"к принятой.) Пример избыточного кодирования показан в вышеприведенной таблице.

Идея помехоустойчивого кодирования заключается как раз в таком разнесении кодовых комбинаций, за счет введения избыточности, при котором искажения элементов не приводит к перерождению комбинаций.

Избыточные коды иногда называют корректирующими. Корректирующую способность оценивают минимальным кодовым расстоянием, которое жестко связано с числом исправляемых или обнаруживаемых ошибок. Минимальное кодовое расстояние - это число разрядов, по которым отличаются кодовые комбинации. Обозначается d.

Коды Хэмминга

Код Хэмминга - наиболее известный из первых самоконтролирующихся и самокорректирующихся кодов. Построен применительно к двоичной системе счисления. Позволяет исправлять одиночную ошибку (ошибка в одном бите) и находить двойную. Построение кодов Хэмминга основано на принципе проверки на четность числа единичных символов: к последовательности добавляется такой элемент, чтобы число единичных символов в получившейся последовательности было четным.

Циклические коды

Циклический код - линейный, блочный код, обладающий свойством цикличности, то есть каждая циклическая перестановка кодового слова также является кодовым словом. Используется для преобразования информации для защиты её от ошибок. Циклические коды незаменимы при необходимости передавать информацию в каналах связи, в которых отсутствует возможность повторной передачи данных. Циклические коды применяются при записи и считывании на HDD, CD и DVD, при использовании USB-портов для обмена информацией, при передаче аудио и видео информации. Среди всего многообразия групповых кодов можно выбрать такие, у которых строки связаны условием цикличности, т.е. все строки матрицы могут быть получены циклическим сдвигом одной строки, которая называется образующей или производящей. Сдвиг осуществляется справа налево, а крайний левый символ перемещается в конец строки, т.е. в крайнее правое положение. Коды, у которых строки матрицы удовлетворяют этому условию, называются циклическими.

Коды БЧХ

Коды Боуза — Чоудхури — Хоквингема (БЧХ-коды) — в теории кодирования это широкий класс циклических кодов, применяемых для защиты информации от ошибок. Отличается возможностью построения кода с заранее определёнными корректирующими свойствами, а именно, минимальным кодовым расстоянием. Частным случаем БЧХ-кодов является код Рида — Соломона. БЧХ-код является циклическим кодом, который можно задать

порождающим полиномом. Для его нахождения в случае БЧХ-кода необходимо заранее определить длину кода и требуемое минимальное расстояние.

Коды Рида-Соломона

Коды Рида — Соломона — недвоичные циклические коды, позволяющие исправлять ошибки в блоках данных. Элементами кодового вектора являются не биты, а группы битов (блоки). Очень распространены коды Рида — Соломона, работающие с байтами (октетами). Код Рида — Соломона является частным случаем БЧХ-кода. В настоящее время широко используется в системах восстановления данных с компакт-дисков, при создании архивов с информацией для восстановления в случае повреждений, в помехоустойчивом кодировании.

Свёрточный код

Сверточные коды это коды, исправляющие ошибки, которые используют непрерывную, или последовательную, обработку информации короткими фрагментами (блоками). Сверточный кодер обладает памятью в том смысле, что символы на его выходе зависят не только от (очередного фрагмента) информационных символов на входе, но и предыдущих символов на его входе. Другими словами, кодер представляет собой последовательную машину или автомат с конечным числом состояний. Состояние кодера определяется содержимым его памяти.

