Title: **Performance Evaluation of Codeblock Based Flexible-Size Channel Interleaving**
Source: **Alcatel-Lucent**
Agenda Item: **5.5 Channel coding**
Document for: **Discussion**

1. Introduction

At the last RAN WG1 meetings in Malta (#48bis) and Kobe (#49), a channel coding architecture with codeblock based rate matching [1] based on circular buffers [2] has been agreed. Currently, there are three different proposals in the discussion on the RAN1 e-mail reflector for channel interleaving: Codeblock based channel interleaving [3], interleaving within one [4] or *n* [5] OFDM symbols or no channel interleaving.

In this contribution we discuss the performance of codeblock based channel interleaving in the framework of codeblock based circular buffer rate matching. We present simulation results for the performance evaluation of codeblock based channel interleaving for downlink transmission. The basic functionality of our codeblock based channel interleaver is described in Appendix B or in [6]. Interleaver segmentation is performed prior to the actual channel interleaving in such a way that interleaver segments either span the entire codeblock or just parts of it.

In our simulations we cannot find a benefit for additional channel interleaving in combination with the interleaving already included in the circular buffer rate matching. Our results also give a very strong indication that OFDM symbol based channel interleavers will not improve the inherent interleaving from circular buffer rate matching.

Based on these observations we come to the conclusion that in the channel coding architecture with circular buffer rate matching there is no need for further explicit channel interleaving in downlink transmission. This significantly simplifies the channel encoding chain.

2. Architecture of codeblock based flexible-size channel interleaver

Figure 1 shows the architecture of the codeblock based channel interleaving studied in this paper. The channel interleaver blocks work as follows:

- First a segmentation into interleaver segments (also called *runs*) is performed, and
- then interleaving within these segments is carried out (potentially in parallel).

The number of interleaver segments depends on the number of bits within a codeblock after rate matching, and on the number of columns M of the fundamental block interleavers. Interleaver segmentation is performed in such a way that the size of the block interleavers does not exceed $4/3 \cdot M$. Setting M to a large value with respect to the codeblock size after rate matching leads to a channel interleaver spanning across the entire codeblock, whereas setting M to a small value leads to channel interleaving within only parts of a codeblock.

This means that the actual number of columns M together with the used physical resources define how much time-diversity can be captured by the channel interleaving. Especially we can also obtain settings where we effectively interleave within one or several OFDM symbols only. We will have a closer look at this in Section 3, where the resource allocation of the codeblocks and the interleaver segment will be shown..

Note that the interleaver segmentation is not shown in Figure 1 but rather discussed in more detail, including other important interleaver features, in Appendix B or in [6].

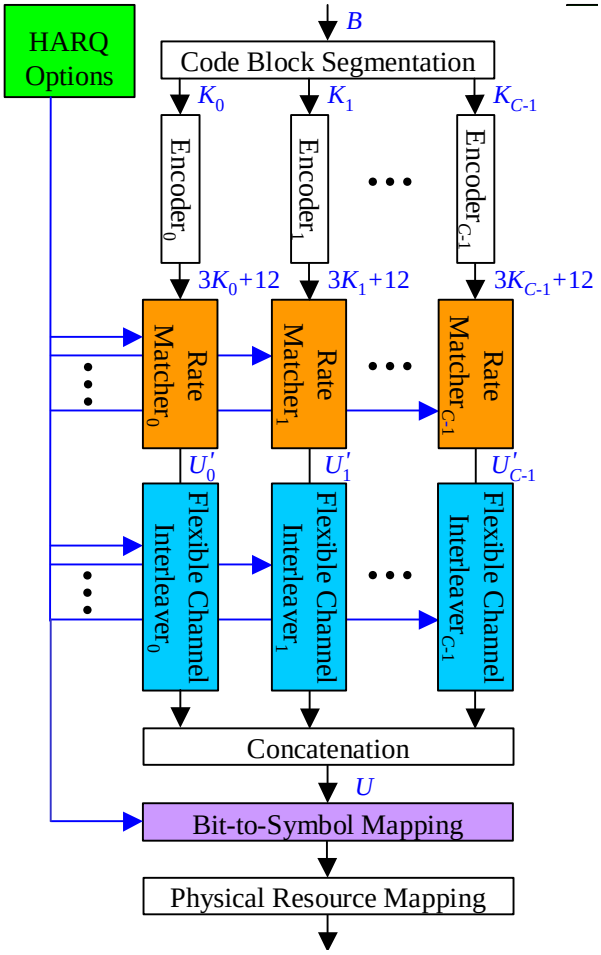

Figure 1: Channel coding architecture with codeblock based rate matching and channel interleaving

As stated above, channel interleaving is only performed on the bits within one codeblock. Thus, at the receiver, the de-interleaving, de-rate matching and turbo decoding of the first segment can start as soon as the first U'_0 bits are received. The codeblock after rate matching may have different sizes U'_i, $i = 0,\ldots,C-1$, because the codeblock segmentation may lead to segments with different sizes K_i, $i = 0,\ldots,C-1$. Therefore, channel interleavers with two different parameter sets according to Appendix B may be necessary.

3. Simulation results

3.1. Overview

We evaluate the performance of the codeblock based channel interleaving for downlink transmission. Our simulations are performed for a bandwidth of 20 MHz. The number of information bits is 1500 bytes equal to 12000 bits. Codeblock segmentation leads to two equal size codeblocks. QPSK modulation is used. The number of used resource blocks is chosen in such a way that the two target code rates 1/2 and 3/4 are obtained. For code rate 1/2 this means that 96 resource blocks are used, whereas 64 resource blocks are occupied for code rate 3/4. Note that both control overhead in the first 3 OFDM symbols within a subframe and reference symbol overhead for 1 Tx antenna are included and therefore 126 QAM symbols are available per resource block.

