



LUND UNIVERSITY

Coded IDMA system performance with parallel interleavers

Hammarberg, Peter; Wu, Shuang; Chen, Xiang; Wang, Jing; Edfors, Ove

Published in:
[Host publication title missing]

DOI:
[10.1109/WIAD.2010.5544488](https://doi.org/10.1109/WIAD.2010.5544488)

2010

[Link to publication](#)

Citation for published version (APA):
Hammarberg, P., Wu, S., Chen, X., Wang, J., & Edfors, O. (2010). Coded IDMA system performance with parallel interleavers. In *[Host publication title missing]* (pp. 1-4). IEEE - Institute of Electrical and Electronics Engineers Inc.. <https://doi.org/10.1109/WIAD.2010.5544488>

Total number of authors:
5

General rights

Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Coded IDMA System Performance with Parallel Interleavers

Peter Hammarberg[†], Shuang Wu^{*}, Xiang Chen^{*}, Jing Wang^{*} and Ove Edfors[†]

[†]Department of Electrical and Information Technology, Lund University, Lund, Sweden

^{*}Department of Electronic Engineering, Tsinghua University, Beijing, P.R.China

[†]{Peter.Hammarberg, Ove.Edfors}@eit.lth.se, ^{*}{pshuangcoo, chenxiang98}@gmail.com

Abstract—In this paper we evaluate the use of parallel interleavers for user separation in coded IDMA systems. The parallel interleavers have previously been evaluated for uncoded systems, and we here present extended results for the coded case. The new results are important for real systems, as they will most likely use performance enhancing channel coding. The parallel interleavers are designed to remove the risk of memory access collision during interleaving. These problems may occur when the interleaving process is being parallelized in order to support parallel processing at the decoder. The results show that the degree of parallelization in the interleaver design has an insignificant effect on the considered coded system performance. Furthermore, our results also show that the use of parallel interleavers gives a performance that is similar to that obtained when using random interleavers.

Keywords- IDMA; parallel interleavers; system performance

I. INTRODUCTION

A new multiple access technique was recently proposed, where the users are separated through their unique interleaving patterns. The technique, referred to as interleave-division multiple-access (IDMA), has been shown to mitigate multiple access interference (MAI) and inter-symbol interference (ISI), simultaneously, to achieve high spectral efficiency [1]-[3]. IDMA shares many properties with CDMA, where user separation is obtained through user-specific spreading codes, and has shown similar performance but with a reduced receiver complexity. The IDMA receiver performs iterative chip-by-chip multi-user detection (MUD), where inter user interference is cancelled using extrinsic information from a channel decoding unit [1].

Since user separations in IDMA can be obtained through user specific interleavers, the design of these are important, and has been covered in several publications. In [1]-[2], an IDMA system using randomly and independently generated interleavers was presented, demanding large memories for storage. More practical interleavers were proposed in [3]-[5], reducing memory requirements, at the cost of an increased computational complexity.

One limiting factor in the interleavers proposed in [1]-[5] is that they can only support sequential processing, without facing the risk of memory addressing collisions. This issue is resolved in [6], where a parallel interleaver design is proposed. The design allows the interleaving process to be parallelized by M processors without any risk of memory collision. The

proposed interleaver scheme allow for reduced processing latency, obtained through the parallelization, at the same time as the computational complexity and memory requirements of the interleaver can be kept low.

The choice of interleaver design will inevitably have an impact on the IDMA system performance, as observed in e.g. [4]. In [6], system performance for an uncoded system was presented, when using the parallel interleaver design. In this paper, we extend and verify these results by considering two different coded systems. Since any practical system implementation would use channel coding, in order to improve system performance, such evaluation is important. Furthermore, we also investigate if the degree of parallelization, in the interleaver design, has any significant affect on the system performance.

The rest of this paper is organized as follows. In Section II, the IDMA system model is first presented before giving a short description of the parallel interleaver design method. Our system simulation results are shown in Section III, and our conclusions are presented in Section IV

II. SYSTEM DESCRIPTION

A. The IDMA system model

In this paper a coded IDMA system is considered, composed of K users transmitting synchronously over an AWGN channel. A simplified view of the system model is shown in Figure 1.