4 Ход работы

4.1 Код Хэмминга

С помощью встроенных функций encode/decode произведём кодирование/декодирование сигнала кодом Хэмминга. Код Matlab представлен на рисунке.

```
msg = [ 0 1 1 1]
% Закодированное сообщение
code_msg = encode(msg, 7, 4)
% Допустим ошибку в закодированном сообщении (1 символ)
code_msg(1) = not(code_msg(1))
% Декодируем закодированное сообщение
dec_msg = decode(code_msg, 7, 4)
if dec_msg == msg
    disp('Success')
end
```

Рис. 4.1: Код Matlab

В результате нашего кода заданный сигнал был сначала закодирован, затем в нём была допущена одна ошибка в первом символе, после чего с помощью встроенной функции decode закодированный сигнал с ошибкой был успешно декодирован, в следствие чего мы получили исходное сообщение [0 1 1 1]. Вывод программы представлен на рисунке.

```
msg =

     0     1     1     1

code_msg =

     0     0     0     0     1     1     1

code_msg =

     1     0     0     0     1     1     1

code_msg =

     1     1     0     0     1     1     1

code_msg =

     1     1     1     0     1     1     1

dec_msg =

     0     1     0     1

Error
```

Рис. 4.2: Резлуьтат выполнения программы

Если убрать ошибку в закодированном сигнале, то мы всё равно получим исходное сообщение. Конечный результат выполнения программы не изменится. Если же сделать несколько ошибкой в закодированном сигнале, то код Хэмминга сумеет исправить лишь одну из них. Пример программы с несколькими ошибками приведён на рисунке.

```
msg = [ 0 1 1 1]
% Закодированное сообщение
code_msg = encode(msg, 7, 4)
% Допустим несколько ошибок в закодированном сообщении
code_msg(1) = not(code_msg(1))
code_msg(2) = not(code_msg(2))
code_msg(3) = not(code_msg(3))
% Декодируем закодированное сообщение
dec_msg = decode(code_msg, 7, 4)
if dec_msg == msg
    disp('Success')
else disp('Error')
end
```

Рис. 4.3: Изменённая программа в Matlab

Результат выполнения изменённой программы приведён на рисунке.

```
msg =

     0     1     1     1


code_msg =

     0     0     0     0     1     1     1


code_msg =

     1     0     0     0     1     1     1


code_msg =

     1     1     0     0     1     1     1


code_msg =

     1     1     1     0     1     1     1


dec_msg =

     0     1     0     1


Error
```

Рис. 4.4: Результат изменённой программы

Как видно, в результате был получен ошибочный результат, на выходе не было получено исходное сообщение. Это следствие из-за большого количества ошибок в закодированном сообщении.

Произведём кодирование/декодирование входного сигнала кодом Хэмминга путём создания генераторной и проверочной матриц, а также вычисление синдрома.

Код, реализующий это, представлен на рисунке.

```
msg = [ 0 1 1 1]
% Проверочная и порождающая матрицы для кода Хэмминга
% с длиной слова (n) 7=2^3 - 1
% и длиной блока исх. сообщения (k) 4 = 7 - 3
% выходной параметр g(k строк, n столбцов)
% parmat - проверочная матрица кода, (n-k)строк,n-столбцов
[parmat, g, n, k] = hammgen(3)
% Принятый вектор
rcv = msg*g
rcv = rem(rcv, ones(1,n).*2)
% Ошибка
rcv(2) = not(rcv(2))
sdrm = rcv*parmat'
sdrm = rem(sdrm, ones(1, n-k).*2)
% Таблица декодирования
trt = syndtable(parmat)
% Конверт. sdrm (первый столбец - старший разряд) в неотр. целое число
sdrm_de = bi2de(sdrm, 'left-msb')
% Вектор коррекции
corrvect = trt(sdrm_de + 1,:)
correctedcode = rem(corrvect+rcv,2)
```

Рис. 4.5: Модифицированный код в Matlab

При умножении исходного сообщения на генераторную матрицу в ее конечной части сохраняется исходная посылка, т.к. соответствующий блок генераторной матрицы представляет собой единичную матрицу.Оставшаяся часть формирует контрольные биты,которые составляют дополнение до нуля суммы по модулю два нужных информационных разрядов. Формирование синдрома происходит с помощью домножения на проверочную матрицу parmat'. С помощью матрицы синдрома был выявлен ошибочный бит в посылке, в нашем случае 2. Затем он был исправлен.

Результат выполнения программы представлен на рисунке.

