

МЕТОД ПСЕВДОНЕРАВНОМЕРНОГО ПЕРЕМЕЖЕНИЯ СИМВОЛОВ ТУРБОКОДА

The article is devoted to a way of data hashing in the interleaver block in turbocoding system. Based on non-uniform interleaving and the given method as much as possible disperse data also shows the numerical theory high speed of interleaver and deinterleaver blocks.

Способы кодирования информации в каналах передачи, известные как кодовая модуляция, созданы для улучшения достоверности приема и характеризуются частотой ошибочных бит – Bit Error Rate, BER.

Доказано, что турбокодирование (кодирование и декодирование данных на основе турбокодов) является одним из высокопроизводительных способов коррекции ошибок для каналов со случайным характером ошибок типа аддитивный белый гауссовский (нормальный) шум или в каналах с постепенно исчезающим сигналом (замиранием) [1]. Рассматриваемые каналы относятся к числу каналов с кодовым разделением (CDMA – Code Division Multiple Access) [1].

Основа структуры устройства турбокодирования (рис.) – наличие перемежителя, который меняет порядок символов входного сигнала или передает блок данных на вход следующего кодирующего устройства. Перемежение выполняется процессором, вызывающим алгоритм перемешивания. Объединение перемеженных блоков данных со входными используется для снижения частоты ошибочных бит в рассматриваемом канале. Процесс перемежения увеличивает разнородность в данных, поступающих в кодер, что приводит к исправлению ошибок в декодере (на приемной стороне) с помощью алгоритма исправления, если символы искажаются при передаче.



Рис. Структурная схема устройства турбокодирования

Блок перемежения функционирует на основе матрицы, строки которой формируются последовательностью входных символов. Затем определенное число бит в зависимости от алгоритма перемежения образует блок. Из перемежителя данные передаются в блок ко-

дера с помощью последовательного вывода столбцов матрицы. В результате сигналы в расположенных рядом строках матрицы в потоке достоверных сигналов отличаются номерами.

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.

Существуют два основных метода перемежения: однородное (равномерное) и неоднородное (псевдонеравномерное). В случае однородного перемежения расстояние между двумя любыми соседними битами будет одинаковым. Это не является положительным качеством, поскольку результирующий выходной сигнал (на выходах обоих кодеров) характеризуется относительно невысоким минимальным расстоянием [2].

Неоднородное перемежение позволяет получить наилучшее «максимальное разбрасывание» данных и «максимальное упорядочивание» выходной последовательности информации. Это означает, что избыточность, вносимая двумя элементарными кодами, равномерно распределена в выходной последовательности турбокодера. Минимальное расстояние между символами в неоднородном перемежении при больших значениях блока увеличивается по сравнению с однородным перемежением. Основная проблема для неоднородного перемежения заключается в достижении достаточной неоднородности. К тому же условное перемежение требует существенного объема памяти в кодере. Матрицы условного перемежения также требуют компенсации задержек, которые ограничивают их применение в соответствии с современными требованиями.

Исходная матрица [14,14]

1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
29	30	31	32	33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98
99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126
127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154
155	156	157	158	159	160	161	162	163	164	165	166	167	168
169	170	171	172	173	174	175	176	177	178	179	180	181	182
183	184	185	186	187	188	189	190	191	192	193	194	195	196

Существует необходимость в разработке новых методов перемежения, которые бы увеличивали неоднородность данных и обеспечивали компенсацию минимальных задержек.

Предлагаемый здесь метод базируется на усовершенствованном алгоритме «разбрасывания» данных за счет обработки бит по частям. Блок перемежения содержит память, предназначенную для обработки и хранения блоков данных. Для удобной работы каждый символ помечается, блоки данных индексируются как множества строк и столбцов, ограниченных в размере. Произведение заранее установленных размеров строк и столбцов соответствует объему блока данных.

Процессор связан с памятью и используется для разделения блоков данных на части. Он также предназначен для генерации перемешивания последовательного множества частей и индексации множества строк.

Первоначальное расположение данных (входные биты) отражено в табл. 1 ($k=1=14$).

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 $t_1 = (p_1 - 1) \cdot (q_1 - 1)$. 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 вы-

полнения условия $3 \cdot 3 \bmod 8 = 1$. Тогда, для примера, определим место расположения второй строки после выполнения процедуры перемежения (перемешивания)

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

Таким образом, вторая строка матрицы после перемешивания станет восьмой.

Результаты первого этапа перемешивания (строк) входной последовательности представлены в табл. 2.

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. Итоговый результат отображен в табл. 3.

Данный метод обеспечивает лучшие характеристики разбрасывания, чем метод условного перемежения, даже при небольших блоках данных (последовательность в несколько сотен). Это достигается за счет использования разнообразного рассредоточения по сравнению с условным перемежением. Описанный способ перемешивания данных может быть использован не только при тубокодировании, а также в других системах.

Матрица [14,14] после перемежения строк

1	2	3	4	5	6	7	8	9	10	11	12	13	14
99	100	101	102	103	104	105	106	107	108	109	110	111	112
155	156	157	158	159	160	161	162	163	164	165	166	167	168
43	44	45	46	47	48	49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80	81	82	83	84
169	170	171	172	173	174	175	176	177	178	179	180	181	182
15	16	17	18	19	20	21	22	23	24	25	26	27	28
113	114	115	116	117	118	119	120	121	122	123	124	125	126
127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154
29	30	31	32	33	34	35	36	37	38	39	40	41	42
85	86	87	88	89	90	91	92	93	94	95	96	97	98
183	184	185	186	187	188	189	190	191	192	193	194	195	196

Таблица 3

Матрица [14,14] после перемежения столбцов

1	8	12	4	5	6	13	2	9	10	11	3	7	14
99	106	110	102	103	104	111	100	107	108	109	101	105	112
155	162	166	158	159	160	167	156	163	164	165	157	161	168
43	50	54	46	47	48	55	44	51	52	53	45	49	56
57	64	68	60	61	62	69	58	65	66	67	59	63	70
71	78	82	74	75	76	83	72	79	80	81	73	77	84
169	176	180	172	173	174	181	170	177	178	179	171	175	182
15	22	26	18	19	20	27	16	23	24	25	17	21	28
113	120	124	116	117	118	125	114	121	122	123	115	119	126
127	134	138	130	131	132	139	128	135	136	137	129	133	140
141	148	152	144	145	146	153	142	149	150	151	143	147	154
29	36	40	32	33	34	41	30	37	38	39	31	35	42
85	92	96	88	89	90	97	86	93	94	95	87	91	98
183	190	194	186	187	188	195	184	191	192	193	185	189	196

Аналитически установлено, что представленный перемежитель совместим с существующей системой турбокодирования и алгоритмами декодирования. Подобные перемежители позволяют достичь лучшей производительности без увеличения системной сложности схем.

В блоке декодирования турбокодов используются конструкции депеременителя данных. Депеременитель, соответствующий перемежителю, может быть основан на использовании перестановочных последовательностях:

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

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

Качественный анализ показал, что предложенный метод псевдонеравномерного переме-

живания улучшает неоднородность и компенсирует минимальные задержки в системах передачи информации на основе турбокодов с перемежением символов.

Литература

1. Урбанович П.П., Пацей Н.В., Шиман Д.В., Мишин К.С. Структура графического интерфейса и функциональной спецификации программного средства моделирования сигнальных конструкций на основе турбокода // 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. Patent EP1103102 H03M 13/27 Improved interleavers for turbo code (2 августа, 1999).