TSG-R1 3(99)249

Title: On the Algebraic Channel Interleaver Design

Source: Nortel Networks[1]

0.0 Summary

In this memo, we present a solution for Inter-Frame (1st) Interleaver and Intra-Frame (2nd) Interleaver design for 3GPP. The first interleaver is optimised and the second interleaver is a simple shuffling of bits from different Transport Channels.

The main advantages of the proposed solution are the following.

- very low complexity for both interleaving and de-interleaving (refer to Section 2.1 and 3.1)
- flexibility to cope with multiple frame sizes (refer to Section 2.1 and 3.2)
- robustness against fading (refer to Section 3.4)
- robustness to puncturing (refer to Section 3.3)
- future proofness due to the flexibility towards frame sizes and the use of a formula as interleaving rule instead of tables storage (refer to Section 2.1 and 3.5)

With respect to the listed advantages, we recommend the adoption of the described interleaver as channel interleaver for 3GPP.

---

[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@nortelnetworks.com

1.0 Introduction

In this memo, we propose to use a fully optimized Algebraic Interleaver to achieve maximum interleaving depth and time span at 1st interleaving stage. In order to support the Rate Matching after the 1st interleaving, such an Algebraic Interleaver possesses a good random spreading property to allow the straightforward Rate Matching (i.e. puncturing and repetition).

The design principles is to map the multiple coded-blocks from channel encoder output, of the TrCHs of the same QoS into a 2-dimensional matrix and then apply linear congruential rules to permute the rows and columns of such a matrix.

The maximum interleaving depth and time span can be achieved by searching a set of best parameters. The consecutive puncturing problem can be avoided by a proper design of the row and column permutation rules, it is straightforward to show that the Algebraic Interleaver can eliminate the problem of MIL 1st interleaving demonstrated in Fig.3 of Tdoc 299/98 and Fig2 in Ad_hoc-4/Ericsson.

In 3rd generation CDMA system, the multi-media services are multiplexed in transport channels to perform channel coding, interleaving and spreading. Based on the very flexible channel interleaver with optimum performance, we propose a universal service multiplexing and channel interlaving scheme.

2.0 Interleaver description

2.1 First Interleaver Algorithm

To make the Algebraic Interleaver simple, we use the following matrix permuting rules:

STEP-1: Convert a N_c 10 ms coded blocks (each with length N_r coded symbols) into a $N_r \times N_c$ matrix

STEP-2: Do matrix row and column permutation based on the following rules:

Row Permutation $I_r(k) = [\alpha_r k + ml] mod N_r$

Column Permutation $I_c(l) = [\alpha_c l] mod N_c$

where the parameter α_r is chosen as the largest prime number less than $\lfloor N_r/2 \rfloor$, the parameter α_c is chosen as the largest prime number less than $\lfloor N_c/2 \rfloor$, the parameter $m = \lceil N_r/N_c \rceil$.

STEP-3: Read out the matrix column-by-column.

Example-1: In this example, we chose the following parameters: and $N_c = 8$, $N_r = 10$, and $\alpha_r = 3$, $\alpha_c = 3$, $m = 1$.

STEP-1: Convert a $N_c = 8$ 10 ms coded blocks (each with length $N_r = 10$ coded symbols) into a 10×8 matrix

TABLE 1. Example: Formulate a Block Matrix

	l							
k	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

STEP-2a: Perform the row permutation

TABLE 2. Example: Row Permutation $(3l + k) mod 8$

25	34	43	52	61	70	79	8
49	58	67	76	5	14	23	32
73	2	11	20	29	38	47	56
17	26	35	44	53	62	71	80
41	50	59	68	77	6	15	24
65	74	3	12	21	30	39	48
9	18	27	36	45	54	63	72
33	42	51	60	69	78	7	16
57	66	75	4	13	22	31	40
1	10	19	28	37	46	55	64