For user k , the information sequence is first encoded using a forward error correction code (FEC), followed by spreading and user specific interleaving. For the FEC, both a convolutional code and a turbo code are considered. The user specific interleavers $\pi^{(k)}$ are either generated randomly or using the parallel design approach in [6], and will for the rest of the paper be referred to as random and parallel interleavers, respectively.

At the receiver, a per chip parallel interference canceling elementary signal estimator (ESE) is implemented, using the following per chip, real valued, baseband model for the received signal

$$r = h_k x_k + \sum_{i \neq k} h_i x_i + n, \quad (1)$$

where h_j , x_j and n is the channel coefficient, transmitted symbol and noise, respectively. Note that the signal

This work was supported by the Chinese research projects no. 2009AA011501 no. 2009ZX03003-009, no. 2007CB310608 and no. 2008DFA12160, along with the Swedish VINNOVA funded project P3358-1.

component for user k has been taken out from the summation to indicate that the received signal can be seen as composed of user signal, interference and noise. The noise is distributed according to $n \sim \mathbb{N}(0, \sigma_n^2)$. The ESE assumes that the interference plus noise can be regarded as coming from a Gaussian process, and produce log-likelihood outputs based on mean and variance estimates. The estimates are obtained using extrinsic information on the transmitted symbols acquired through the channel decoders (see [1] for more details). After the ESE, the separated user streams are de-interleaved, de-spread and decoded. The extrinsic information output of soft-input soft-output (SISO) decoders is then re-spread and re-interleaved before feed back to the ESE.

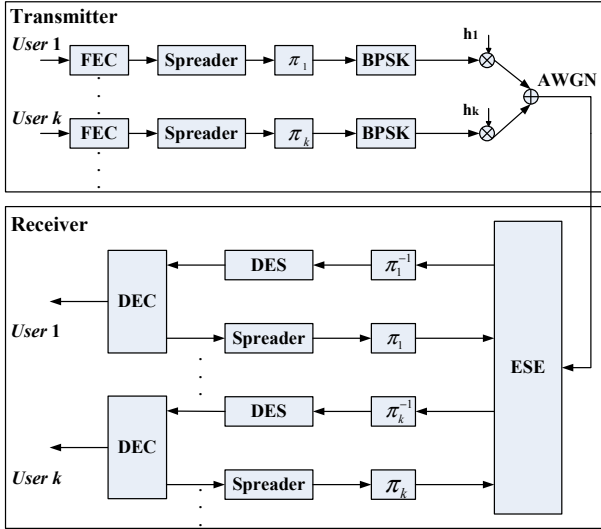


Figure 1. A schematic description of the considered IDMA system, consisting of k users, transmitting simultaneously to a receiver over an AWGN channel.

B. Parallel interleavers

Parallel processing of iterative multi-user detection is of great importance for high-speed detectors/decoders, demanding low latency. For an IDMA system with many users, and many receiver iterations, the decoding process may be a time consuming task, thus parallelization may be required to avoid performance limiting process delays.

One important aspect to be addressed in parallel processing in iterative receivers is the interleaving and de-interleaving of soft information. During the de-interleaving a memory access conflict may occur. This means that two or more processors try to access the data in the same memory unit simultaneously, which may have a severe impact on the receiver performance. For IDMA, this problem can be removed if using appropriately designed interleavers, e.g. using the parallel interleavers proposed in [6]. Below, the main design principles of the parallel interleavers will be presented.

For ease of later use, let us define the following parameters:

- N number of chips in a frame
- D number of information bits per frame per user

M degree of parallel processing

W number of chips to be processed in each processor

Here, $N = DS = MW$ and W are constrained. For example, with linear block codes, W is an integer multiple of the coding length, and with repetition codes, W is an integer multiple of the spreading length S .

The algorithm for the parallel interleaver design, can be divided into five stages, which are given as

- **Stage 1:** Assign the initial matrices with M rows and W columns.
- **Stage 2:** Put the coded bits into matrices by rows.
- **Stage 3:** Perform column-wise interleaving for each column.
- **Stage 4:** Perform row-wise interleaving for each row.
- **Stage 5:** Read out the bits by rows.

