
Title: Optimal Rate Matching for Algebraic Interleaver

Source: Nortel Networks¹

0.0 Summary

In this contribution, we present optimal Rate Matching for Algebraic Interleaver.

¹ Contact Person: Catherine Gauthier, Nortel Networks
1 Place des Freres Montgolfier, 78042 Guyancourt, France
Tel: +33 1 39 44 57 47
Fax: +33 1 39 44 50 12
e_mail: gauth@nortel.com

1.0 Optimized Rate Matching Algorithms

1.1 The Siemens Rate Matching Puncturing Algorithm

In TSGR1#3(99)203, the optimized puncturing algorithm was proposed to design the offset position for the 1st interleaver on a un-permuted matrix. Such a design is based on the 2 criteria (1) maximize the puncturing distance of the coded transport block for a specific QoS transport. (2) evenly distribute the punctured bit across all the 10ms TTI blocks.

The Siemens algorithm can be summarized as follows:

- Compute the average puncturing distance $q_o = \lfloor N_R / |N_R - N_{\bar{R}}| \rfloor$
- If $(q_o) \bmod 2 = 0$, compute: $q = q_o - \gcd(q_o, K) / K$ else $q = q_o$
- Compute: $S(\lceil iq \rceil \bmod K) = \lceil iq \rceil \text{div} K$ for $i = 0, 1, 2 \dots K-1$
- Compute: $e_{offset}(i) = [2S(i)|N_R - N_{\bar{R}}| + N_R] \bmod (2N_R)$

Example: The optimized puncturing grid based on the Siemens algorithm is shown in Table 1, where $N_R = 18, N_{\bar{R}} = 15$.

TABLE 1. Optimized Puncturing Grid on Un-Permuted Matrix

	0	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

1.2 Optimized Puncturing on the Permuted Matrix

In what follows, we show that optimal puncturing can be achieved on the permuted matrix (the matrix is permuted by optimized Algebraic Interleaver).

The Algebraic Interleaver based on the 18x8 matrix can be summarized as:

- Perform the column permutation as:

$$I_C(j) = [7j + 2\text{ceil}(i/q_o) + 1] \bmod 8 \quad i = 1, 2 \dots 18, j = 1, 2 \dots 8$$

- Perform the row permutation as:

$$I_R(i) = [7i + 2j + 1] \bmod 18 \quad i = 1, 2 \dots 18, j = 1, 2 \dots 8$$

Following the example in section 1.1, the algebraic permutation of columns is listed in Table 2. Note that the column is block wise permuted. The block size spans q rows. Such that the arbitrary blockwise column permutation does not destroy the equal puncturing bits in each column. Based on the column permutation pattern. The column permuted matrix is listed in Table 3.

TABLE 2. Column Permutation by Algebraic Interleaver

	Ic(j)							
i	2	1	8	7	6	5	4	3
	2	1	8	7	6	5	4	3
	2	1	8	7	6	5	4	3
	2	1	8	7	6	5	4	3
	2	1	8	7	6	5	4	3
	2	1	8	7	6	5	4	3
	4	3	2	1	8	7	6	5
	4	3	2	1	8	7	6	5
	4	3	2	1	8	7	6	5
	4	3	2	1	8	7	6	5
	4	3	2	1	8	7	6	5
	4	3	2	1	8	7	6	5
	6	5	4	3	2	1	8	7
	6	5	4	3	2	1	8	7
	6	5	4	3	2	1	8	7
	6	5	4	3	2	1	8	7
	6	5	4	3	2	1	8	7
	6	5	4	3	2	1	8	7

The Table 4 lists the row permutation patterns, it is obvious that the row permutation will not destroy the property that each row will contain equal number of puncturing bits. The row permuted the matrix based on Table 3 is listed in Table 5.

TABLE 3. Optimized Puncturing Grid on Column-Permuted Matrix

	Ic(j)							
	1	0	7	6	5	4	3	2
	9	8	15	14	13	12	11	10
	17	16	23	22	21	20	19	18
	25	24	31	30	29	28	27	26
	33	32	39	38	37	36	35	34
	41	40	47	46	45	44	43	42
	51	50	49	48	55	54	53	52
	59	58	57	56	63	62	61	60
	67	66	65	64	71	70	69	68
	75	74	73	72	79	78	77	76
	83	82	81	80	87	86	85	84
	91	90	89	88	95	94	93	92
	101	100	99	98	97	96	103	102
	109	108	107	106	105	104	111	110
	117	116	115	114	113	112	119	118
	125	124	123	122	121	120	127	126
	133	132	131	130	129	128	135	134
	141	140	139	138	137	136	143	142

In Table 5, the we can see the optimal puncturing grid design on the un-permuted matrix remains optimal on the algebraic permuted matrix. However after the algebraic permutation, the fading distance of each column in Table 5 is optimized. Where the fading distance is defined as the adjacent bit distance of the each column.