STEP-2b: Perform the column permutation

TABLE 3. Example: Column Permutation $(3k + l) mod 10$

43	70	25	52	79	34	61	8
67	14	49	76	23	58	5	32
11	38	73	20	47	2	29	56
35	62	17	44	71	26	53	80
59	6	41	68	15	50	77	24
3	30	65	12	39	74	21	48
27	54	9	36	63	18	45	72
51	78	33	60	7	42	69	16
75	22	57	4	31	66	13	40
19	46	1	28	55	10	37	64

STEP-3: Read-out column-by-column $I(n) = [43, 67, 11, 35, 59\ldots, 40, 64]$

2.2 Second interleaver description

If the 2nd Interleaver is not designed properly, it can degrade the 1st Interleaver resulting in an overall poor performance. This issue has been identified and extensively discussed. One solution is to design 2nd Interleaving as shuffling bits evenly from different Transport Channels. Multiplexing of all TrChs, which occurs before this second interleaving, is assumed to consist of aggregation of bits of these TrChs in each 10ms block.

Since at 1st-stage interleaving, the interleavers are optimized for each Rate Matched data stream with different QoS, by using a simple shuffling operation which can interlace interleaved data with different QoSs, the spreading property of each QoS interleaved stream can be preserved. It will even be improved due to insertion of bits from others TrCHs inbetween bits of one TrCHs, puting the bits even further apart while not changing their order in one TrCh. Due to the randomization effect of Algebraic interleaving, multiplexing, shuffling, physical channel segmentation and DPCH mapping will not degradate the 1st stage interleaving performance.

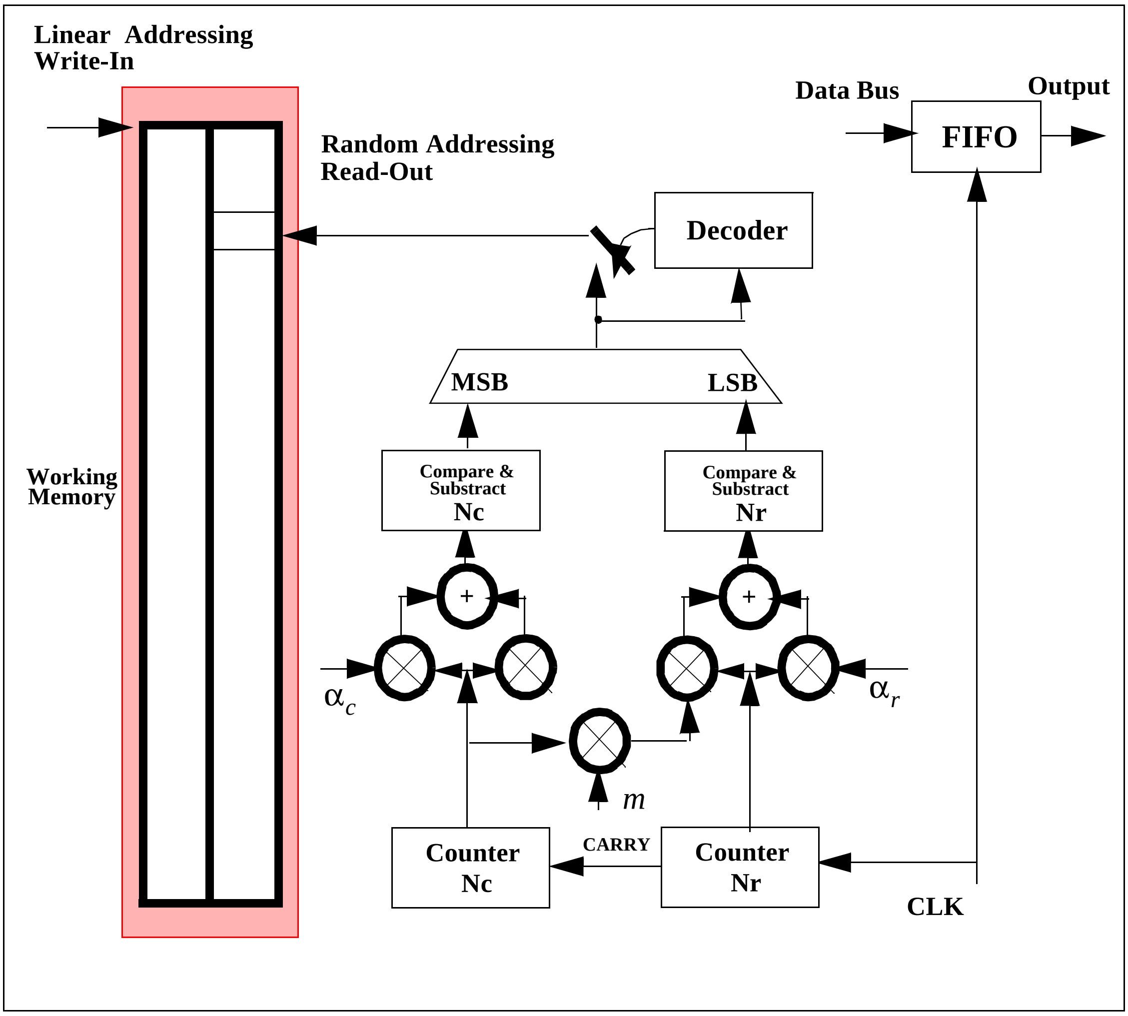
However, by introducing the 2nd interleaver after multiplexing of data, the interleaving performance might be further improved, this is currently under investigation.