A typical urban channel is used with a UE velocity of 120 km/h.

The channel interleaver uses different numbers of interleaver segments and different numbers of columns for its basic block interleavers. The mapping of codeblocks and interleaver segments to the actual physical resources will be shown in the following subsections. Note that the number of

interleaver columns *M* and interleaver segments is directly related by the equations given in Appendix B. The greater the number of columns M, the lower is the number of interleaver segments.

A table with the simulation parameters is included in Appendix A. A list of the simulations is given in Table 1.

Table 1: Overview of simulations

Modulation	RBs	QAM symbols	Inf. bits	Codeblocks (CB)	Bits per CB after RM	RV	Interleaver colums	Interleaver segments per CB	Bits per interleaver segment
QPSK	96	12096	12000	2	12096	7	58	5	2418
									2418
									2420
									2420
									2420
QPSK	96	12096	12000	2	12096	7	86	2	6048
									6048
QPSK	96	12096	12000	2	12096	7	106	1	12096
QPSK	96	12096	12000	2	12096	7	No interleaving		---
QPSK	64	8064	12000	2	8064	0	46	5	1612
									1612
									1612
									1614
									1614
QPSK	64	8064	12000	2	8064	0	74	2	4032
									4032
QPSK	64	8064	12000	2	8064	0	86	1	8064
QPSK	64	8064	12000	2	8064	0	No interleaving		---

3.2. Resource allocation for QPSK 1/2 using 96 resource blocks

Figure 2 illustrates the resource allocation for the codeblocks for QPSK with a target code rate of 1/2. The first 3 OFDM symbols are reserved for control information and are not used for data. Pilot overhead is included but not explicitely marked in order not to overload the figure. The green color marks the control overhead and the 4 unused resource blocks. The red color stands for the first codeblock and the blue color for the second codeblock. Here, the number of interleaver columns is set to 106 and there is only one interleaver segment per codeblock. The codeblocks span across 5.5 OFDM symbols (not taking into account unused resource blocks).

Next, Figure 3 shows the resource allocation for two interleaver segments per codeblock and 86 interleaver columns. Here, each segment spans across 2.75 OFDM symbols.

Then, Figure 4 gives the resource allocation for 5 interleaver segments per codeblock with 58 interleaver columns. A segment spans across 1.1 OFDM symbols.

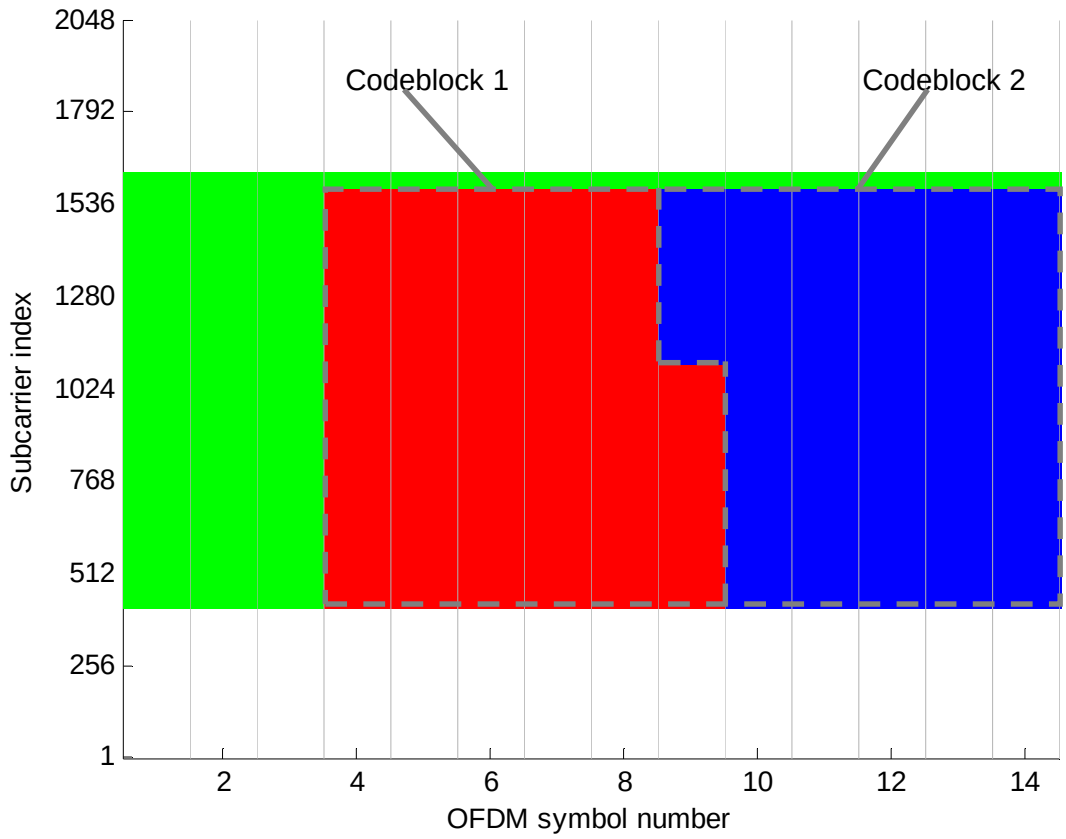

Figure 2: Resource allocation for QPSK 1/2 using 96 resource blocks with one interleaver segment per codeblock and 106 interleaver columns.