TABLE 4. Row Permutation by Algebraic Interleaver

Ir(i)	10	12	14	16	18	2	4	6
	17	1	3	5	7	9	11	13
	6	8	10	12	14	16	18	2
	13	15	17	1	3	5	7	9
	2	4	6	8	10	12	14	16
	9	11	13	15	17	1	3	5
	16	18	2	4	6	8	10	12
	5	7	9	11	13	15	17	1
	12	14	16	18	2	4	6	8
	1	3	5	7	9	11	13	15
	8	10	12	14	16	18	2	4
	15	17	1	3	5	7	9	11
	4	6	8	10	12	14	16	18
	11	13	15	17	1	3	5	7
	18	2	4	6	8	10	12	14
	7	9	11	13	15	17	1	3
	14	16	18	2	4	6	8	10
	3	5	7	9	11	13	15	17

TABLE 5. Optimized Puncturing Grid on Row-Permuted Matrix

	j							
Ir(i)	75	90	107	122	137	12	27	42
	133	0	23	38	55	70	85	102
	41	58	73	88	105	120	143	10
	101	116	131	6	21	36	53	68
	9	24	47	56	79	94	111	126
	67	82	99	114	129	4	19	34
	125	140	15	30	45	62	77	92
	33	50	65	80	97	112	135	2
	91	108	123	138	13	28	43	60
	1	16	39	48	71	86	103	118
	59	74	89	106	121	136	11	26
	117	132	7	22	37	54	69	84
	25	40	57	72	95	104	127	142
	83	100	115	130	5	20	35	52
	141	8	31	46	63	78	93	110
	51	66	81	98	113	128	3	18
	109	124	139	14	29	44	61	76
	17	32	49	64	87	96	119	134

2.0 Sub-Optimum Block Puncturing

In order to simplify the Rate Matching algorithm in both UL and DL, it is possible to puncture the permuted matrix of the 1st Interleaver output. Suppose the input radio frame length is N_R and the desired radio frame length after puncturing or repetition is $N_{\bar{R}}$. The simplest approach is puncture/repeat the first $|N_R - N_{\bar{R}}|$ rows of the permuted matrix.

TABLE 6. Optimized Puncturing Grid on Row-Permuted Matrix

	Ic(j)							
Ir(i)	75	90	107	122	137	12	27	42
	133	0	23	38	55	70	85	102
	41	58	73	88	105	120	143	10
	101	116	131	6	21	36	53	68
	9	24	47	56	79	94	111	126
	67	82	99	114	129	4	19	34
	125	140	15	30	45	62	77	92
	33	50	65	80	97	112	135	2
	91	108	123	138	13	28	43	60
	1	16	39	48	71	86	103	118
	59	74	89	106	121	136	11	26
	117	132	7	22	37	54	69	84
	25	40	57	72	95	104	127	142
	83	100	115	130	5	20	35	52
	141	8	31	46	63	78	93	110
	51	66	81	98	113	128	3	18
	109	124	139	14	29	44	61	76
	17	32	49	64	87	96	119	134

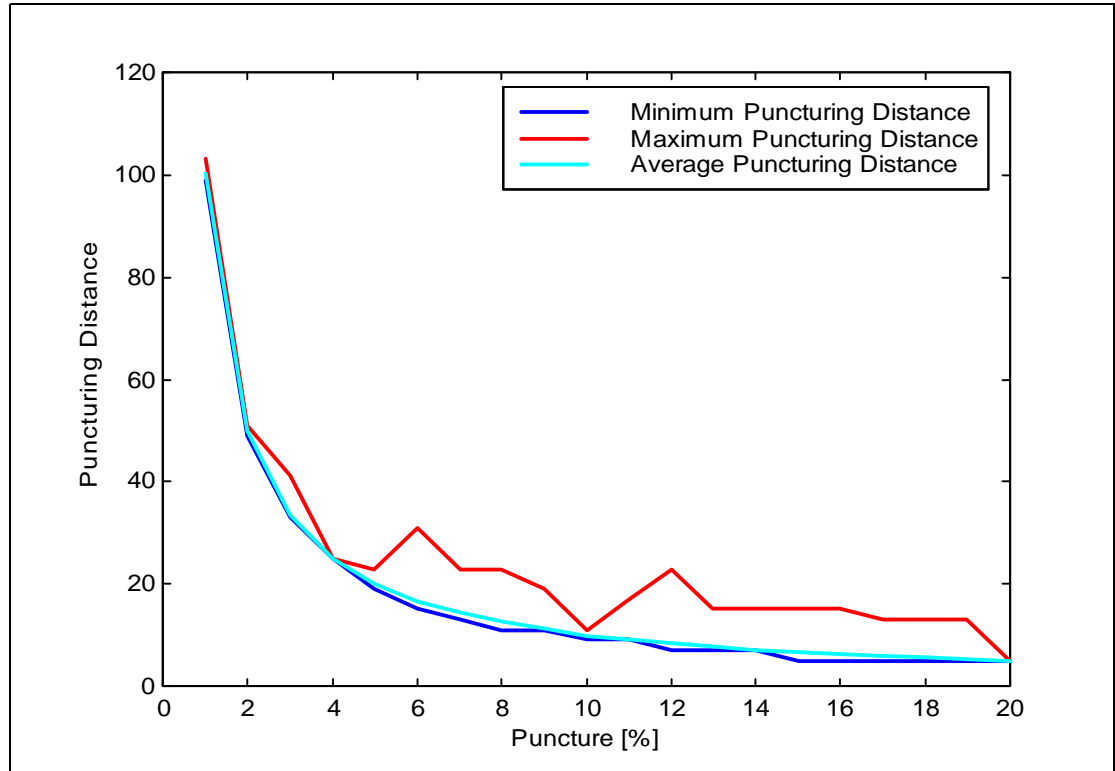
It can be shown that the average puncturing distance is the same as the optimum puncturing, however, such a simple puncturing scheme could result in lower minimum puncturing distance. On the other hand, the optimum puncturing algorithm is achieved by adding almost same complexity of channel interleaving.