3.0 Advantages of proposed channel interleaver

3.1 Low complexity of De-Interleaving and Implementation of Channel Interleaver

One of the most interesting feature for the proposed Algebraic channel interleaver is that *the deinterleaver is the same as the interleaver*. In addition, the algebraic interleaving address can be generated on-the-fly. Such an interleaver structure allows a very simple hardware implementation, see Figure 1.

FIGURE 1. Implementation of Algebraic Interleaver



3.2 Flexibility to Adapt to Arbitrary Interelaving Size

If we chose the interframe interleaving range from 1 to 8 frames, i.e. $N_c = 1, \ldots 8$, then by Euclid's Division Lemma, for any interleaving size N, we can find the matrix row number N_r, such that $N = N_r N_c + r$, where $r = \{1, \ldots 8\}$, these additonal address should be deleted to match arbitrary interleaving frame size. It is straightforward to see that maximum number of such index deletions is 8 addresses. This index deletion for frame size matching can be implemented by an address decoder in conjuction with a pre-filled FIFO with a depth of 8 symbols. See Figure 1.

3.3 Robustness to puncturing

One of the issues of the 3GPP uplink channel interleaver is how to avoid consecutive puncture of adjacent symbols of the coded blocks from the channel encoder output. This is important to ensure the performance of the channel decoder, especially in the case when Turbo coding is employed. The inter-frame interleaving scheme using MIL interleaver suffers from such a problem. Although the so-called Potential Puncturing Grid (PPG) concept is introduced to remedy such a problem. However, the choice of appropriate PPG depends on the frame size, number of frames, puncturing rate and the puncturing position. This leads to a proposal (Tdoc in Ad Hoc 4) to modify the rate matching rules.

To reduce the decoder performance degradation, the distance adjacent punctured symbol should be separated as far as possible. *It can be shown that the Algebraic interleaver can guarantee that the distance of adjacent punctured symbol is maximized for arbitrary puncturing rate and position.* Such an interleaver is PPG free. For example, In Table-3, if the horizontal shaded row is punctured the distance of punctured symbol is 9. In fact, to puncture any row, the distance of punctured symbol is always 9. However due the wrap up effect of the PPG, the minimum distance of the punctured symbol is always less than that of the Algebraic interleaver.

The minimum puncturing symbol distance for AL, MIL and BL interleaver is listed in Table 4.

