

## КОМПЛЕКСНОЕ ПРЕОБРАЗОВАНИЕ СИМВОЛОВ ПЕРЕМЕЖИТЕЛЯ И МЕТОД СЖАТИЯ LZ

The article is devoted to complex transformation of symbols in the interleaver block in turboencoding system and to a way of compression mixed entrance and blocks of data. Transformation of symbols is based on pseudo-non-uniform interleaver and a method of compression LZ compensates delays in the turboencoding scheme and enables increases in the interleaver block.

Наиболее заметным достижением в теории помехоустойчивого кодирования за последнее десятилетие безусловно является изобретение турбокодов. Впервые они были описаны в 1993 г. и несмотря на очень большой выбор помехоустойчивых кодов для создания новых систем связи эти коды спустя всего 5–6 лет после своего рождения получили «прописку» как в современных стандартах радиосвязи с космическими объектами, так и в стандартах систем мобильной связи третьего поколения. Причина этому – уникальная способность турбокодов обеспечивать характеристики помехоустойчивости передачи информации по каналам со случайным характером ошибок типа аддитивный белый гауссовский шум или в каналах с постепенно исчезающим сигналом (замирание) при умеренной сложности оборудования для кодирования и декодирования. Рассматриваемые каналы относятся к числу каналов с кодовым разделением (CDMA – Code Division Multiple Access) [1].

Главной особенностью турбокода является наличие двух или более кодеров рекурсивных сверточных кодов (РСК) и устройств перемежения (рисунки).



Рисунок. Структурная схема устройства турбокодирования

Турбокод представляет собой систематический код, в котором проверочная группа образуется из проверочных битов, генерируемых двумя или более составными кодерами рекурсивных сверточных кодов, причем информационная последовательность подается в кодер первого рекурсивного сверточного кода непосредственно, а в кодер второго рекурсивного сверточного кода – через блок перемежения.

Перемежение выполняется процессором, вызывающим алгоритм перемешивания. Объединение перемешанных блоков данных со входными используется для снижения частоты ошибочных бит в рассматриваемом канале. Процесс перемежения увеличивает разнородность данных, поступающих в кодер, что приводит к исправлению ошибок в декодере (на приемной стороне) с помощью алгоритма исправления, если символы искажаются при передаче [2].

Блок перемежения функционирует на основе матрицы, строки которой формируются последовательностью входных символов. Затем определенное число бит образует блок [3].

Предлагаемый алгоритм перемежения основывается на использовании простых чисел. Для начала выбираются взаимно простые числа  $p_1$  и  $q_1$ , такие, что  $n_1 = p_1 \cdot q_1$ . Вычисляется  $t_1 = (p_1 - 1)(q_1 - 1)$ . Находятся два простых числа ( $e_1$  и  $d_1$ ), таких, чтобы  $e_1 \cdot d_1 \bmod t_1 = 1$ . Перемешивание множества строк  $s_i$  ( $1 \leq i \leq k$ ) осуществляется на основе формулы

$$s_i = s_i^{d_1} \bmod n_1. \quad (1)$$

Например,  $p_1 = 3$  и  $q_1 = 5$ . Тогда  $n_1 = 3 \cdot 5 = 15$ ,  $t_1 = 2 \cdot 4 = 8$ . Выбираем  $e_1 = 3$  и  $d_1 = 3$  для выполнения условия  $3 \cdot 3 \bmod 8 = 1$ . Тогда, для примера, определим место расположения второй строки после выполнения процедуры перемежения:

$$s_2 = 2^3 \bmod 15 = 8.$$

Таким образом, вторая строка матрицы после перемешивания станет восьмой.

Второй этап перемешивания связан с обработкой столбцов. Местоположение столбца  $st_j$  ( $1 \leq j \leq l$ ) после перемешивания определяется согласно формуле

$$st_j = st_j^{d_2} \bmod n_2, \quad (2)$$

где  $n_2$  – произведение взаимно простых чисел  $p_2$  и  $q_2$ . Причем  $t_2 = (p_2 - 1)(q_2 - 1)$ . Числа  $d_2$  и  $e_2$  выбирают так, чтобы выполнялось условие

$$d_2 \cdot e_2 \bmod t_2 = 1. \quad (3)$$

После перемежения строк и столбцов местоположение данных меняется [3].