Detailed descriptions of **Stage 3** and **Stage 4** are given as follows

Stage 3: column permutation

- Different random seeds are allocated to different users.
- For user k , independent random sequences with length M are generated by its own random seed.

Stage 4: row permutation

- Let π_{mi} be the initial interleaver with length W .
- π_k , the interleaver of the first row of user k , is generated by cyclically shifting the initial interleaver by Lk steps, where $L = \text{int}(W/K)$, and $\text{int}(x)$ returns the greatest integer that is no larger than x .
- For user k , the interleaving patterns of the following row are generated by cyclically shifting the previous row's interleaver one step.

At the receiver, the received bits have to be de-interleaved, which is just a process of memory addressing. In Figure 2, a schematic view of the address generation and memory access procedure is shown for one of the users. The process is briefly explained below.

The address generation may be seen as composed of three steps, all performed for each of the W chips. First, the row indices $I_{R,m}$ are generated, then the corresponding column index $I_{C,w}$ is found, and finally the two indices are mapped to a physical memory address. The address generation is performed for all M rows in parallel. One column, of the available W , is handled per clock cycle.

When a system clock pulse arrives, the random number generator generates the w :th column interleaver of length M , using the same seed as in the transmitter. Using the obtained column index the m :th column address generator generate the corresponding column index. Once the row and column indices have been obtained, they are sent to the address generator

which provides the physical address $A_{m,w}$ of the chip in the memory. Finally, the data is written to the parallel decoders (DEC). Note that in this paper, parallel processing of the decoder is not implemented.

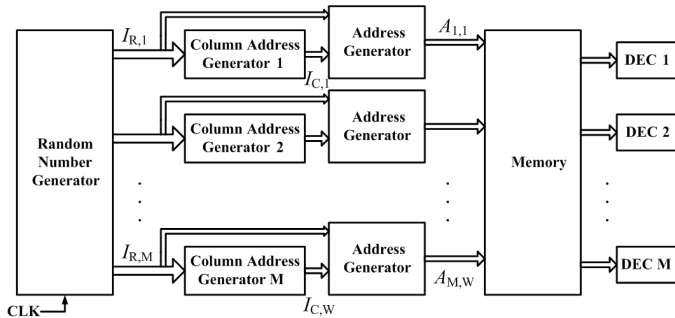


Figure 2. De-interleaving structure of the parallel interleavers for one user. The picture describe how the memory address is generated and how data is then written from the memory to the parallel decoders.

III. SIMULATION RESULTS

Monte Carlo simulations have been carried out in order to evaluate the performance of the parallel interleavers in a coded IDMA system, as well as for investigating any performance impact of the degree of parallelization of the parallel interleavers. A system overview of the IDMA system was given in section II, and a more detailed description follows.

Two different FEC's were considered for the coded IDMA system, one recursive systematic convolutional code and one turbo code. The convolutional code was implemented with generator polynomials $(7,5)_8$, thus generating output code bits at a rate of $1/2$. The second code considered was a parallel concatenated convolutional code (PCCC). The two parallel codes both use the same generator polynomials, $(13,15)_8$, and are both forced to the zero state using two tail bits. Together with the systematic bits, the two output streams create a rate $1/3$ turbo code. The inner interleaver of the turbo code was the same as the one specified for WCDMA [7]. For both of the considered decoders, MAP decoding was used to obtain the needed extrinsic information outputs [8]-[9].

Unless otherwise stated, the users transmit BPSK modulated symbols using a spreading sequence of length $S = 16$, and a degree of parallelization for the parallel interleaver of $M = 8$. For the AWGN channel, the variance of the noise was set to $\sigma_n^2 = N_0 / 2$ and the channel coefficients $h_k = 1$, for all k .

A. Convolutional coded system performance

To start with, the case with convolutional coding is considered. The simulation results are shown in Figure 3, where the results for $K=1, 8, 16$, and 32 users are shown. For all cases, the number of iterations in the iterative receiver was set to 10. For the case of 32 users, the system is transmitting data at a rate of 1 bit/chip, with a data block size of 512 bits.

