

Отчет по лабораторной работе №7

Мокроусов В.Д.

25 мая 2018 г.

Помехоустойчивое кодирование.

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, кодом БЧХ, кодом Рида-Соломона. Оценить корректирующую способность кода.

3 Теоретический раздел

Помехоустойчивое кодирование Существенно сократить избыточность в передаваемых сообщениях позволяет помехоустойчивое кодирование. В общем плане помехоустойчивое кодирование можно понимать как такое кодирование сообщений, при котором элементы связаны определенной зависимостью, позволяющей при ее нарушении указать ошибки и восстановить информацию. Помехоустойчивые коды рассчитаны на определенные ошибки. Это значит, что при других ошибках они могут оказаться недостаточно эффективными.

Сообщения по каналу связи передаются в виде кодограммы. Кодограммой, или кодовой комбинацией, называется упорядоченный набор k элементов, каждый из которых может принимать m значений. Множество всех кодовых комбинаций, поставленных в соответствие сообщениям, называется k -разрядным кодом с основанием m . Таким образом, код - множество кодовых комбинаций, а не одна комбинация.

Коды могут быть:

1. Безыбыточными - комбинации ставятся в соответствие каким-то сообщениям. В безыбыточном коде искажение любого элемента приводит к перерождению кодовой комбинации, т.е. изменению смыслового содержания сообщения.

2. Избыточными - использованы не все возможные комбинации. В этих кодах искажение элемента не всегда приводит к искажению сообщения. (Так, при использовании двоичного кода в случае приема комбинации 1100 неизвестно, какая цифра передана, но и ошибки в распознавании не происходит, так как такой комбинации в коде нет. Ошибка в этом случае обнаруживается. Скорее всего переданы комбинации 1000 или 0100, поскольку они "ближе" к принятой.) Пример избыточного кодирования показан в вышеприведенной таблице.

Идея помехоустойчивого кодирования заключается как раз в таком разнесении кодовых комбинаций, за счет введения избыточности, при котором искажения элементов не приводит к перерождению комбинаций.

Избыточные коды иногда называют корректирующими. Корректирующую способность оценивают минимальным кодовым расстоянием, которое жестко связано с числом исправляемых или обнаруживаемых ошибок. Минимальное кодовое расстояние - это число разрядов, по которым отличаются кодовые комбинации. Обозначается d .

Коды Хэмминга

Код Хэмминга - наиболее известный из первых самоконтролирующихся и самокорректирующихся кодов. Построен применительно к двоичной системе счисления. Позволяет исправлять одиночную ошибку (ошибка в одном бите) и находить двойную. Построение кодов Хэмминга основано на принципе проверки на четность числа единичных символов: к последовательности добавляется такой элемент, чтобы число единичных символов в получившейся последовательности было четным.

Циклические коды

Циклический код - линейный, блочный код, обладающий свойством цикличности, то есть каждая циклическая перестановка кодового слова также является кодовым словом. Используется для преобразования информации для защиты её от ошибок. Циклические коды незаменимы при необходимости передавать информацию в каналах связи, в которых отсутствует возможность повторной передачи данных. Циклические коды применяются при записи и считывании на HDD, CD и DVD, при использовании USB-портов для обмена информацией, при передаче аудио и видео информации. Среди всего многообразия групповых кодов можно выбрать такие, у которых строки связаны условием цикличности, т.е. все строки матрицы могут быть получены циклическим сдвигом одной строки, которая называется образующей или производящей. Сдвиг осуществляется справа налево, а крайний левый символ перемещается в конец строки, т.е. в крайнее правое положение. Коды, у которых строки

матрицы удовлетворяют этому условию, называются циклическими.

Коды БЧХ

Коды Боуза — Чоудхури — Хоквингема (БЧХ-коды) — в теории кодирования это широкий класс циклических кодов, применяемых для защиты информации от ошибок. Отличается возможностью построения кода с заранее определёнными корректирующими свойствами, а именно, минимальным кодовым расстоянием. Частным случаем БЧХ-кодов является код Рида — Соломона. БЧХ-код является циклическим кодом, который можно задать порождающим полиномом. Для его нахождения в случае БЧХ-кода необходимо заранее определить длину кода и требуемое минимальное расстояние.

Коды Рида-Соломона

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

Свёрточный код

Свёрточные коды это коды, исправляющие ошибки, которые используют непрерывную, или последовательную, обработку информации короткими фрагментами (блоками). Сверточный кодер обладает памятью в том смысле, что символы на его выходе зависят не только от (очередного фрагмента) информационных символов на входе, но и предыдущих символов на его входе. Другими словами, кодер представляет собой последовательную машину или автомат с конечным числом состояний. Состояние кодера определяется содержимым его памяти.

4 Ход работы

4.1 Код Хэмминга

С помощью встроенных функций encode/decode произведём кодирование/декодирование сигнала кодом Хэмминга.

```
%Код Хэмминга
```

```
mes = [1 0 1 0]
code_mes = encode(mes, 7, 4)
code_mes(4) = not(code_mes(4))
dec_mes = decode(code_mes, 7, 4)

if dec_mes == mes
    disp('SUCCESS');
else disp('ERROR');
end
```

В результате работы программы заданная посылка была сначала закодирована, затем была допущена одна ошибка в 4 символе, после чего с помощью встроенной функции decode закодированная посылка с ошибкой была успешно декодирована, в следствие чего мы получили исходное сообщение [1 0 1 0].

```
mes =      1      0      1      0
code_mes = 0      0      1      1      0      1      0
code_mes = 0      0      1      0      0      1      0
dec_mes =  1      0      1      0
```

```
SUCCESS
```

Если сделать больше одной ошибки в закодированном сообщении, то код Хэмминга не сможет исправить ошибки.

```
%Код Хэмминга
```

```
mes = [1 0 1 0]
code_mes = encode(mes, 7, 4)
code_mes(1) = not(code_mes(1))
code_mes(4) = not(code_mes(4))
```

```

dec_mes = decode(code_mes, 7, 4)

if dec_mes == mes
    disp('SUCCESS');
else disp('ERROR');
end

```

Результат выполнения изменённой программы

```

mes =      1      0      1      0
code_mes = 0      0      1      1      0      1      0
code_mes = 1      0      1      1      0      1      0
code_mes = 1      0      1      0      0      1      0
dec_mes = 0      0      1      0

```

ERROR

В результате получен ошибочный результат.