В блоке декодирования турбокодов используются конструкции деперемежителя данных. Деперемежитель, соответствующий пере-

межителю, может быть основан на использовании перестановочных последовательностей:

$$s_i = s_i^{e1} \bmod n_1; \quad (4)$$

$$st_j = st_j^{e2} \bmod n_2. \quad (5)$$

В соответствии с идеальной ситуацией количество столбцов и строк надо выбрать так, чтобы расстояние (разница в индексах символов входной последовательности) между соседними битами после перемежителя было наибольшим (для более эффективного исправления ошибок в декодере).

Из-за неконтролируемого увеличения числа строк образуется сигнальная задержка в схеме. В результате необходимо ограничивать размеры перемежителя с целью обеспечения задержки сигналов в допустимых пределах. С другой стороны, уменьшение размеров матрицы ограничивает разделение сигналов на большее расстояние, а следовательно – снижает эффективность коррекции ошибок в декодере.

Для увеличения блока перемежения с целью лучшего перемешивания данных и компенсации временной задержки, возникающей с этим процессом, предлагается использование алгоритма сжатия LZ после смешивания входных и перемешанных блоков данных. Сжатие сокращает объем информации и количество времени, необходимого для передачи информации по каналу связи установленной ширины пропускания.

В основе словарного кодировщика LZ лежит метод сжатия заменой групп последовательных символов (фраз) индексами некоторого словаря. Словарь есть список таких фраз, которые, как ожидается, будут часто использоваться. Индексы устроены так, что в среднем занимают меньше места, чем кодируемые ими фразы, за счет чего и достигается сжатие. Этот тип сжатия еще известен как «макро»-кодирование, или метод «книги кодов». Словарные методы обычно быстры, в частности по тем причинам, что один код на выходе соответствует нескольким входным символам и размер кода обычно соответствует машинным словам.

Словарная модель LZ дает достаточно хорошее сжатие. Нужно сказать, что отличительной чертой большинства словарных кодировщиков является экономия ресурсов.

Алгоритм сжатия LZ принадлежит семье алгоритмов, происходящих из работы Лемпеля и Зива [4]. Сущность состоит в том, что фразы заменяются указателем на то место, где они в тексте уже ранее появлялись. Это семейство алгоритмов называется методом Зива – Лемпеля и обозначается как LZ-сжатие. Данный метод быстро приспосабливается к структуре тек-

ста и может кодировать короткие функциональные слова, т. к. они очень часто в нем появляются. Новые слова и фразы могут также формироваться из частей ранее встреченных слов.

Раскодирование сжатого текста осуществляется напрямую – происходит простая замена указателя готовой фразой из словаря, на которую тот указывает. На практике LZ-метод добивается хорошего сжатия, его важным свойством является очень быстрая работа раскодировщика.

Одна из форм такого указателя – пара  $(m, l)$ , которая заменяет фразу из  $l$  символов, начинающуюся со смещения  $m$  во входном потоке. Например, указатель  $(7, 2)$  адресует 7-й и 8-й символы исходной строки. Используя это обозначение, строка «*abbaabbbabab*» будет закодирована как «*abba(1, 3)(3, 2)(8, 3)*».