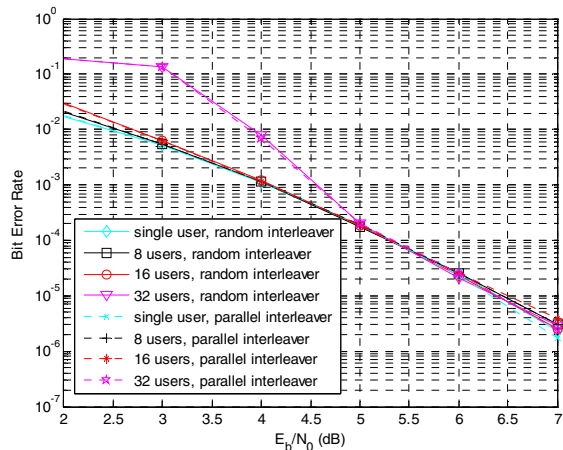


Figure 3. BER performance for a convolutional code with generators $(7,5)_8$ for different number of users, when using both parallel and random interleavers. The number of receiver iterations were set to 10, and the degree parallelization for the parallel interleaver was set to $M=8$.

As can be observed, the performance difference when using random interleavers, compared to when using parallel interleavers, is negligible. Only minor differences may be seen at high SNR, but this is most likely to do with the lack of statistics, and cannot be the base for any further conclusions. Furthermore, we also see that close to single user performance can be obtained for a system load of at least 32 users, above an E_b/N_0 of 5dB.

B. Turbo coded system performance

Next, we will look at the simulation results for a turbo coded system. For every receiver iteration, where interference cancellation is performed, 3 iterations of the turbo code were carried out. The number of receiver iterations was set to 10, and the resulting BER performance can be seen in Figure 4.

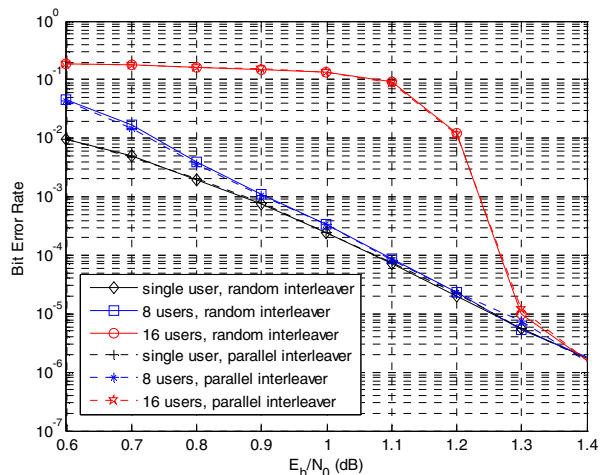


Figure 4. BER performance for a turbo code (PCCC) with generators $(13,15)_8$ for different number of users, when using both parallel and random interleavers. The number of receiver iterations were set to 10, and the degree parallelization for the parallel interleaver was set to $M=8$.

The figure shows the results for $K=1, 8$ and 16 users. Longer blocks are common when using turbo codes and therefore the block length was increased, compared to the convolutionally coded case, and yield 4096 information bits.

If we compare the performance obtained when using the parallel interleavers, with that of the random interleavers, we can see that the difference, just as for the convolutionally coded case, is negligible. We can thus conclude that the somewhat correlated inter-user interference, introduced by the parallel interleavers, does not seem to have any significant impact on the decoding of the turbo code. To further ensure that interleaver correlation does not degrade performance, simulations have been performed with shorter block lengths and no significant degradation has been observed down to 256 bits per block.

For the considered system, the BER converges to single user performance for system loads of up to 16 users, at an E_b/N_0 of about 1.3 dB.

C. Impact of parallel processing on performance

In order to investigate how the degree of parallel processing affects the performance, simulations were performed with a fixed set of parameters, apart from the degree of parallel processing, M . The considered system is the same as described in Section III A. In Figure 5, the bit error rate (BER) performance is shown for the case of $K=32$ users, for different degree of parallel processing, after 5 iterations in the receiver.

As can be seen, the degree of parallelization has a negligible impact of the BER performance for the considered system settings. For the hardware implementation, with an increase in the degree of parallelization, the processing latency could be reduced and the processing throughput increased. We can conclude from the results that the parallel interleavers preserve a good performance, even for a system with a high degree of parallelization.