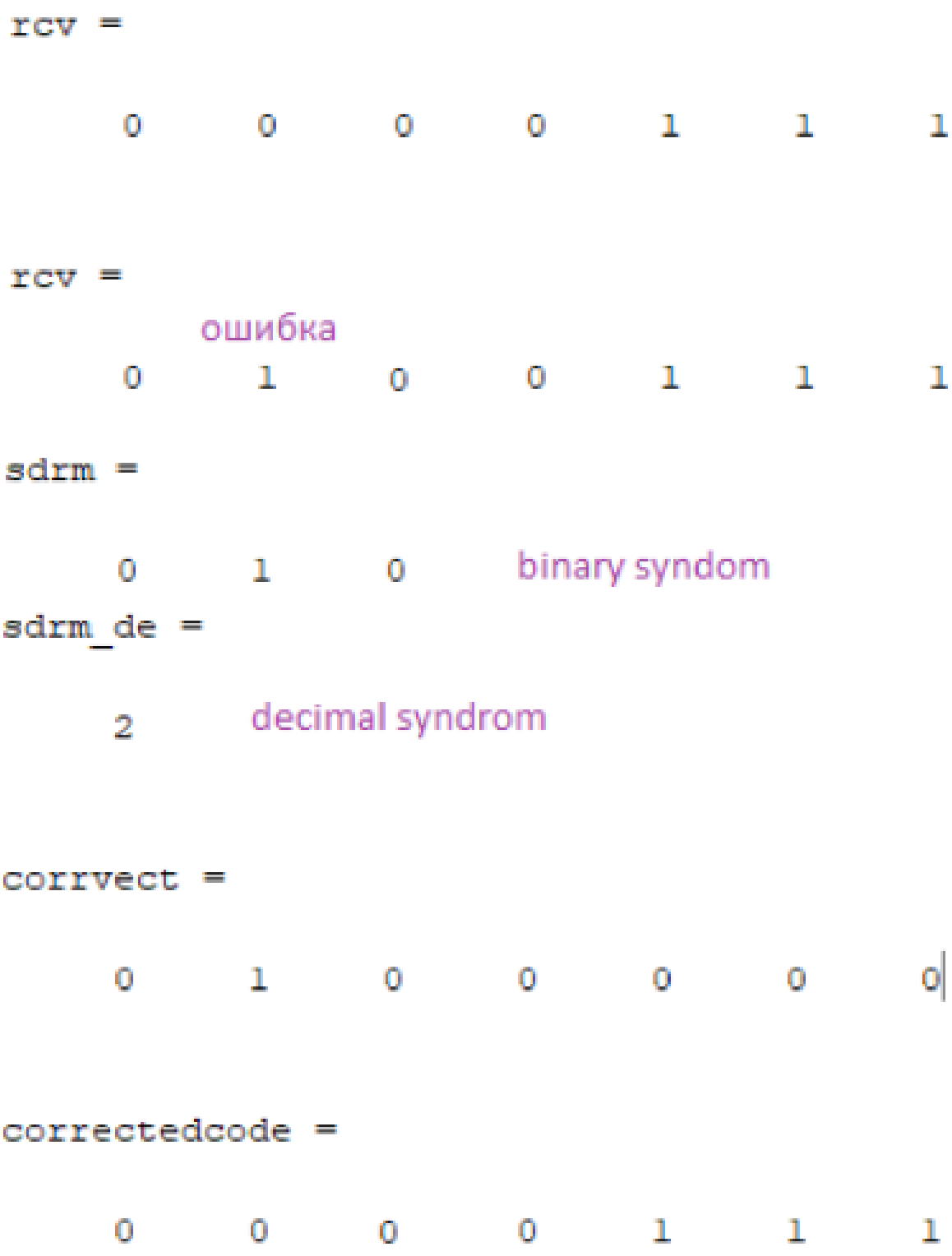

Рис. 4.6: Результат выполнение кода в Matlab

Корректирующая способность кода равна 1

4.2 Циклический код

Было оставлено всё тоже сообщение [0 1 1 1]. В нём произведено кодирование/декодирование:

```
msg = [ 0 1 1 1]
n = 7;
k = 4;
genpoly = cyclpoly(n,k, 'max');
[h,g] = cyclgen(7, genpoly);
code = msg*g;
code = rem(code, ones(1,n).*2)
% Ошибка во 2 символе
code(2)=not(code(2))
sdrm=code*h'
sdrm=rem(sdrm,ones(1,n-k).*2)
% Таблица декодирования
trt = syndtable(h)
sdrm_de = bi2de(sdrm, 'left-msb')
corrvect = trt(sdrm_de + 1,:)
correctedcode = rem(corrvect+code,2)
```

