

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

Телекоммуникационные технологии
Лабораторная работа №7
Помехоустойчивое кодирование

Выполнил:
студент гр.33501/4
Курякин Д. А.
Проверила:
Богач Н.В.

Санкт-Петербург
2018

1 Цель

Изучение методов помехоустойчивого кодирования и сравнение их свойств

2 Постановка задачи

- Провести кодирование/декодирование сигнала, полученного с помощью функции `randert` 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.

3 Теоретическое обоснование

Обычно в процессе кодирования информация преобразуется из формы, удобной для непосредственного использования, в форму, удобную для передачи, хранения или автоматической обработки. В более узком смысле кодированием информации называют представление информации в виде кода. Средством кодирования служит таблица соответствия знаковых систем, которая устанавливает взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем. Декодирование — это процесс, обратный процессу кодирования, процесс выявления первоначального смысла, исходной идеи отправителя, понимания смысла его сообщения.

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 $2^m - 1$. 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 `rsdecnof` осуществляют кодирование и декодирование текстового файла. Функция `rspoly` генерирует порождающие полиномы для кодов Рида-Соломона.

4 Ход работы

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

```
msg =  
    0    1    1    0  
code =  
    1    0    0    0    1    1    0  
code =  
    0    0    0    0    1    1    0  
dec =  
    0    1    1    0  
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 =
    1x7 logical array
    1    1    0    0    1    0    1

```

Рис.2 Результат работы программы

3. Результат кодирования/декодирования циклическим кодом:

```

msg =
    0    1    1    0
pol =
    1    0    1    1
tmp =
    2
z =
    0    1    0    0    0    0    0
rez =
    1x7 logical array
    0    0    1    0    1    1    0

```

Рис.3 Результат работы программы

4. Результат кодирования/декодирования кодом БЧХ:

```

msg =
    0    1    1    0
codebch =
    comm.BCHEncoder with properties:
        CodewordLength: 7
        MessageLength: 4
        ShortMessageLengthSource: 'Auto'
        ShortMessageLength: 4
        GeneratorPolynomialSource: 'Auto'
        PrimitivePolynomialSource: 'Auto'
        PuncturePatternSource: 'None'
decbch =
    comm.BCHDecoder with properties:
        CodewordLength: 7
        MessageLength: 4
        ShortMessageLengthSource: 'Auto'
        ShortMessageLength: 4
        GeneratorPolynomialSource: 'Auto'
        PrimitivePolynomialSource: 'Auto'
        PuncturePatternSource: 'None'
        ErasuresInputPort: false
        NumCorrectedErrorsOutputPort: true
code =
    0    1    1    0    0    0    1
code =
    0    0    1    0    0    0    1
decode =
    0    1    1    0

```

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

errnum =
    1
    2
    -1

```

Рис.5 Результат работы программы

5 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.