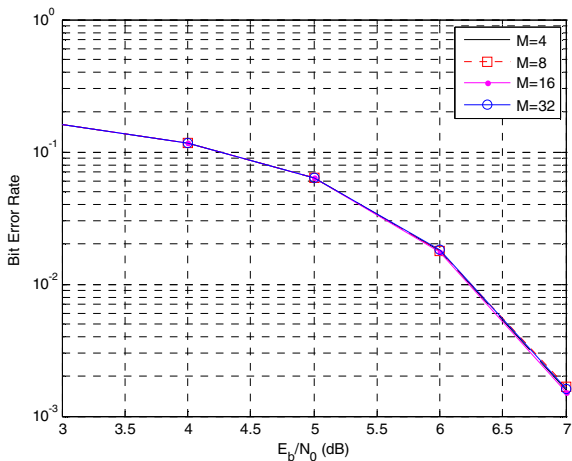


Figure 5. Comparing the BER performance for different parallelization factors used in the parallel interleaver generation. The considered system has 32 users, employing a convolutional code with generators (7,5)₈, and perform 5 receiver iterations.

In [4] a correlation metric for interleavers was presented. The metric was used to define a good set of interleavers for

user separation in IDMA. As observed in [6], even sets of interleavers with a poor correlation performance can show good performance, indicating that the correlation metric may not be the best metric. Furthermore, in [6] it is seen that different degree of parallelization gives different correlation properties. In Figure 5, it is seen that the performance of the coded IDMA systems is insensitive to the degree of parallelization, which also indicates that the system is insensitive to the correlation properties of the interleavers.

IV. CONCLUSION

In this paper we have evaluated the system performance of a coded IDMA system, when using parallel interleavers for user separation. The interleaver design has previously been presented and evaluated for an uncoded system, and with this contribution we extended the results to a coded system. Both a convolutional code, as well as a turbo code is being considered. Since coding is most likely to be used in a real system, the obtained results are important for future system design. The performance is compared to that obtained when using random interleavers.

In our results we observe that there is a negligible difference in performance when using either of the two interleavers, for both of the considered codes. This has been observed in a previous publication for an uncoded system, and we can now conclude that these results are also valid for a coded system.

Moreover, simulations also show that the degree of parallelization in the interleaver design has a minimal impact on system performance. This means that extensive parallelization of the interleaving and decoding can be performed without any significant loss in performance due to the user specific interleavers.

REFERENCES

- [1] L. Ping, L. Liu, K. Wu, and W. K. Leung, "Interleave-Division Multiple-Access," IEEE Transactions on Wireless Communication, vol. 5, no.4, pp. 938-947, April 2006.
- [2] L. Ping, L. Liu, and W. K. Leung, "A simple approach to near-optimal multiuser detection: interleave-division multiple-access," IEEE Wireless Communications and Networking Conference, WCNC 2003, vol. 4, no. 1, pp. 391-396, March 2003.
- [3] W. Hao, L. Ping, and A. Perotti, "User-specific chip-level interleaver design for IDMA systems," IEE Electronic Letters, vol. 42, no. 4, pp. 233-234, February 2006.
- [4] I. Pupeza, A. Kavcic, and L. Ping, "Efficient generation of interleavers for IDMA," IEEE International Conference on Communications, ICC 2006, vol. 4, pp. 1508-1513, June 2006.
- [5] C. Zhang and J. Hu, "The shifting interleaver design based on PN sequence for IDMA systems," 2007 International Conference on Future Generation Communication and Networking, FGCN 2007, Korea, December 2007.
- [6] S. Wu, X. Chen, and S. Zhou, "A parallel interleaver design for IDMA systems," IEEE 2009 International Conference on Wireless Communications and Signal Processing, WCSP09, China, July 2009.
- [7] 3GPP TS 25.212 v6.4.0 (2005-03)
- [8] W. Ryan, "A turbo code tutorial," unpublished paper, available at <http://www.ece.arizona.edu/~ryan/publications/turbo2c.pdf>.
- [9] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for minimizing symbol error rate", IEEE Transactions on Information Theory, vol. IT-20(2), pp.284-287, March 1974.