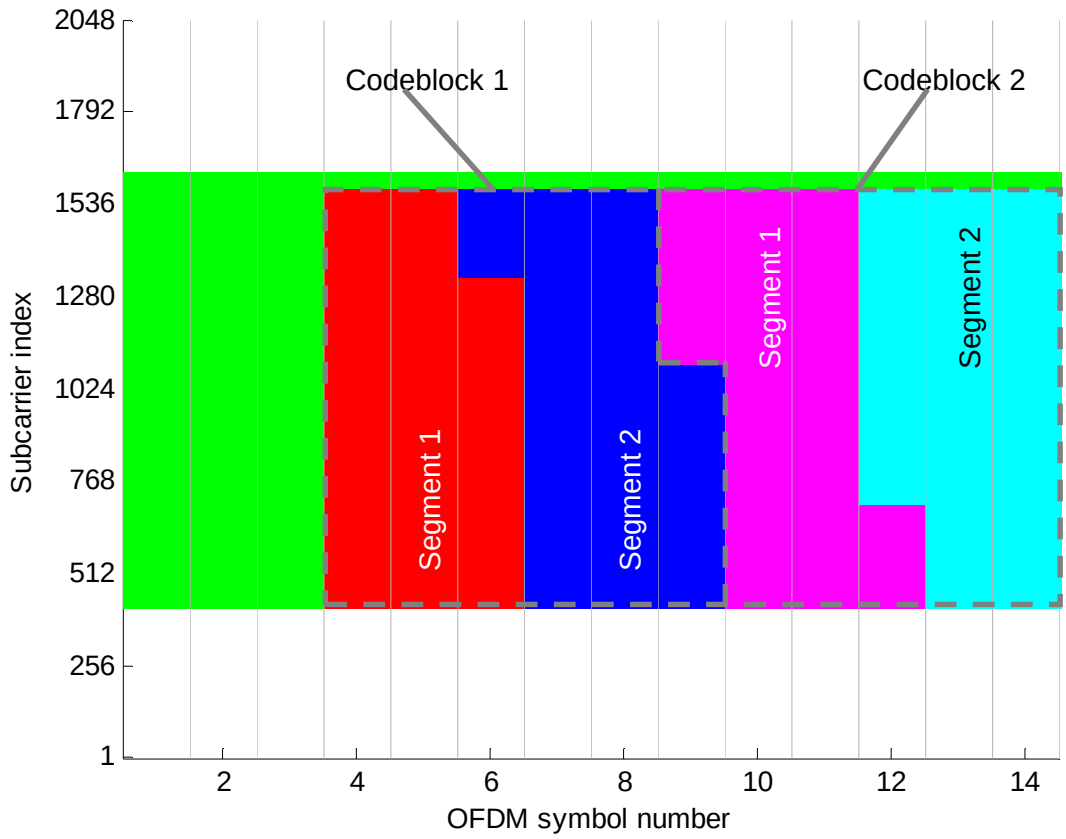

Figure 3: Resource allocation for QPSK 1/2 using 96 resource blocks with two interleaver segments per codeblock and 86 interleaver columns.

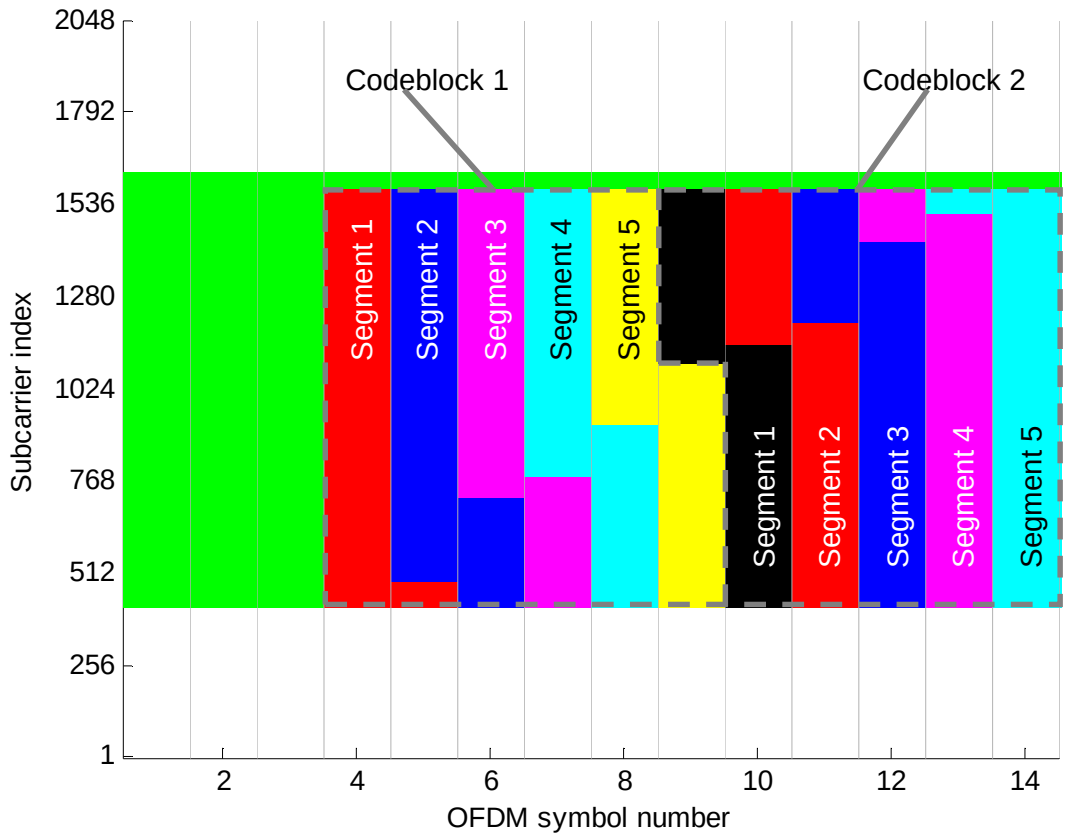

Figure 4: Resource allocation for QPSK 1/2 using 96 resource blocks with 5 interleaver segments per codeblock and 58 interleaver columns.

