

Санкт-Петербургский государственный политехнический университет
Кафедра компьютерных систем и программных технологий

Телекоммуникационные технологии
Лабораторная работа №7
Помехоустойчивое кодирование

Выполнил:
студент гр.33501/4
Корсков Алексей
Проверила:
Богач Н.В.

Санкт-Петербург
2018

Цель

Изучение методов помехоустойчивого кодирования и сравнение их свойств

Постановка задачи

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

Теоретическое обоснование

Кодированием и декодированием (в широком смысле) называют любое преобразование сообщения в сигнал и обратно, сигнала в сообщение, путем установления взаимного соответствия. Преобразование следует считать оптимальным, если в конечном итоге производительность источника и пропускная способность канала окажутся равными, т.е. возможности канала будут полностью использованы. Данное преобразование разбивается на два этапа:

- модуляция-демодуляция, позволяющая осуществить переход от непрерывного сигнала радиоканала к дискретному;
- кодирование-декодирование (в узком смысле), во время которого все операции выполняются над последовательностью символов.

В свою очередь, кодирование-декодирование делится на два противоположных по своим действиям действиям этапа:

- устранение избыточности в принимаемом от источника сигнале (экономное кодирование);
- внесение избыточности в передаваемый по каналу цифровой сигнал (помехоустойчивое или избыточное кодирование) для повышения достоверности передаваемой информации.

Циклический код — подкласс линейных кодов, обладающие следующим свойством: циклическая подстановка символов в кодированном блоке дает другое возможное кодовое слово того же кода.

К циклическим кодам относятся коды Хэмминга, которые являются одним из немногочисленных примеров совершенных кодов. Длина кодированного блока равна $2m-1$. Порождающая и проверочная матрицы для кодов Хэмминга генерируются функцией `hammgen`.

Среди циклических кодов широкое применение нашли коды Боуза-Чоудхури-Хоквингема (БЧХ). Для работы с ними есть функции `bchenco` (кодирование) и `bcddeco` (декодирование). Функция `bchpoly` позволяет рассчитывать и считывать параметры или порождающий полином для двоичных кодов БЧХ.

Частным случаем БЧХ кодов являются коды Рида-Соломона - подкласс циклических блочных кодов. Это единственные поддерживаемые пакетом Communications недвоичные коды. Для работы с этим кодом есть функции `rsenco` (кодирование) и `rsdeco` (декодирование). Функции `rsencof` и `rsdecocf` осуществляют кодирование и декодирование текстового файла. Функция `rspoly` генерирует порождающие полиномы для кодов Рида-Соломона.

Ход работы

1. Результат кодирования/декодирования кодом Хэмминга с использованием стандартных функций:

```
msg =  
    0    1    0    1  
  
code =  
    1    1    0    0    1    0    1  
  
code =  
    0    1    0    0    1    0    1  
  
dec =  
    0    1    0    1  
  
err =  
    1
```

Рис.1 Результат работы программы

2. Результат кодирования/декодирования кодом Хэмминга с использованием проверочной и генераторной матриц и вычисление синдрома:

msg =

0 1 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

n =

7

k =

4

synd =

1 1 0

tmp =

6

z =

0 0 0 1 0 0 0

rez =

5

1 1 0 0 1 0 1

Рис.2 Результат работы программы

3. Результат кодирования/декодирования циклическим кодом:

```
msg =  
    0    1    0    1  
  
pol =  
    1    0    1    1  
  
tmp =  
    2  
  
z =  
    0    1    0    0    0    0    0  
  
rez =  
    1    0    0    0    1    0    1
```

Рис.3 Результат работы программы

4. Результат кодирования/декодирования кодом БЧХ:

```

msg =

    0    1    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    1    0    1    1    0    0

code =

    0    0    0    1    1    0    0

decode =

    0    1    0    1

```

Рис.4 Результат работы программы

5. Результат кодирования/декодирования кодом Рида-Соломона:


```
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	4	5	3	2	2
6	7	6	2	7	3

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

```
Array elements =
```

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

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

```
Array elements =
```

0	1	2
3	4	5
5	3	3

```
errnum =
```

```
1
2
-1
```

Рис.5 Результат работы программы

Вывод

В ходе данной работы были получены навыки кодирования цифровых сигналов. Кодирование таких сигналов происходит по принципу избыточности. Каждый из исследованных кодов имеет свои преимущества и недостатки, поэтому использование конкретного из них должно быть обусловлено постановкой определенной задачи. Код Хэмминга достаточно простой в использовании, не требует больших мощностей. Однако он может исправить только одну допущенную ошибку в переданном сообщении. Код Рида-Соломона способен исправлять несколько ошибок, так же он может оперировать десятичными числами, а не только двоичными.