TABLE 4. Comparison of Puncturing Distance (without PPG)

	Minimum Distance of Punctured Symbols			Average Distance of Punctured Symbols		
Frame Size	**AL**	**MIL**	**BL**	**AL**	**MIL**	**BL**
64x8	57	1	1	58	1	1
160x8	153	1	1	154	1	1
480x8	473	1	1	474	1	1
1120x8	1113	1	1	1114	1	1

As we can see the obvious advantage of Algebraic interleaver over MIL and BL interleavers in terms of distance of punctured symbols. On other hand, AL interleaver allow totally flexibility of Rate Matching puncturing and it is PPG free.

3.4 Robustness to fading

The main objective of channel interleaver is to allow the consecutive faded symbol at decoder input to be spread as far as possible by de-interleaving. Hence, if assume $I(n)$ in the interleaver output, then a good channel interleaver should satisfy the following condition:

$$max\{|I(n)-I(n-1)|, |I(n)-I(n-2)|, \ldots, |I(n)-I(n-N_c)|\}$$

One of the major advantages for the algebraic interleaver is that it allows to optimize the parameters such α_r and α_c with respect to the above criterion.

Our goal is to fully optimize the channel interleaver during the inter-frame interleaving. As a comparison example, in what follows, we assume only one QoS TrCH with 80ms frame, we compare optimized Algebraic interleaver as 1st stage interleaver, with 1st stage and 2nd stage MIL interleaver, the comparison is based on the interleaving spreading distance and the puncturing distance.

Example-2: A 80ms frame with a frame size of 64*8. The parameters for Algebraic interleaver are $\alpha_c = 3$, $\alpha_r = 31$, $m = 8$, the parameter for MIL interleaver is defined as in Tdoc029/99 $64[4[2\times 2]\times 16[4[2\times 2]\times 4[2\times 2]]]$, in addition we compare also the block interleaver. We defined the following measures:

$$\bar{D}(d) = E\left\{\frac{|I(n)-I(n-d)|}{N_c}\right\} \qquad D_{min}(d) = min\left\{\frac{|I(n)-I(n-d)|}{N_c}\right\}$$

