

Санкт-Петербургский Государственный Политехнический Университет
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

ОТЧЕТ

По лабораторной работе №7

Дисциплина: Телекоммуникационные технологии

Тема: Помехоустойчивое кодирование

Выполнила студентка гр. 33501/2 _____ Белобородова В. Г.

Преподаватель _____ Богач Н.В.

< ____ > _____ 2018 г.

Санкт-Петербург
2018

Оглавление

1. Цель работы.....	3
2. Постановка задачи	3
3. Теоретические сведения.....	3
4. Ход работы	4
Хэмминг – встроенные функции	4
Хэмминг – матрицы	5
Циклический код.....	6
Код БЧХ	7
Код Рида-Соломона.....	8
5. Вывод	9

1. Цель работы

Изучение методов помехоустойчивого кодирования и сравнение их свойств.

2. Постановка задачи

1. Провести кодирование/декодирование сигнала кодом Хэмминга двумя способами с помощью встроенных функций encode/decode, а также через создание проверочной и генераторной матриц и вычисление синдрома. Провести кодирование/декодирование с помощью циклических кодов

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

3. Теоретические сведения

Кодирование – представления дискретных сигналов, передаваемых по цифровому каналу связи, с целью передачи данных, представленных в цифровом виде, на расстояние по физическому каналу связи (такому как оптическое волокно, витая пара, коаксиальный кабель, инфракрасному излучению). Физическое кодирование также применяется для записи данных на цифровой носитель. При физическом кодировании уделяют внимание характеристикам формируемого сигнала: ширину полосы частот, гармонический состав сигнала, способность к синхронизации приёмника с передатчиком. При физическом кодировании решаются вопросы синхронизации, управления полосой пропускания сигнала, скорость передачи данных и расстояние на которое необходимо передать данные.

Кодирование передаваемого сообщения позволяет осуществлять его проверку на наличие ошибок при получении, а в некоторых случаях и исправлять их. Данная возможность достигается за счет введения информационной избыточности, что уменьшает удельное количество полезной информации в сообщении.

Основными задачами кодирования являются повышение помехоустойчивости передаваемых сообщений, удаление избыточности из закодированных сообщений и защита информации от несанкционированного доступа (постороннего прослушивания).

Автоматическое кодирование осуществляется в устройстве называемом *кодером*, а обратный процесс декодирование происходит в *декодере*.

Код Хэмминга является циклическим самокорректирующимся кодом. Помимо информационных бит в сообщении передается набор контрольных бит, которые вычисляются как сумма по модулю 2 некоторых информационных бит (зависит от номера контрольного бита). Для m контрольных бит максимальное число информационных бит составляет $2^m - n - 1$. Код Хэмминга позволяет обнаружить до двух ошибок при передаче и исправить инверсную передачу одного двоичного разряда.

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 (Боуза – Чоудхури – Хоквингема) позволяют при необходимости исправлять большее число ошибок в разрядах за счет внесения дополнительной избыточности. Они принадлежат к категории блочных кодов. Частным случаем БЧХ кодов являются коды **Рида-Соломона**, которые работают с недвоичными данными или группами двоичных кодов. Их корректирующая способность, соответственно, не ниже, чем у кодов Хэмминга.

Свёрточные коды это коды, исправляющие ошибки, которые используют непрерывную, или последовательную, обработку информации короткими фрагментами (блоками). Свёрточный 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

Хэмминг – встроенные функции

С помощью встроенных функций исследуем передачу сообщения из 4 бит:

```
msg = [1 0 0 1]
code = encode(msg, 7, 4)
code(1) = not(code(1));
[dec, err] = decode(code, 7, 4)
code(6) = not(code(6));
[dec, err] = decode(code, 7, 4)
```

Результат выполнения:

```
msg =
    1     0     0     1

code =
    0     1     1     1     0     0     1

dec =
    1     0     0     1

err =
    1

dec =
    1     1     1     1

err =
    1
```

Т.е. как и ожидалось, 1 ошибку алгоритм вычислил и исправил, а 2 ошибки оказалось слишком много для верного восстановления сообщения.

Хэмминг – матрицы

```
msg = [1 0 1 0]
[h,g,n,k] = hamngen(3) % Получение матриц

m = msg*g % Досложить на генераторную м.
m = rem(m,ones(1,n).*2) % Нужно по модулю 2

m(4) = not(m(4)) % Внесение ошибки
synd = m*h' % Получение синдрома
synd = rem(synd,ones(1,n-k).*2) % Нужно по модулю 2

stbl = syndtable(h) % Produce syndrome decoding table
tmp = bi2de(synd,'left-msb')
z = stbl(tmp+1,:)
rez = xor(m,z)
```

Результат выполнения:

```
msg =
    1     0     1     0

h =
    1     0     0     1     0     1     1
    0     1     0     1     1     1     0
    0     0     1     0     1     1     1

g =
    1     1     0     1     0     0     0
    0     1     1     0     1     0     0
    1     1     1     0     0     1     0
    1     0     1     0     0     0     1

n =
    7

k =
    4

m =
    2     2     1     1     0     1     0

m =
    0     0     1     1     0     1     0

m =
    0     0     1     0     0     1     0

synd =
    1     1     2

synd =
```