3.3. Resource allocation for QPSK 3/4 using 64 resource blocks

Figure 5 illustrates the resource allocation for the codeblocks for QPSK with a target code rate of 3/4. No interleaver segmentation is performed and the number of interleaver columns is 86. The codeblocks span across 5.5 OFDM symbols.

Next, Figure 6 shows the resource allocation for two interleaver segments per codeblock and 74 interleaver columns. Here, each segment spans across 2.75 OFDM symbols.

Then, Figure 7 gives the resource allocation for 5 interleaver segments per codeblock with 46 interleaver columns. A segment spans across 1.1 OFDM symbols.

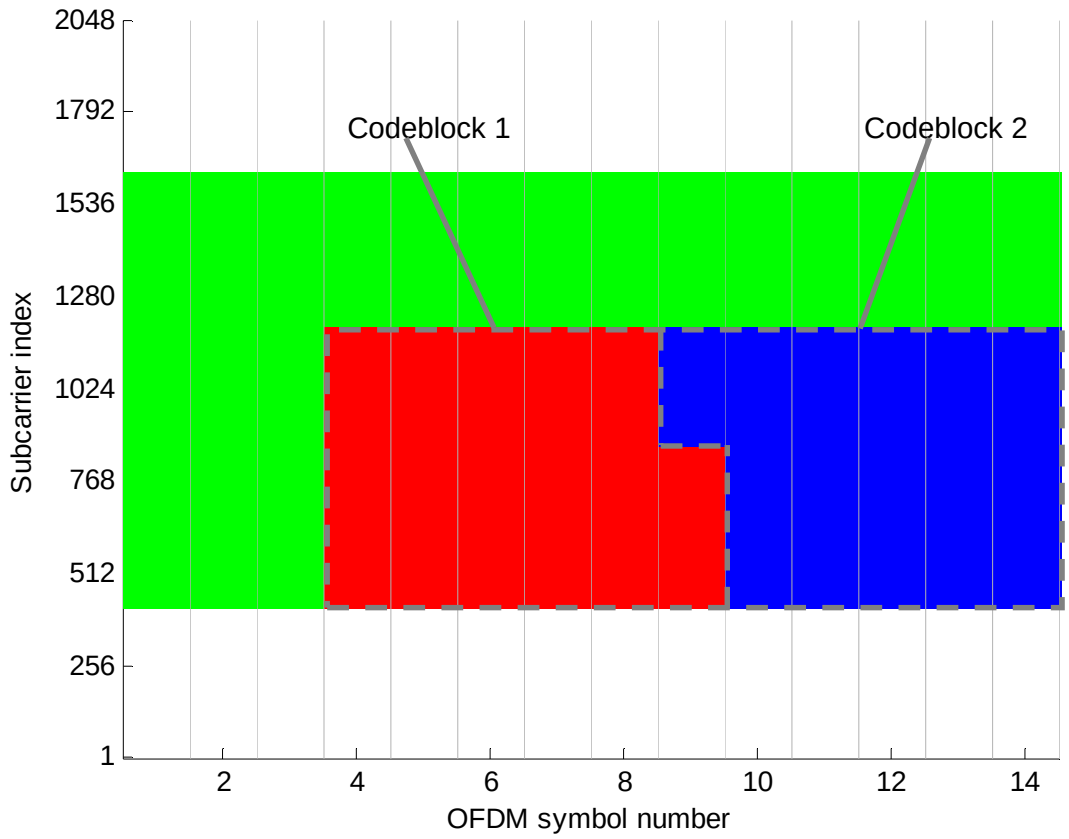

Figure 5: Resource allocation for QPSK 3/4 using 64 resource blocks with one interleaver segment per codeblock and 86 interleaver columns.