FIGURE 2. Comparison of Mean Spreading

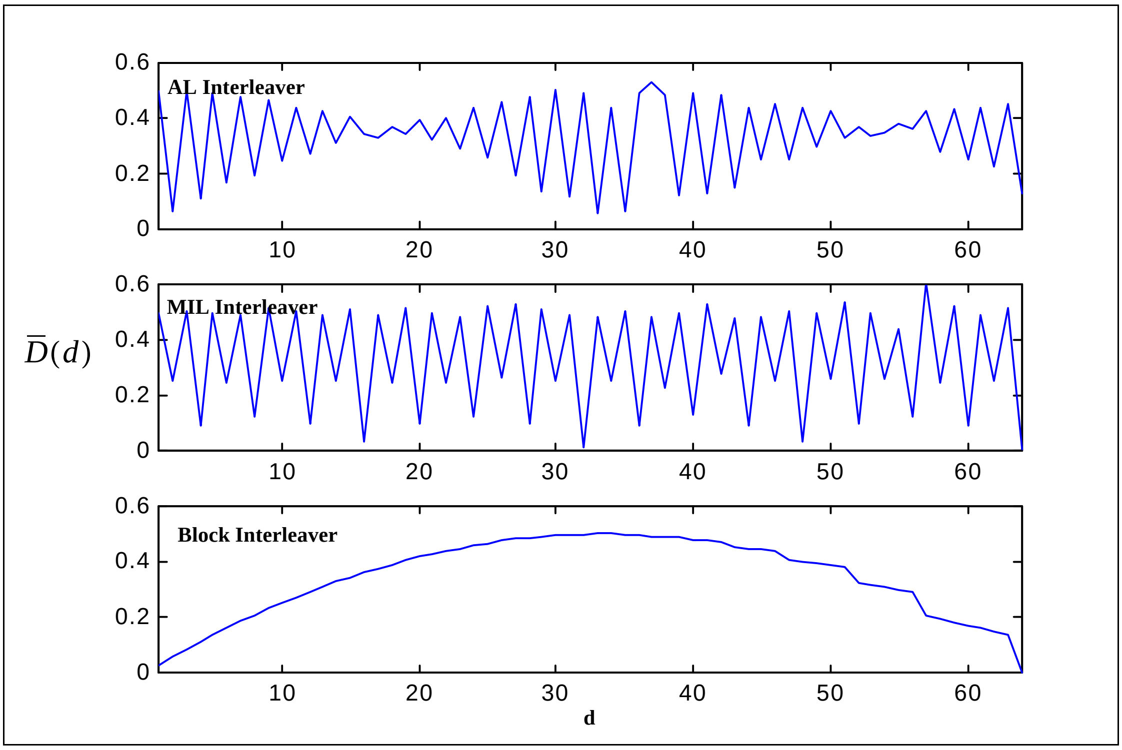


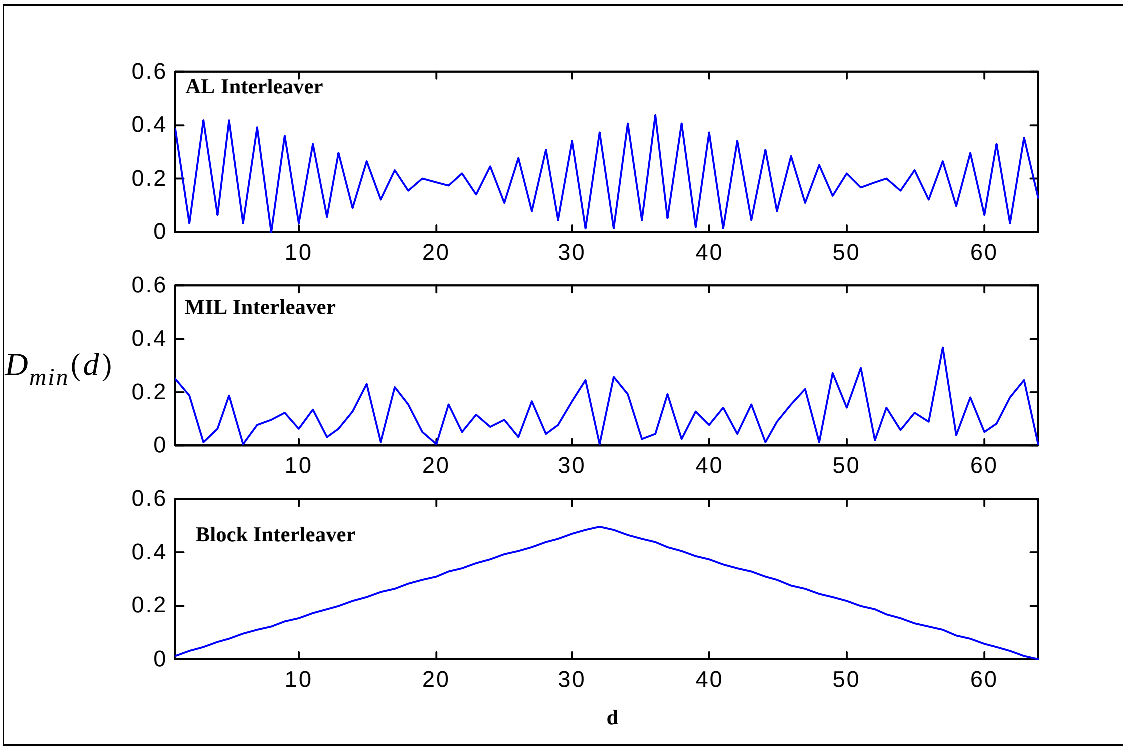
FIGURE 3. Comparison of Minimum Spreading

From Figure 2, we can see both AL and MIL interleaver are optimized in terms of interleaving spreading. Algebraic interleaver possesses a larger spreading than the MIL interleaver. Note that the ideal interleaver should achieve $\bar{D}(d) = 0,5 \quad \forall d$. From Figure 3, we can see the AL interleaver possesses a larger spreading than the MIL interleaver in terms of minimum spreading distance. Note also the ideal interleaver should achieve $D_{min}(d) = \bar{D}(d) = 0,5$.