	1	1	0				
stbl =							
	0	0	0	0	0	0	0
	0	0	1	0	0	0	0
	0	1	0	0	0	0	0
	0	0	0	0	1	0	0
	1	0	0	0	0	0	0
	0	0	0	0	0	0	1
	0	0	0	1	0	0	0
	0	0	0	0	0	1	0
tmp =							
	6						
z =							
	0	0	0	1	0	0	0
rez =							
	0	0	1	1	0	1	0

В результате удалось восстановить верное сообщение после 1 ошибки. В чём бы Матлаб не измерял синдром, главное, он его исправляет.

Как было выяснено в предыдущем пункте, более 1 ошибки исправить не удастся.

Циклический код

```

msg = [1 0 0 1]
n = 7;
k = 4;
pol = cyclpoly(7,4)
[h,g] = cyclgen(7,pol)

code = msg*g
code = rem(code,ones(1,n).*2)

code(3) = not(code(3))
synd = code*h'
synd = rem(synd,ones(1,n-k).*2)

stbl = syndtable(h)
tmp = bi2de(synd,'left-msb')
z = stbl(tmp+1,:)
rez = xor(code,z)

```

Результат выполнения:

msg =	1	0	0	1			
pol =	1	0	1	1	- полином $x^3 + x^1 + 1$		
h =	1	0	0	1	1	1	0

	0	1	0	0	1	1	1
	0	0	1	1	1	0	1
g =							
	1	0	1	1	0	0	0
	1	1	1	0	1	0	0
	1	1	0	0	0	1	0
	0	1	1	0	0	0	1
code =							
	1	1	2	1	0	0	1
code =							
	1	1	0	1	0	0	1
code =							
	1	1	1	1	0	0	1
synd =							
	2	2	3				
synd =							
	0	0	1				
stbl =							
	0	0	0	0	0	0	0
	0	0	1	0	0	0	0
	0	1	0	0	0	0	0
	0	0	0	0	0	0	1
	1	0	0	0	0	0	0
	0	0	0	1	0	0	0
	0	0	0	0	0	1	0
	0	0	0	0	1	0	0
tmp =							
	1						
z =							
	0	0	1	0	0	0	0
rez =							
	1	1	0	1	0	0	1

Если допустить ещё одну ошибку, сообщение будет уже не восстановить. Поэтому корректирующая способность кода = 1.

Код БЧХ

```
msg = [1 0 0 1]

codebch = comm.BCHEncoder(7,4);
decbch = comm.BCHDecoder(7,4);
temp = msg';
code = step (codebch , temp(:))'
```

```
code(3) = not(code(3))
decode = step (decbch, code')'
```

Результат выполнения:

```
msg =
    1     0     0     1

code =
    1     0     0     1     1     1     0

code =
    1     0     1     1     1     1     0

decode =
    1     0     0     1
```

Если допустить ещё одну ошибку, сообщение будет уже не восстановить. Поэтому корректирующая способность кода = 1.

Код Рида-Соломона

Чтобы узнать корректирующую способность кода, будем постепенно вносить ошибки (строка в коде помечена):

```
m = 3;
n = 2^m - 1;
k = 3;

msg = gf([0 1 2; 3 4 5; 6 7 6], m)
code = rsenc(msg,n,k)
errs = gf([0 0 0 0 0 0 0; 2 0 0 0 0 0 0; 0 0 0 0 0 0 0], m); % Сюда!
code = code + errs

[dec,errnum] = rsdec(code,n,k)
```

Результат выполнения с 1 ошибкой:

```
msg = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)
Array elements =
    0     1     2
    3     4     5
    6     7     6

code = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)
Array elements =
    0     1     2     2     3     1     3
    3     4     5     3     2     2     4
    6     7     6     2     7     3     3

code = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)
Array elements =
```


0	1	2	2	3	1	3
1	4	5	3	2	2	4
6	7	6	2	7	3	3

```
dec = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)
```

```
Array elements =
```

0	1	2
3	4	5
6	7	6

```
errnum =
```

```
0
```

```
1
```

```
0
```

Сообщение с одной ошибкой восстановлено верно.

Программа с двумя ошибками отличается только матрицей «% Сюда». С двумя ошибками сообщение восстановлено также верно.

Если в одной строке допустить более 2 ошибок, то сообщение декодируется неверно.

Корректирующая способность кода = 2.

5. Вывод

В данной работе проведены кодирования и декодирования посылок кодами Хэмминга, циклическим, БЧХ, Рида-Соломона. Выбирать метод кодирования стоит в зависимости от типа посылки и зашумлённости канала. Коды Рида-Соломона позволяют работать с произвольной системой счисления, кроме того избыточность позволяет восстановить до 2 ошибок. Остальные исследованные коды позволяют исправить только 1 ошибку в бинарном коде.