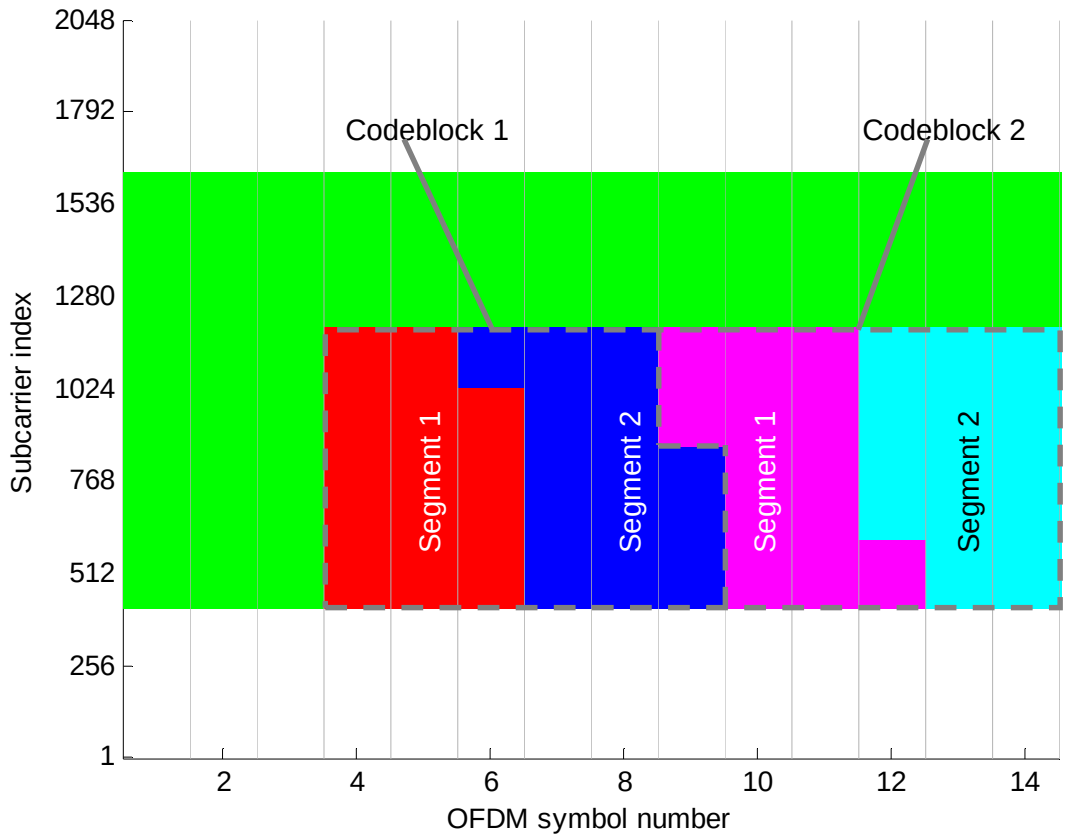

Figure 6: Resource allocation for QPSK 3/4 using 64 resource blocks with two interleaver segments per codeblock and 74 interleaver columns.

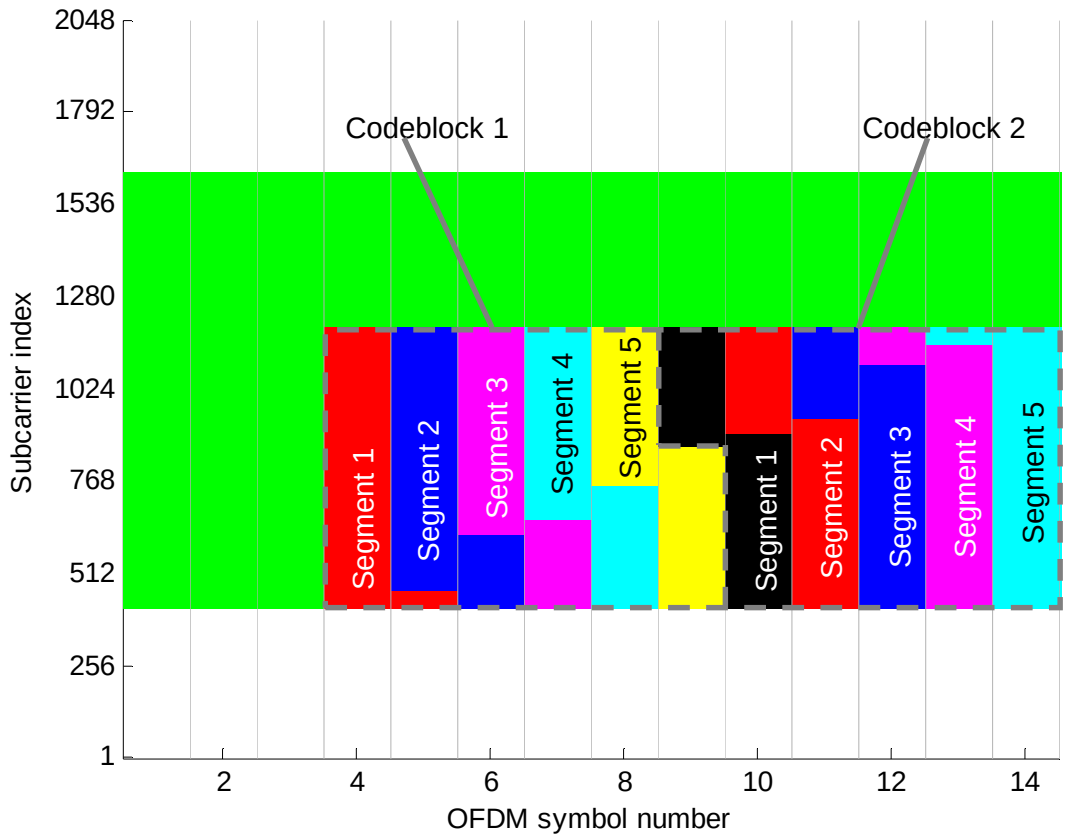

Figure 7: Resource allocation for QPSK 3/4 using 64 resource blocks with five interleaver segments per codeblock and 46 interleaver columns.

3.4. Simulation results

Figure 8 depicts our simulation results in the form of BLER vs. SNR. We observe that both for code rate 1/2 and for code rate 3/4 all curves lie almost directly upon each other. Channel interleaving in addition to the circular buffer rate matching inherent interleaving does not give any additional benefit, no matter of across how many OFDM symbols the individual interleaver segments are spread. This also indicates that OFDM symbol based channel interleaving will very likely not give any benefits, as the time diversity captured there is even less.