3.5 Future proofness

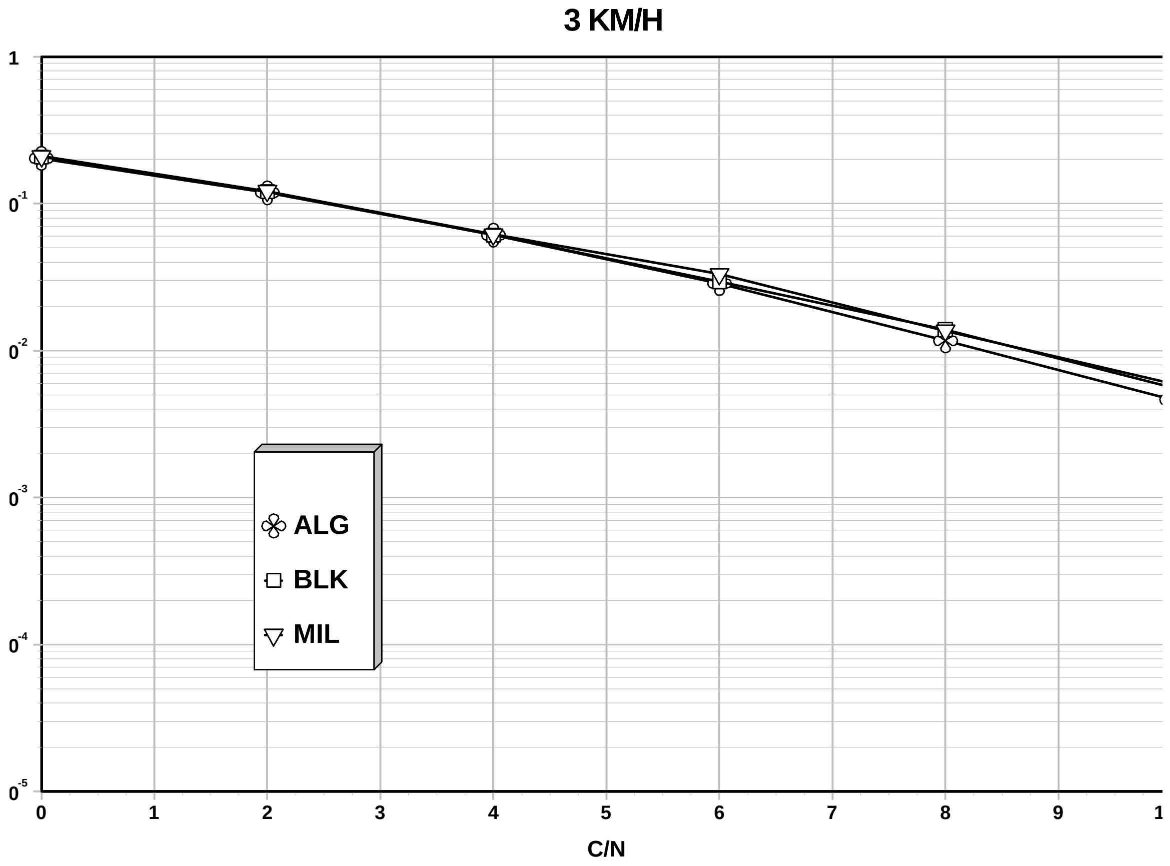
Thanks to the use of a formula to define the interleaver indexes, this interleaver will naturally cope with all frames sizes defined in the future without any need for defining new index tables or patterns specific to frame sizes.

