

Санкт-Петербургский государственный технический
университет
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе №7
Помехоустойчивое кодирование
Телекоммуникационные технологии

Кузьмин Н.А. 33501/1
Богач Н.В.

Санкт-Петербург 2018

Содержание

1	Цель	3
2	Постановка задачи	3
3	Ход работы	3
3.1	Код Хэмминга	3
3.2	Циклический код	5
3.3	Коды БЧХ	6
3.4	Коды Рида-Соломона	7
4	Вывод	8

1 Цель

Изучение методов помехоустойчивого кодирования и сравнение их свойств.

2 Постановка задачи

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

3 Ход работы

3.1 Код Хэмминга

Код Хэмминга является самокорректирующимся кодом. Помимо информационных бит, в коде содержатся также контрольные биты, вычисляющиеся как сумма информационных бит по модулю два (всех, кроме одного бита). Позволяет исправить одиночную ошибку и находить двойную. Сначала выполним кодирование/декодирование сигнала кодом Хэмминга с помощью встроенных функций `encode/decode`.

```
msg = [1 1 0 0]
code = encode(msg, 7, 4);
code(3) = not(code(3));
res = decode(code, 7, 4)
```

Как мы видим, при совершенной ошибке в третьем бите сообщение декодировалось верно:

```
msg = [1 1 0 0]
code = encode(msg, 7, 4);
code(3) = not(code(3));
res = decode(code, 7, 4)
```

Теперь выполним кодирование сигнала кодом Хэмминга вручную.

```
msg = [1 1 0 0]
[h,g,n,k] = hamngen(3);
m = msg*g
m = rem(m, ones(1,n).*2)

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

```

stb1 = syndtable(h);
tmp = bi2de(syndrom, 'left-msb');
z = stb1(tmp+1,:);
rez = xor(z,m)

```

Сначала мы нашли генераторную матрицу g и проверочную матрицу h . При умножении msg на g мы получили сообщение в коде Хэмминга. Для того, чтобы избавиться от двоек, было произведено вычисление остатка от деления на 2. Затем совершаем ошибку в 4 бите. Далее вычислим синдром и с его помощью исправляем нужный бит. В итоге получаем первоначальное сообщение.

`msg =`

```

1      1      0      0

```

`m =`

```

1      0      1      1      1      0      0

```

`rez =`

```

1      0      1      1      1      0      0

```

3.2 Циклический код

Циклический код - это линейный блочный код, обладающий свойством цикличности: циклическая перестановка кодового слова является также кодовым словом.

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

code = msg*g;
code = rem(code, ones(1,7).*2)
code(3) = not(code(3)); %mistake

syndrom = code * h';
syndrom = rem(syndrom, ones(1, 3).*2);

stb1 = syndtable(h);
tmp = bi2de(syndrom, 'left-msb');
z = stb1(tmp+1,:);
res = xor(code, z)
```

С помощью функции `cyclpoly` генерируется вектор коэффициентов, а с помощью функции `cyclgen` мы получаем проверочную и генераторную матрицы. Совершив ошибку в третьем бите, мы все равно получаем правильное исходное сообщение. Количество ошибок, которые может исправить циклический код, также равно единице.

```
msg =

    1    1    0    0

code =

    0    1    0    1    1    0    0

res =

    0    1    0    1    1    0    0
```

3.3 Коды БЧХ

БЧХ код - это циклический код, применяемый для защиты информации от ошибок. Отличается возможностью построения кода с заранее определёнными корректирующими свойствами, а именно, минимальным кодовым расстоянием.

Минимальное кодовое расстояние d - это минимальное количество искаженных символов, необходимое для перехода из одной разрешенной комбинации в другую.

```
M = 4;
n = 2^M-1; % Codeword length
k = 5; % Message length
nwords = 10; % Number of words to encode
msgTx = gf(randi([0 1],nwords,k))
code = bchenc(msgTx, n, k)
t = bchnumerr(n,k)
noisycode = code + randerr(nwords,n,t);
res = bchdec(noisycode, n, k)
isequal(msgTx,res)
```

Функция `bchnumerr` служит для определения количества символов, которые мы можем исправить. Если при генерации `noisycode` (кода с t ошибок) сгенерировать не t , а $t+1$ ошибок, то декодированное сообщение не будет совпадать с исходным. В нашем случае $t=3$.

3.4 Коды Рида-Соломона

недвоичные циклические коды, позволяющие исправлять ошибки в блоках данных. Элементами кодового вектора являются не биты, а группы битов (блоки). Очень распространены коды Рида — Соломона, работающие с байтами (октетами). Код Рида — Соломона является частным случаем БЧХ-кода.

```
nwords = 3;
m = 3;           % Number of bits per symbol
n = 2^m - 1;     % Codeword length
k = 3;           % Message length

msg = gf(randi([0 2^m-1], nwords, k), m)
code = rsenc(msg, n, k)
errors = 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 + errors;
[res, err] = rsdec(code, n, k)
```

В данном случае мы используем код, состоящий из 3 посылок по 3 блока по 3 бита в каждом. Для кодирования будет использовано по 7 блоков из трех бит на посылку.

В результате выполнения программа вернет нам массив [1 2 -1], означающий, что в первой посылке была одна ошибка, во второй две, а в третьей больше двух ошибок.

4 Вывод

В ходе выполнения данной работы были изучены основные методы кодирования сообщений помехоустойчивыми кодами. Код Хэмминга прост в использовании, но при этом обнаружить и исправить мы можем только одну ошибку.

Циклические коды обеспечивают более простое декодирование. Кодирование циклическим кодом осуществляется с помощью порождающего полинома. Подклассами циклического кода являются БЧХ коды и коды Рида-Соломона.

БЧХ-кодирование удобно тем, что мы можем сами определить минимальное кодовое расстояние,

Код Рида-Соломона настоящее время широко используется в системах восстановления данных с компакт-дисков, при создании архивов с информацией для восстановления в случае повреждений, в помехоустойчивом кодировании.