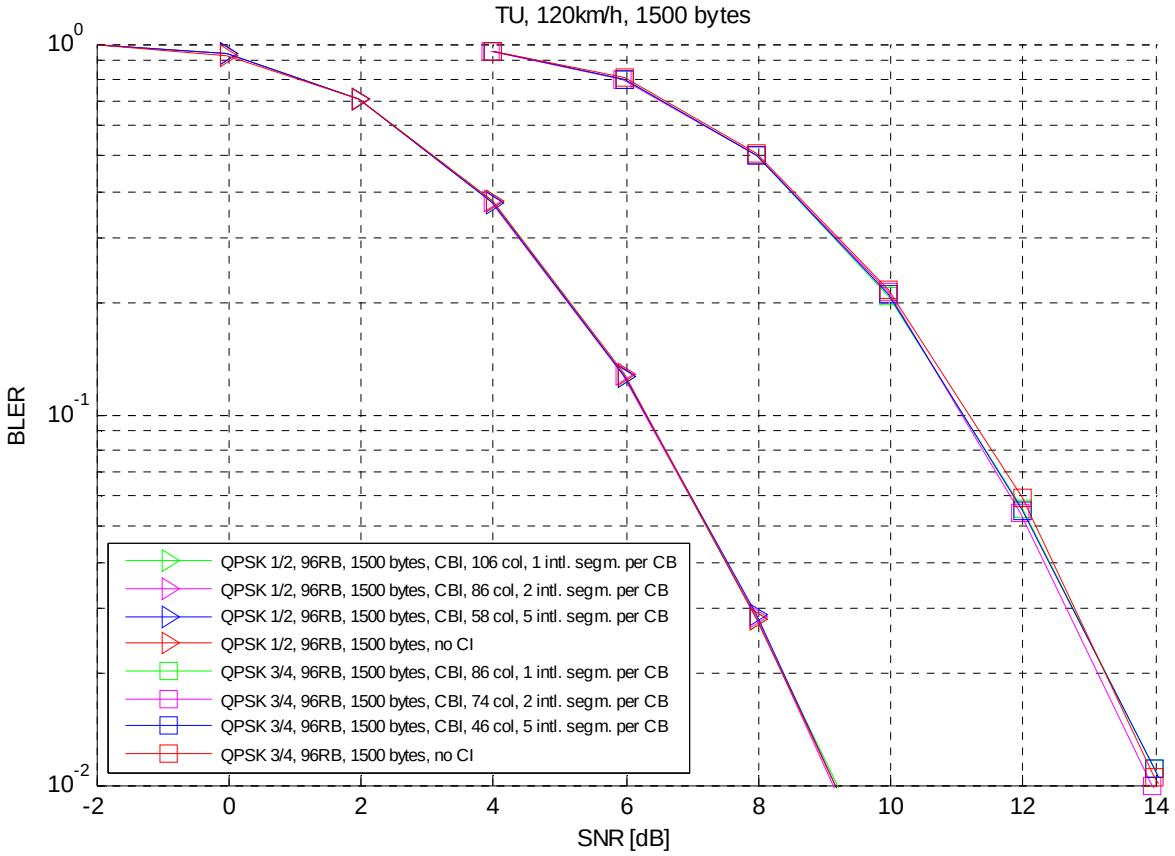

Figure 8: BLER vs. SNR for different number of interleaver segments per codeblock with codeblock based channel interleaving (CBI) and no channel interleaving (no CI)

4. Conclusion

We have presented simulation results for codeblock based channel interleaving. Interleaver segmentation was done in such a way that the actual interleaving was performed either across complete codeblocks or smaller segments thereof. The greater the number of interleaver segments per codeblock, the smaller is the amount of time-diversity that can be captured by channel interleaving.

Our results show that independent of the number of interleaver segments per codeblock, there is no benefit of additional channel interleaving in combination with the interleaving already included in the circular buffer rate matching.

Our results also give a very strong indication that OFDM symbol based channel interleavers will not improve the inherent interleaving from circular buffer rate matching.

Based on our observations we come to the conclusion that in the channel coding architecture with circular buffer rate matching there is no need for further explicit channel interleaving.

References

[1] R1-072001, Draft Report of 3GPP TSG RAN WG1 #48b v0.3.0, St. Julians, Malta, March, 2007.

[2] R1-072604, Ericsson, ETRI, Freescale, ITRI, LGE, Motorola, Qualcomm, Samsung, ZTE, “Way forward on HARQ rate matching for LTE”, 3GPP TSG RAN WG1 #49, Kobe, Japan, May 2007.

[3] R1-072046, Alcatel-Lucent, “Flexible-size segmentation for E-UTRA channel interleaving”, 3GPP TSG RAN WG1 #49, Kobe, Japan, May 2007.

[4] R1-071588, Ericsson, “Design Considerations for Hybrid ARQ Processing Chains”, 3GPP TSG RAN WG1 #48bis, St. Julians, Malta, March 2007.
[5] R1-072246, Samsung, “Channel coding and interleaving to enable pipelining in LTE”, 3GPP TSG RAN WG1 #49, Kobe, Japan, May 2007.
[6] R1-071426, Alcatel-Lucent, “Flexible channel interleaver for E-UTRA”, 3GPP TSG RAN WG1 #48bis, St. Julians, Malta, March 2007.
[7] 3GPP TS 25.212 V6.10.0, Multiplexing and channel coding (FDD) (Release 6), December 2006.
[8] 3GPP TS 36.212 V1.0.0, Multiplexing and channel coding (Release 8), March 2007.

Appendix A: Simulation parameters

Table 2: General simulation parameters

Parameter	Setting
Transmission bandwidth	DL 20 MHz
FFT size	2048
TTI length	1.0 ms (1 subframe)
Control overhead	3 OFDM symbols per subframe
Number of data QAM symbols per RB	126
Modulation scheme	QPSK, code rate approx. 1/2 or 3/4
Number of used RBs	96 for QPSK 1/2; 64 for QPSK 3/4
Turbo encoding	According to [8] incl. QPP turbo code internal interleaver
Rate matching	Circular buffer rate matching according to [2]
Turbo decoding	MaxLogMAP algorithm, max. 8 iterations
Transmission scheme	SISO
Channel model	Typical urban, 120 km/h
Channel estimation	Ideal channel estimation

Appendix B: Flexible channel interleaver details

B.1 Key features

We want to recall the key features of our proposed channel interleaver. These are:

- Full flexibility with respect to the number of input bits.
- Two-dimensional OFDM time-frequency resource mapping taken into account.
- Adjacent bits from the interleaver input are not transmitted over the same subcarrier.
- Memory requirement independent of number of input bits but dependent only on maximum segment size.

