Санкт-Петербургский государственный политехнический университет

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе № 7

на тему

Помехоустойчивые коды

Работу выполнила:
студентка гр. 33501/2 Акимова М.А.

Преподаватель:
Богач Н.В.

2018

1. Цель работы

Изучение методов помехоустойчивого кодирования и сравнение их свойств.

2. Постановка задачи

Провести кодирование/декодирование сигнала кодом Хэмминга 2мя 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

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

3. Теоретическая часть

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 на характеристики 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 всех информационных бит, кроме одного. 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.

Коды БЧХ 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. Их корректирующая способность, соответственно, не ниже, чем у кодов Хэмминга.

4. Ход работы

4.1 Код Хэмминга

С помощью встроенных функций encode/decode произведём кодирование/декодирование сигнала кодом Хэмминга.

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

При передаче сообщения без ошибки количество ошибок равно нулю.

Допустим ошибку в 1 бите обнаруживается одна ошибка err = 1, которая исправляется и сообщение декодируется верно.

```
message =

     0     0     0     1


code =

     1     0     1     0     0     0     1


code =

     0     0     1     0     0     0     1


dec =

     0     0     0     1


err =

     1
```

Произведем кодирование/декодирование сигнала кодом Хэмминга через создание проверочной и генераторной матриц и вычисление синдрома.

```
msg = [0 0 0 1]
[h,g,n,k] = hammgen(3)


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


m(4) = not(m(4))
synd = m*h'
synd = rem(synd,ones(1,n-k).*2)


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

Синдром был вычислен домножением на матрицу h', после чего с помощью матрицы синдрома выявляем ошибочный бит в посылке и исправляем его:

```
msg =

     0     0     0     1

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

m =

     1     0     1     0     0     0     1

m =

     1     1     1     0     0     0     1

synd =

     2     1     2

synd =

     0     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 =

     2

s =

     0     1     0     0     0     0     0

res =

     1     0     1     0     0     0     1
```

4.2 Циклический код

Выполним кодирование и декодирование сообщения [0 0 0 1]:

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


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


code(2) = not(code(2))
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)
```

msg = 0 0 0 1

pol = 1 0 1 1

tmp = 2

z = 0 1 0 0 0 0 0

rez = 0 1 1 0 0 0 1

Сначала строится порождающий полином циклического кода: x^3+x+1. Далее, использовав этот полином в качестве одного из параметров функции cyclgen, получили порождающую и проверочную матрицы для данного кода. Допущенная ошибка вj 2 разряде была успешно обнаружена и исправлена синдромом. Исправляющая способность кода равна 1

4.3 Код БХЧ

Произведем кодирование и декодирование сообщения [0 0 0 1] при помощи кодов БЧХ:

```
msg = [0 0 0 1]


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


code(2) = not(code(2))
decode = step (decbch , code')'
```

```
msg =

     0     0     0     1


codebch =

  System: comm.BCHEncoder

  Properties:
              CodewordLength: 7
               MessageLength: 4
    PrimitivePolynomialSource: 'Auto'
    GeneratorPolynomialSource: 'Auto'
        PuncturePatternSource: 'None'


decbch =

  System: comm.BCHDecoder

  Properties:
                CodewordLength: 7
                 MessageLength: 4
      PrimitivePolynomialSource: 'Auto'
      GeneratorPolynomialSource: 'Auto'
          PuncturePatternSource: 'None'
             ErasuresInputPort: false
    NumCorrectedErrorsOutputPort: true


code =

     0     0     0     1     0     1     1


code =

     0     1     0     1     0     1     1


decode =

     0     0     0     1
```

Допущенная ошибка во 2 разряде была успешно успешно обнаружена и исправлена. Исправляющая способность кода равна 1.

4.4 Код Рида-Соломона

Произведем кодирование и декодирование посылки при помощи кодов Рида-Соломона. Количество информационных бит равно 3, количество бит на символ 3, общее число бит таким образом будет равно 7.

```
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 4 0 0 0; 2 0 0 0 2 0 0; 3 4 5 0 0 0 0 ],m);
code = code + errs


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

```
errnum =

     1
     2
    -1
```

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.

Корректирующая способность кода равна 2.

5. Выводы:

В ходе лабораторной работы были рассмотрены методы кодирования такие, как код Хэмминга, циклический код, код БХЧ, код Рида-Соломона. Они являются самокорректирующимися. Выбор кодирования осуществляется в зависимости от поставленной задачи. По корректирующей способности коды Хэмминга уступают более сложным кодам, например, БХЧ, но имеют более прозрачную реализацию. Код Рида-Соломона имеет лучшую корректирующую способность, позволяет оперировать с десятичными числами и параллельно обрабатывать несколько потоков данных. Каждый из исследованных кодов имеет преимущества и недостатки, поэтому выбор используемого кода зависит от поставленной задачи.