Рассмотрим нагляднее процесс переработки потока информации длиной 196 бит. Допустим эта последовательность выглядит так:

```
01000101010111101010100010001110101000101
01011001110011100111101101010001100101001
01110001101001010101011010101100101000111
01000110011010000111010001100100010011001
01010001101010100011110000110101
```

Согласно псевдонеровномерному методу перемежения символов поступающая информация в блок перемежения записывается построчно – образуя матрицы размером  $14 \times 14$ . В блоке идет процесс перемежения данных по блокам. Сначала обрабатывается строки, а затем столбцы. Из блока перемежения вывод данных осуществляется по столбцам. На выходе измененная последовательность принимает вид

```
00010001001111100110000100001011010001000
11010111100100010000101101100111010000000
01011011000101100110000000011110101111010
10111101000001111010111101010010000111101
11101010001000101101111010000001
```

Применяя к выходной последовательности метод сжатия LZ, получаем

```
0001(1, 4)0011111(9, 4)0(1, 7)0101101(1, 8)1010
(12, 7)(1, 7)(28, 7)(8, 6)(24, 6)000(62, 8)(81, 12)
(77, 9)(42, 11)(44, 9)1(105, 17)(51, 9)111(112, 9)
(1, 9)(160, 9)(79, 7).
```

После сжатия данных последовательность составила 149 бит, это на 47 бит меньше чем исходная.

Таким образом, применение к выходной последовательности блока перемежения метода сжатия LZ дает максимум 70% сжатия инфор-

мации при минимальном блоке перемежения в 196 бит. При размерах блока перемежения, равных 2048 бит, сжатие составляет максимум 50% от исходных данных.

Процесс разархивации проходит за один проход, не требует больших ресурсов, и временных задержек в этом процессе не будет.

Исходя из частотной и канальной спецификации стандарта CDMA следует, что нагрузочная способность канала на одно телефонное соединение составляет 9600 бит в секунду и пользовательские данные группируются в фреймы длительностью 20 мс [5].

Отсюда следует, что пропускная способность каналов CDMA имеет ограничение и при увеличении блока перемежения без использования дополнительных методов преобразования информации приводит к сбоям в работе схемы и к появлениям временных задержек.

Качественный анализ показал, что комплексное преобразование символов перемежителя и алгоритма сжатия LZ дает возможность решать эти проблемы. Использование такого преобразования позволяет увеличить размер блока перемежения до 2 раз, и соответственно повысить пропускную способность канала. Увеличение пропускной способности канала осуществляется не за счет расширения самого канала, а за счет передачи по нему сжатой информации.

Временная задержка, вызванная увеличением блока перемежения, компенсируется путем подачи в канал передачи данных сжатой информации. Так как процесс архивации (сжатия) и разархивации происходит в один проход и не требует больших ресурсов, временных задержек в этих процессах не будет.

## Литература

1. Урбанович П. П., Пацей Н. В., Шиман Д. В., Мишин К. С. Структура графического интерфейса и функциональной спецификации программного средства моделирования сигнально-кодовых конструкций на основе турбо кода // Материалы МНТК «Организационно-техническое управление в межотраслевых комплексах» 28–29 октября, 2004 г. – Мн.: БГТУ, 2004. – С. 217–219.

2. Зяблов В. В., Цветков М. А. Дистанционные свойства турбокодов с различными перемежителями // Информационные процессы. – 2003. – Т. 3. – № 2. – С. 83–96.

3. Шиман Д. В. Метод псевдонеравномерного перемежения символов турбокода // Труды БГТУ. Сер. VI. Физ.-мат. науки и информ. – 2005. – Вып. XIII. – С. 168–170.

4. J. Ziv and A. Lempel. A universal algorithm for sequential data compression. IEEE Transactions on Information Theory. – Vol. IT-23, N 3, May 1977. – P. 337–343.

5. Орлов С. Технология CDMA особенности и преимущества // ChipNews. – 2000. – № 4.

6. Невдяев Л. CDMA: кодирование и перемежение // Сети. – 2000. – № 12.

7. Смородинов А. А. Перспективы использования турбо кодирования в системах передачи информации // Труды Третьей международной школы-семинара БИКАМП'01. – СПб. – 25–29 июня 2001. – С. 188–191.

8. Dolinar S., Divsalar D., Pollara F. Code performance as a function of block size // The Telecommunications and Mission Operations Progress Report 42-133, May, 1998. Jet Propulsion Laboratory, Pasadena, California. – May 15, 1998. – P. 1–23.