- Bits from different interleaver runs (i.e. interleaver segments) can be processed in parallel.
- All interleaver runs process approx. the same number of bits ("first runs" and "last runs" differ by two bits) using the same operations. Therefore, all bits experience the same level of interleaving protection.

The working principle of the proposed channel interleaver is as follows:

In analogy to Release 6 channel coding [7], there is one basic interleaver for QPSK or two identical basic interleavers for 16-QAM. For 64-QAM we extend this principle so that we have three identical basic channel interleavers as shown in Figure 9 below. As the operations of the two or three basic interleavers for 16-QAM or 64-QAM, respectively, are identical, we will focus only on one basic interleaver in the following.

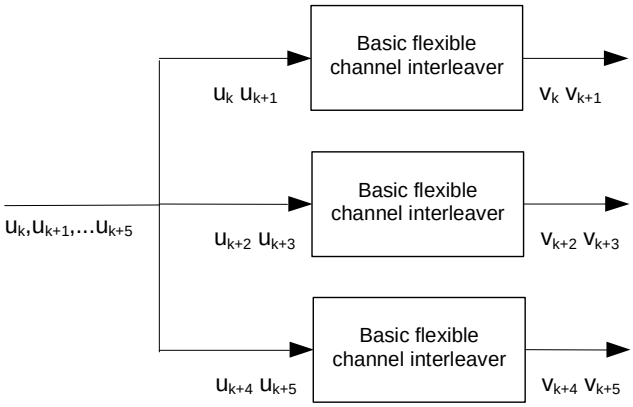

Figure 9: General channel interleaver structure for 64 QAM data channel

In our flexible channel interleaver approach a flexible-size segmentation is performed. The segmentation is done with the following parameter set depending on the number of input bits:

- Number of first runs and number of last runs
- Number of rows N and number of columns M respectively.

In this context, a *run* denotes the block interleaving of a subset (i.e. segment) of the input bits into the basic interleaver.

The interleaving of the input bits to one basic channel interleaver comprises the following steps:

- Selection/Calculation of interleaver parameters according to number of input bits
- Variable segmentation of input bits for first und last runs
- Block interleaving in first and last runs (may be processed in parallel)
- Concatenation of interleaved bits from first and last runs

See Figure 10 for an illustrative example of the concept of first and last runs (interleaver segments). There, 412 bits are interleaved in 4 first runs and 2 last runs. Note, that Figure 10 is for illustrative purposes only and does not use the proposed number of columns and rows.

In contrast to our proposed channel interleaver, the Release 6 channel interleaver for HS-DSCH (HSDPA channel interleaver) performs a fixed-size segmentation (physical channel segmentation). Each segment comprises 960 bits and interleaving on this segment is performed using a 32x20 block interleaver. This works only with a very coarse granularity in the number of bits to be interleaved.

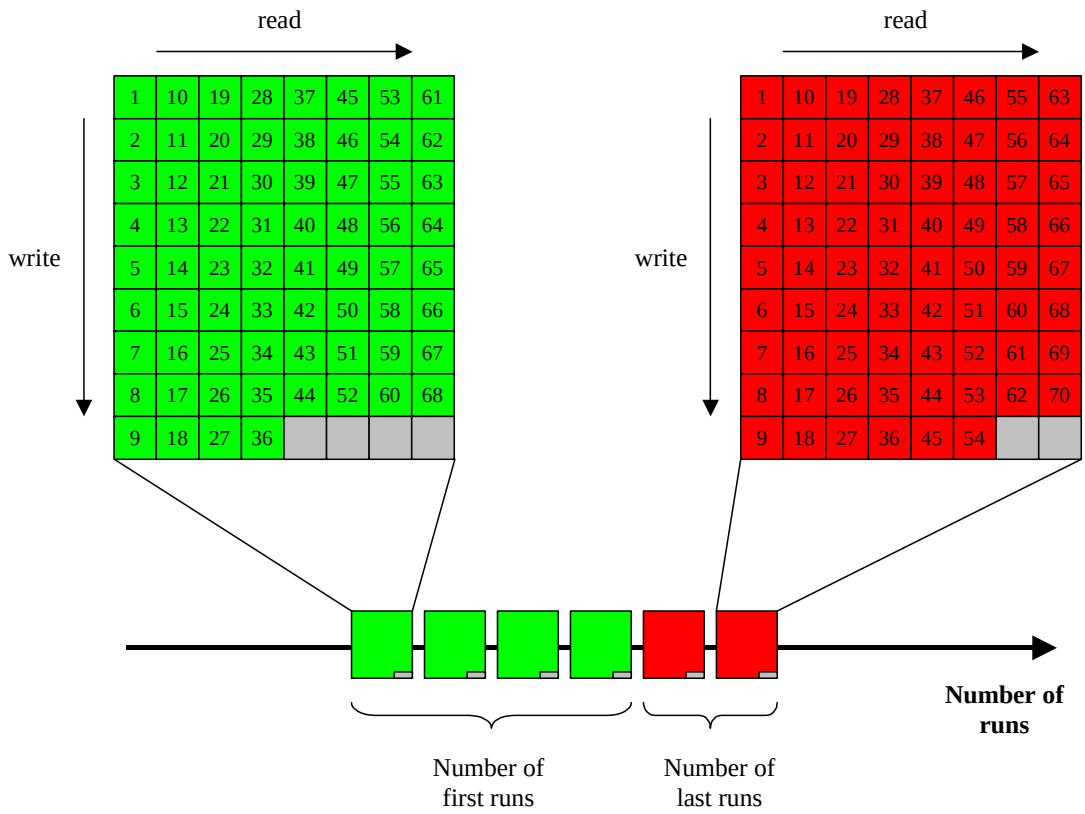

Figure 10: Flexible interleaver example with NxM = 9x8, first runs = 4, last runs = 2

B.2 Parameter assignment for segmentation

For the sake of completeness and convenience, we include the parameters and calculations from [6].