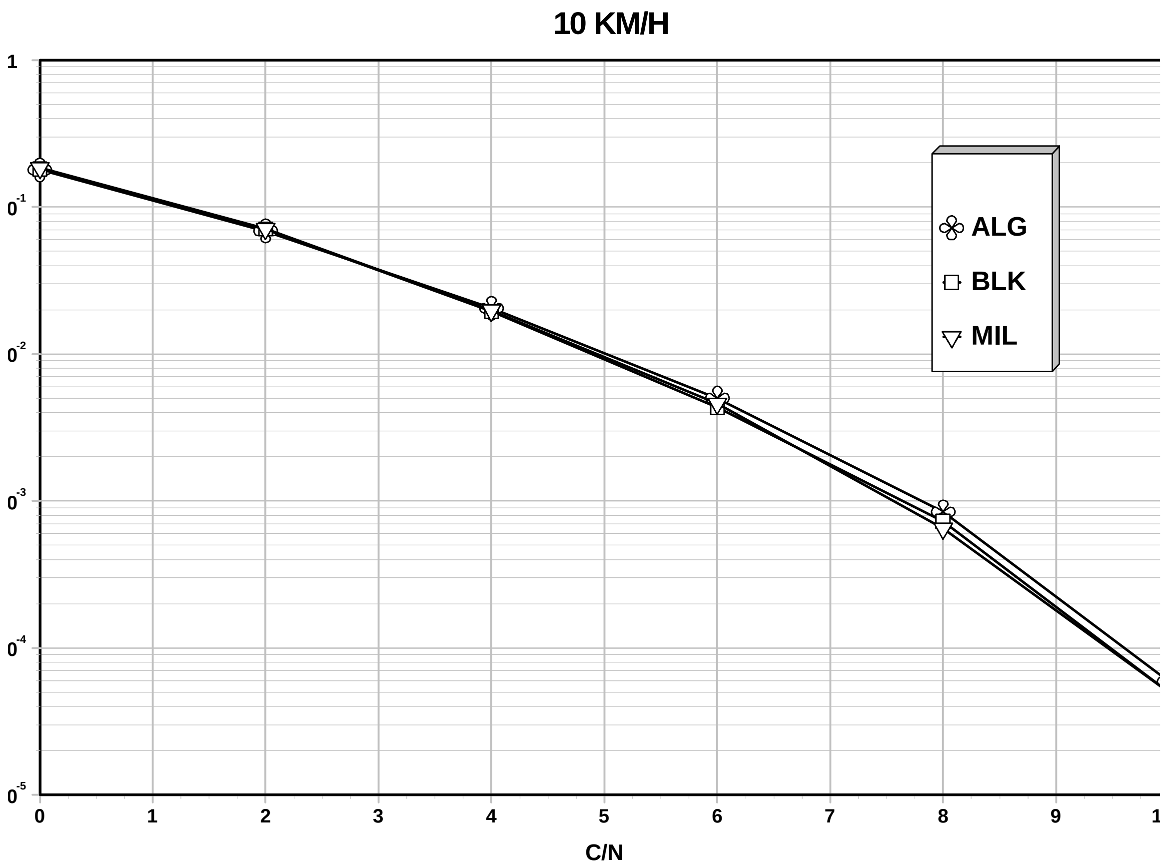
In the same way, any rate of puncturing can be applied withour the need to re-define some specific puncturing patterns.



In this respect, it is claimed that this interleaver will adapt smoothly to any evolution of the whole chain of Transport Channel Coding, Rate Matching and Multiplexing.

4.0 Performance




10 KM/H
1
0-1
0-2
0-3
0-4
0-5
0
1
2
3
4
5
6
7
8
9
1
C/N
ALG
BLK
MIL

5.0 Conclusion

In this document, we proposed a channel interleaver scheme which has a very low complexity, a great flexibility towards frame sizes, which is robut towards fading and puncturing, and does not need to have specific puncturing patterns defined. It is also remarkably future proof due to the use of a formula instead of index tables storage and the adaptability to all puncturing rates.

Therefore we recommend that this channel interleaver is adopted for UTRA in 3GPP.