3.0 Puncturing Distance and Fading Distance

3.1 The Puncturing Distance Evaluation

As shown in previous section, the optimal design of puncturing distance can be preserved after the algebraic permutation of the interleaver matrix. Figure 1 shows the puncturing distance profile for the combined Optimized 1st Interleaver and Optimized Puncturing Grid.

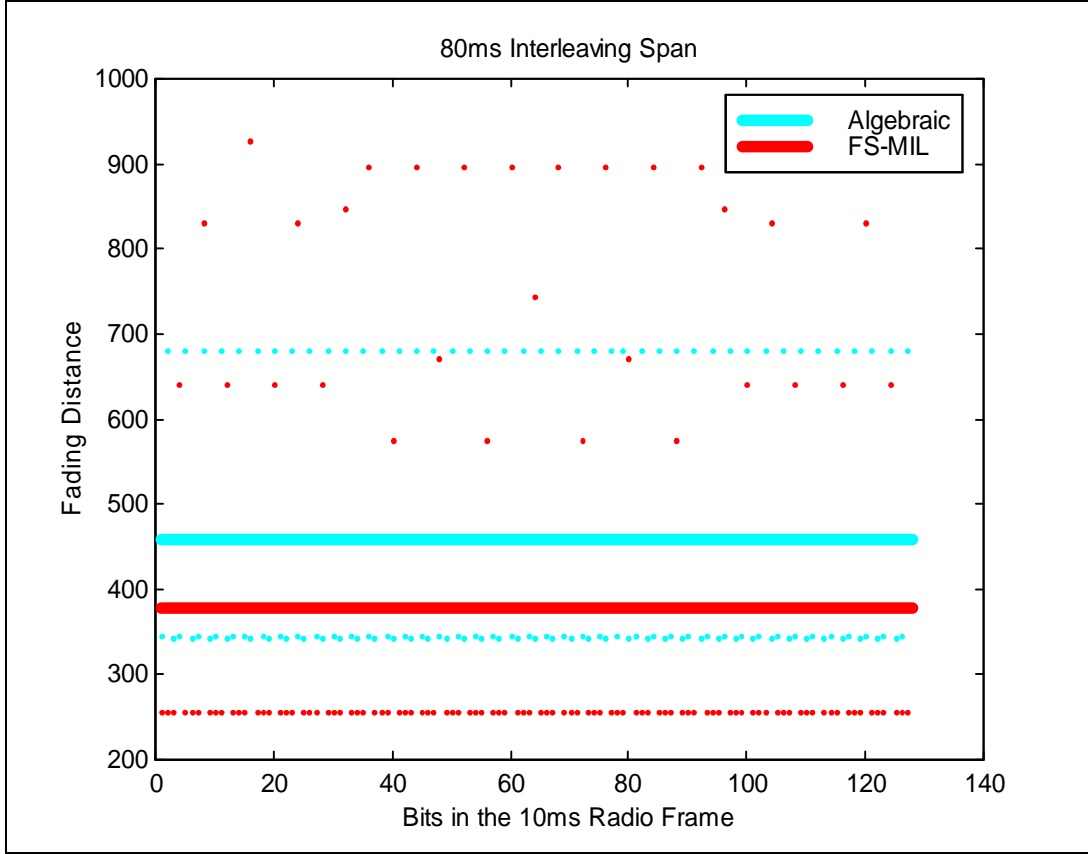
FIGURE 1. The Optimal Puncturing Distance Distribution



3.2 The Fading Distance Property of the 1st Stage Algebraic Interleaver

The optimized 1st interleaver can maximize the distance of consecutive adjacent bits of the each column of the permuted interleaver matrix. Also such an interleaver can randomized the bit position of the interleaved radio frame. These two properties will enhance the algebraic interleaver in the fading resistant ability than the conventional block interleaver. Figure 2 shows the fading distance for 1st algebraic interleaver and 1st&2nd stages for FS-MIL interleaver, where 128x8 matrix is interleaved. The fading distance is evaluated for 1st Algebraic Interleaver and combined 1st&2nd FS-MIL Interleaver. The bold lines are the average distance and the dots are the distance at each bit position for a 10ms radio frame. We can see the optimized Algebraic Interleaver is consistently better than FS-MIL Interleaver.

FIGURE 2. Fading Distance Comparison



Summary

In this contribution, we show that the optimal rate matching can be easily integrated into the 1st optimized algebraic interleaver while achieving superior fading resistant capability.