The number of rows N and number of columns M, number of matrix elements, and number of first and last runs for the proposed channel interleaver are derived within the following steps:

1. Number of columns: $M \in \{14,34,46,58,74,86\}$. (Note: The actual value of M must be matched to the number of used resource blocks. In this contribution, however, M was used as a free parameter to influence the number of interleaver segments or runs.)
2. Minimum Number of Rows: $N_{\min} = \min\left(\left\lfloor \frac{N_{data}}{(m+1)\cdot M} \right\rfloor, \left\lfloor \frac{2M}{3} + \frac{1}{2} \right\rfloor\right)$
3. Number of interleaver runs assuming $N_{\min}$ rows: $k = \left\lfloor \frac{N_{data}}{(m+1)\cdot M \cdot N_{\min}} \right\rfloor$
4. Number of bits r, which remain after multiple $N_{\min}$ runs: $r = \frac{N_{data}}{m+1} - k \cdot M \cdot N_{\min}$
5. Number of rows: $N = \left\lceil \frac{r}{k \cdot M} \right\rceil + N_{\min}$
6. Total number of filling bits: $N_{fill_total} = N \cdot M \cdot k \cdot (m+1) - N_{data}$
7. Number of filling bits per interleaver during last runs: $N_{fill_\min} = 2 \cdot \left\lfloor \frac{N_{fill_total}}{2\cdot(m+1)\cdot k} \right\rfloor$
8. Number of first runs u with $N_{fill_\min}+2$: $u = \frac{1}{2} \cdot \left[\frac{N_{fill_total}}{m+1} - k \cdot N_{fill_\min} \right]$
9. Number of last runs v with $N_{fill_\min}$: $v = k - u$
10. Number of matrix elements: $N_{elements} = N \cdot M - (N_{fill_\min})$

The modulation parameter m can take values from 0 for QPSK, over 1 for 16-QAM to 2 for 64-QAM. N_{data} is the number of bits to be interleaved.

The maximum number of rows N in step 5 is $2 \cdot N_{min}$. This defines the memory requirement for the interleaver implementation, if the individual interleaver runs are performed successively, or the maximum processing delay, if the individual interleaver runs are performed in parallel.

B.3 Permutation

A column permutation should be applied to break correlations in the two-dimensional OFDM time-frequency map. We propose that the columns are permuted according to the permutation patterns $P=\{p_1,p_2,\ldots,p_M\}$:

P={0, 3, 6, 9, 12, 1, 4, 7, 10, 13, 2, 5, 8, 11} for M=14,

P={0, 5, 10, 15, 20, 25, 30, 1, 6, 11, 16, 21, 26, 31, 2, 7, 12, 17, 22, 27, 32, 3, 8, 13, 18, 23, 28, 33, 4, 9, 14, 19, 24, 29} for M=34,

P={0,6,12,18,24,30,36,42,1,7,13,19,25,31,37,43,2,8,14,20,26,32,38,44,3,9,15,21,27,33,39,45,4,10, 16,22,28,34,40,5,11,17,23,29,35,41} for M=46,

P={0,7,14,21,28,35,42,49,56,5,12,19,26,33,40,47,54,3,10,17,24,31,38,45,52,1,8,15,22,29,36,43, 50,57,6,13,20,27,34,41,48,55,4,11,18,25,32,39,46,53,2,9,16,23,30,37,44,51} for M=58,

P={0,8,16,24,32,40,48,56,64,72,7,15,23,31,39,47,55,63,71,6,14,22,30,38,46,54,62,70,5,13,21,29, 37,45,53,61,69,4,12,20,28,36,44,52,60,68,3,11,19,27,35,43,51,59,67,2,10,18,26,34,42,50,58,66,1, 9,17,25,33,41,49,57,65,73} for M=74,

P={0,9,18,27,36,45,54,63,72,81,4,13,22,31,40,49,58,67,76,85,8,17,26,35,44,53,62,71,80,3,12,21, 30,39,48,57,66,75,84,7,16,25,34,43,52,61,70,79,2,11,20,29,38,47,56,65,74,83,6,15,24,33,42,51, 60,69,78,1,10,19,28,37,46,55,64,73,82,5,14,23,32,41,50,59,68,77} for M=86,

P={0,10,20,30,40,50,60,70,80,90,100,3,13,23,33,43,53,63,73,83,93,103,6,16,26,36,46,56,66,76, 86,96,9,19,29,39,49,59,69,79,89,99,2,12,22,32,42,52,62,72,82,92,102,5,15,25,35,45,55,65,75,85, 95,105,8,18,28,38,48,58,68,78,88,98,1,11,21,31,41,51,61,71,81,91,101,4,14,24,34,44,54,64,74,84, 94,104,7,17,27,37,47,57,67,77,87,97} for M=106.

These permutation patterns generate large distances in frequency direction of originally adjacent bits in the two dimensional OFDM time-frequency map.