Рис. 4.7: Код в Matlab

Был построен полином циклического кода

$$x^3 + x + 1 \tag{4.1}$$

, которым использовался в качестве параметра функции cyclgen. Были получены порождающая и проверочная матрицы. Ошибка была произведена во 2 разряде. Результат выполнения программы представлен на рисунке.

```
msg =

     0     1     1     1

code =

     0     1     0     0     1     1     1

sdrm =

     0     1     0

sdrm_de =

     2


corrvect =

     0     1     0     0     0     0     0


correctedcode =

     0     0     0     0     1     1     1
```

Рис. 4.8: Результат выполнение кода в Matlab

Корректирующая способность кода равна 1

4.3 Коды БЧХ

Произведено кодирование/декодирование сообщения [0 1 1 1] при помощи кодов БЧХ Данный код представлен на рисунке.

```
msg = [0 1 1 1]
code_msg = comm.BCHEncoder(7,4)
dec_msg = comm.BCHDecoder(7,4)
tmp = msg';
code = step(code_msg, tmp(:))'
% Ошибка
code(2) = not(code(2))
decode = step(dec_msg, code')'
```

Рис. 4.9: Код в Matlab

Результат выполнения программы показан на рисунке.

```
msg =

     0     1     1     1


code =

     0     1     1     1     0     1     0


code =

     0     0     1     1     0     1     0


decode =

     0     1     1     1
```

Рис. 4.10: Результат выполнение кода в Matlab

Допущенная во 2 разряде ошибка была успешно обнаружена и исправлена, что свидетельствует о правильной работе программы.

Корректирующая способность кода равна 1. Существует возможность увеличить её до 2.

4.4 Коды Рида-Соломона

Было сгенерировано три слова по три символа. Также допущены одна ошибка в первом слове, две во втором и три в третьем. Код Matlab представлен на рисунке.

```
m = 3; % Число бит на символ
n = 2^m - 1; % Общее число бит (7)
k = 3; % Длина сообщения

msg = gf([0 1 2; 3 4 5; 6 7 7], m);
code = rsenc(msg, n, k);
err = gf([2 0 0 0 0 0 0; 3 4 0 0 0 0 0; 5 6 7 0 0 0 0], m);
err_code = code + err;

[dec_code, cnum] = rsdec(err_code, n, k);
cnum
```

Рис. 4.11: Код в Matlab

Результат выполнения программы представлен на рисунке.

```
cnum =

     1
     2
    -1
```

Рис. 4.12: Результат выполнение кода в Matlab

Видно, что количество исправленных ошибок в первой и второй строке равно количеству сделанных. В третьем сообщении количество = -1, так как (7,3) RS код не может исправить более 2-х ошибок.

Из всего вышесказанного следует, что корректирующая способность кода равна 2.

5 Выводы

Были рассмотрены такие методы кодирования, как: коды Хэмминга, циклические коды, коды Боуза-Чоудхури-Хоквингема и коды Рида-Соломона. Они являются самокорректирующимися. Выбор кодирования осуществляется в зависимости от поставленной задачи. По корректирующей способности коды Хэмминга уступают более сложным кодам, таким как коды БЧХ, но имеют более прозрачную реализацию. Код Рида-Соломона, в свою очередь, имеет более хороший показатель по корректирующей показатель, а так же позволяет оперировать с десятичными числами и параллельно обрабатывать несколько потоков данных. Корректирующая способность циклического кода определяется видом образующего многочлена.