САНКТ-ПЕТЕРБУРГСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет
по лабораторной работе №7
на тему: "Помехоустойчивое кодирование"

выполнила:
Шевченко А.С.
группа: 33501/1
преподаватель:
Богач Н.В.

Санкт-Петербург
2018

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;
3. Выполнить кодирование/декодирование кодом БЧХ;
4. Выполнить кодирование/декодирование кодом Рида-Соломона;
5. Оценить корректирующую способность кода.

3. Теоретическая часть: Помехоустойчивое кодирование

1. Код Хэмминга

Коды Хэмминга являются одним из подклассов циклических блочных кодов. Порождающий полном для кодов Хэмминга неприводим и примитивен, а длина кодированного блока равна 2m – 1. Порождающая и проверочная матрицы для кодов Хэмминга генерируются функцией hammgen.

2. Циклические коды

Циклические коды — это подкласс линейных кодов, обладающие тем свойством, что циклическая перестановка символов в кодированном блоке дает другое возможное кодовое слово того же кода. Для работы с циклическими кодами в пакете Communications есть две функции. Задав число символов в кодируемом и закодированном блоках, с помощью функции cyclpoly можно получить порождающий полином циклического кода. Далее, использовав этот полином в качестве одного из параметров функции cyclgen, можно получить порождающую и проверочную матрицы для данного кода.

3. Коды БЧХ

Коды БЧХ являются одним из подклассов циклических блочных кодов. Для работы с ними функции высокого уровня вызывают специализированные функции bchenco (кодирование) и bchdeco (декодирование). Кроме того, функция bchpoly позволяет рассчитывать параметры или порождающий полином для двоичных кодов БЧХ.

4. Коды Рида-Соломона

Коды Рида—Соломона являются одним из подклассов циклических блочных кодов. Это единственные поддерживаемые пакетом Communications недвоичные коды. Для работы с кодами Рида—Соломона функции высокого уровня вызывают специализированные функции rsenco (кодирование) и rsdeco (декодирование). Кроме того, функции rsencode и rsdecode позволяют использовать при кодировании и декодировании экспоненциальный формат данных, а функции rsencof и rsdecof осуществляют кодирование и декодирование текстового файла. Функция rspoly генерирует порождающие полиномы для кодов Рида—Соломона.

4. Ход работы

Продемонстрируем работу кода Хэмминга. Длина посылки будет равна 4, следовательно потребуется 3 контрольных бита. Итого закодированное сообщение будет длиной 7 бит.**Код Хэмминга**

```
%% 1
x = 4;
y = 7;

message = [1 0 1 0];
code = encode(message, y, x, 'hamming/binary');
result = decode(code, y, x, 'hamming/binary');

code(3) = xor(code(3), 1);
result_err = decode(code, y, x, 'hamming/binary');
```

Рис. 1: Скрипт для кодирования кодом Хэмминга

После кодирования и раскодирования посылки 1010, мы получим один и тот же результат. Попробуем инвертировать третий бит, чтобы смоделировать поведение кода при ошибке. После запуска скрипта, как и ожидалось, результат верный - 1010, ошибка исправлена.

```
x = 4;
y = 7;

message = [1 0 1 0];
[h, g] = hammgen(3, [1 0 1 1]);
code = message * g;

for i = 1:length(code)
    code(i) = mod(code(i), 2);
end

code(3) = xor(code(3), 1);
syndrom = code * h';

for i = 1:length(syndrom)
    syndrom(i) = mod(syndrom(i), 2);
end
```

Рис. 2: Скрипт для кодирования кодом Хэмминга с использованием генераторной матрицы

Данный код демонстрирует кодирование Хэмминга с ипользованием генераторной и проверочной матрицы. Домножив посылку на проверочную матрицу, мы получим синдром, который поможит идентифицировать бит с ошибкой.

2. **Циклический код**

```
x = 4;
y = 7;

gen_poly = cyclpoly (y, x, 'max');
[h, g] = cyclgen(y, gen_poly);

message = [1 0 1 0];
code = message * g;

for i = 1:length(code)
    code(i) = mod(code(i), 2);
end

%code(3) = xor(code(3), 1);
syndrom = code * h';

for i = 1:length(syndrom)
    syndrom(i) = mod(syndrom(i), 2);
end
```

Рис. 3: Скрипт для кодирования циклическим кодом

Данный иклический код может обнаружить и исправить одну ошибку. Кодирование осуществляется с помощью генераторной матрицы, синдром также можно получить, умножив закодированное сообщение на проверочную матрицу.

3. **Код БЧХ**

```
k = 4;
n = 7;%2^m-1

err_num = bchnumerr(n, k);

message = gf(randi([0 1], 1, k))
code = bchenc(message, n, k);

noise_code = code + randerr(1, n, 1:err_num);
result = bchdec(noise_code, n, k)
```

Рис. 4: Скрипт для кодирования кодом БЧХ

Коды БЧХ могут исправлять более одной ошибки. Мы передаем сообщение длительностью 4 бит и делаем ошибку в одном бите. В результате мы получили исправленную посылку.

4. **Код Рида-Соломона**

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

message = gf(randi([0 n], 1, k), m);
code = rsenc(message, n, k);

error = gf([2 0 0 0 0 0 0], m)
noise_code = code + error;

[result, enumerr] = rsdec(noise_code, n, k);

```

Рис. 5: Скрипт для кодирования кодом Рида-Соломона

Коды Рида-Соломона могут работать с недвоичными данными. Здесь мы генерируем случайное сообщение и также добавляем ошибку в старшем бите. Массив enumerr уведомит нас о том, что в посылке есть ошибка.

5. Выводы

Таким образом, мы продемонстрировали разлицные виды помехоустойчивого кодирования.

Код Хэмминга прост в применении, однако недостаток в том, что мы можем исправить только одну ошибку. Коды БЧХ исправляют более одной ошибки, в зависимости от длины сообщения. Если количество ошибок будет превышать норму для заданного сообщения, то декодер начнет ошибаться.Коды Рида-Соломона позволяют работать с небинарными данными и исправлять более одной ошибки.Циклический код обнаруживает все одиночные ошибки, если образующий полином содержит более одного члена.