Произведём кодирование/декодирование сообщения кодом Хэмминга путём создания генераторной и проверочной матриц, а также вычислим синдром.

%Генераторная и проверочная матрица

```

mes = [1 0 1 0]
[par, g, n, k] = hammgen(3);

r = mes*g
r = rem(r, ones(1,n).*2)

r(2) = not(r(2))

syndr = r*par'
syndr = rem(syndr, ones(1, n-k).*2)

t = syndtable(par);

syndr_de = bi2de(syndr, 'right-msb')

corrvect = t(syndr_de + 1,:)
correctedcode = rem(corrvect + r,2)

```

При умножении исходного сообщения на генераторную матрицу в ее конечной части сохраняется исходная посылка, т.к. соответствующий блок генераторной матрицы представляет собой единичную матрицу. Оставшаяся часть формирует контрольные биты, которые составляют дополнение до нуля суммы по модулю два нужных информационных разрядов. Формирование синдрома происходит с помощью домножения на проверочную матрицу par' . С помощью синдрома был выявлен ошибочный бит в посылке, в нашем случае 2. Затем он был исправлен.

Результат выполнения программы.

```
mes = 1      0      1      0
r =  0      0      1      1      0      1      0
r =  0      1      1      1      0      1      0
syndr = 0      1      0
syndr_de = 1
corrvect =      0      1      0      0      0      0      0
correctedcode = 0      0      1      1      0      1      0
```

Корректирующая способность кода равна 1

4.2 Циклический код

Сообщение [1 0 1 0]. Произведено кодирование/декодирование этого сообщения:

```
%Циклический код
```

```
n = 7;
```

```
k = 4;
```

```
mes = [1 0 1 0]
```

```
gp = cyclpoly(n,k, 'max')
```

```
[h,g] = cyclgen(7, gp);
```

```
code = mes*g;
```

```
code = rem(code, ones(1,n).*2)
```

```
code(2) = not(code(2))
```

```
syndr = code * h'
```

```
syndr = rem(syndr, ones(1, n-k).*2)

t = syndtable(h)
syndr_de = bi2de(syndr, 'right-msb')
corrvect = t(syndr_de + 1,:)
correct = rem(corrvect + code, 2)
```

Построен полином циклического кода

$$x^3 + x^2 + 1 \quad (1)$$

, который использовался в качестве параметра функции `cyclgen`. Получены порождающая и проверочная матрицы. Ошибка была произведена в 2 разряде. Результат выполнения программы

```
mes = 1      0      1      0
gp = 1      1      0      1
code = 0      0      1      1      0      1      0
code = 0      1      1      1      0      1      0
syndr = 0      1      0
syndr_de = 2
corrvect = 0      1      0      0      0      0      0
correct = 0      0      1      1      0      1      0
```

Корректирующая способность кода равна 1

4.3 Коды БЧХ

Произведено кодирование/декодирование сообщения [1 0 1 0] при помощи кодов БЧХ

```
%Коды БЧХ
mes = [1 0 1 0]

code_mes = comm.BCHEncoder(7,4)
dec_mes = comm.BCHDecoder(7,4)
tmp = mes';
code = step(code_mes, tmp(:))'

code(5) = not(code(5))
decode = step(dec_mes, code')'
```

Результат выполнения программы.


```

mes = 1      0      1      0
code =1      0      1      0      0      1      1
code =1      0      1      0      1      1      1
decode = 1    0      1      0

```

Допущенная во 5 разряде ошибка успешно обнаружена и исправлена.
Корректирующая способность кода равна 1.

4.4 Коды Рида-Соломона

Сгенерировано три слова по три символа. Допущены ошибки: одна ошибка в первом слове, две во втором и три в третьем.

```
%Коды Рида-Соломона
```

```
m = 3;
```

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

```
k = 3;
```

```
mes = gf([2 7 3; 4 0 6; 5 1 1], m);
```

```
code = rsenc(mes, n, k);
```

```
error = gf([2 0 0 0 0 0 0; 3 4 0 0 0 0 0; 5 6 7 0 0 0 0], m);
```

```
error_code = code + error;
```

```
[dec_code, cnum] = rsdec(error_code, n, k);
```

```
cnum
```

```
cnum
```

Результат выполнения программы.

```
cnum =
```

```

1
2
-1

```

Количество ошибок в первых двух строчках совпадает с количеством сделанных ошибок в error. В третьей строчке -1, так как коды Рида-Соломона не могут исправлять больше двух ошибок.

Получаем, что корректирующая способность кода равна 2.

5 Выводы

В ходе данной лабораторной работы рассмотрены такие методы кодирования, как: коды Хэмминга, циклические коды, коды Боуза-Чоудхури-Хоквингема и коды Рида-Соломона. Они являются самокорректирующимися. Выбор кодирования осуществляется в зависимости от поставленной задачи. Код Рида-Соломона имеет наибольшую корректирующую способность, а так же позволяет оперировать с десятичными числами и параллельно обрабатывать несколько потоков данных.