

Санкт-Петербургский Государственный Политехнический Университет
Институт Информационных Технологий и Управления
Кафедра Компьютерных Систем и Программных Технологий

Отчёт по лабораторной работе №7
на тему
Помехоустойчивые коды

Работу выполнила
Студентка группы 33501/1
Михалёва М.В.
Преподаватель
Богач Н.В.

1 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

2 Теоретическое обоснование

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.

Значительную долю кодов составляют **блочные коды**. При их применении передаваемое сообщение разбивается на блоки одинаковой длины, после чего каждому блоку сопоставляется код в соответствии с выбранным способом кодирования.

Другая характеристика, позволяющая выделить коды в отдельный класс - цикличность. У кодов этого класса циклическая перестановка букв слова также является кодовым словом.

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.

Кодировку сообщения производят с помощью генераторной матрицы, домножение на которую столбца создает кодированное сообщение. На приемной стороне сообщение домножается на проверочную матрицу, полученный результат называется синдромом и позволяет определить наличие ошибок и их местоположение, если корректирующая способность кода достаточна.

3 Ход работы

3.1 Код Хэмминга

Произведем кодирование/декодирование сигнала кодом Хэмминга с помощью встроенных функций encode/decode.

```
message = [0 1 1 0]
```

```
code = encode(message,7,4)
```

```
code(3) = not(code(3))
```

```
[dec,err] = decode(code,7,4)
```

При передаче сообщения без ошибки количество ошибок равно нулю. При допущении ошибки в 3 бите обнаруживается одна ошибка $err = 1$, которая исправляется и сообщение декодируется верно.

```
code =
```

```
1    0    0    0    1    1    0
```

```
code =
```

```
1      0      1      0      1      1      0
```

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

```
message = [0 1 1 0]
[h,g,n,k] = hamngen(3)

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

m(3) = not(m1(3))
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:

```
mes = 1      0      \textbf{1}      0      1      1      0
z   = 0      0      \textbf{1}      0      0      0      0
rez = 1      0      \textbf{0}      0      1      1      0
```

Исправляющая способность кода равна 1.

3.2 Циклический код

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

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

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

code(4) = not(code(4))

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)
```

Сначала строится порождающий полином циклического кода: $x^3 + x + 1$. Далее, используя этот полином в качестве одного из параметров функции `cyclgen`, получили порождающую и проверочную матрицы для данного кода. В результате сообщение было закодировано следующим образом: [0 0 1 0 1 1 0]. Допущенная ошибка в 4 разряде была успешно обнаружена и исправлена синдромом.

Исправляющая способность кода равна 1.

3.3 Коды БЧХ

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

```
msg = [0 1 1 0]
```

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

```
code(4) = not(code(4))
decode = step (decbch , code')'
```

Закодированное сообщение: [0 1 1 1 0 0 1] Допущенная ошибка в 4 разряде была успешно обнаружена и исправлена.

Исправляющая способность кода равна 1.

При $k = 7$, $n = 15$ корректирующая способность кода БЧХ стала равна 2, что позволило исправлять 2 ошибки.

3.4 Коды 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.

```
l = 3;
n = 7;
k = 3;
m = 3;

msg = gf(randi([0 2m-1],l,k),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)
```

В первом слове была допущена одна ошибка, во втором и третьем по две. Исходные посылки:

```
[ 3      7      5]
[ 2      1      3]
[ 6      1      7]
```

Закодированные посылки:

```
[ 3      7      5      0      1      6      0]
[ 0      1      3      6      2      4      5]
[ 5      5      2      4      0      2      3]
```

В результате все ошибки были успешно обнаружены и исправлены. Корректирующая способность кода равна 2.

4 Вывод

В ходе данной работы были получены навыки кодирования цифровых сигналов. Кодирование таких сигналов происходит по принципу избыточности. В зависимости от различных задач следует использовать разные коды. Код Хэмминга просто и быстро реализуем, но он предлагает исправления одной ошибки, тогда как код Рида-Соломона способен оперировать десятичными числами и способен параллельно обрабатывать несколько потоков.