New Multi-User OFDM Scheme: Braided Code Division Multiple Access

Tavares, Marcos B.S.; Lentmaier, Michael; Zigangirov, Kamil; Fettweis, Gerhard

Published in:
[Host publication title missing]

DOI:
10.1109/ACSSC.2008.5074554

2008

New Multi-User OFDM Scheme: Braided Code Division Multiple Access

Marcos B.S. Tavares†, Michael Lentmaier†, Kamil Sh. Zigangirov‡ and Gerhard P. Fettweis†
† Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, 01062 Dresden, Germany
Emails: {tavares, michael.lentmaier, fettweis}@ifn.et.tu-dresden.de
‡ Department of Electrical Engineering, Unviversity of Notre Dame, Notre Dame, IN 46556, USA
Email: kzigangi@nd.edu

***Abstract*— In this paper, we present a new multiple access scheme using OFDM modulation which we call *braided code division multiple access* (BCDMA). The principle behind this novel multiple access technique is to combine in one single scheme a very efficient error correction code, interleaving for channel diversity exploitation and, simultaneously, a multiple access method for a large number of users. BCDMA is based on *braided codes*, which are iteratively decodable codes defined on graphs and provide excellent performance with relatively low decoding complexity.**

I. Introduction

There is a huge demand for higher data rates in next generation wireless systems. The typical multi-user scenario, where several users access the same medium and the same resources to perform communication, requires powerful and efficient broadband transmission techniques that enable the utilization of the spectrum resources with very high efficiency. *Orthogonal frequency division multiplexing* (OFDM) is a transmission technique that elegantly addresses this problem and is currently being considered as base technology for current and future wireless communication systems. Moreover, error correcting codes are indispensable elements to improve the overall capacity of communication systems.

With the discovery of *turbo codes* [1], coding theorists and practitioners have learned how to approach Shannon's capacity with practical codes. Later, Gallager's *low-density parity-check* (LDPC) codes [2] have been also recognized to be capacity-approaching [3]. In [4], the convolutional counterparts of Gallager's LDPC block codes, i.e., *LDPC convolutional codes* (LDPC-CCs), were introduced. The LDPC-CCs have the advantage that they are not limited to a fixed block length as in the case of the LDPC block codes. Furthermore, the Tanner graphs underlying the LDPC-CCs have some interesting properties that enable the so-called *pipeline decoding* [4] and low-complexity encoding via shift-register operations. In [5]–[9], the LDPC-CCs were generalized to the case when the variable nodes and check nodes of the underlying Tanner graph are substituted by some simple component codes (e.g., Hamming codes and convolutional codes). It was shown that these generalized LDPC-CCs – which were named *braided codes* (BCs) – have excellent performance, good distance properties, and also that they can be continuously decoded with high speeds and low complexity using pipeline decoders.

In this paper, we present a novel multiple access technique that is based on OFDM and BCs. We have denominated this new scheme *braided code division multiple access* (BCDMA).[1] As we will discuss throughout this paper, the BCDMA scheme results in a conceptual change in the organization of modems for multiple access systems. This is due to the fact that error correction coding, interleaving for diversity exploitation, modulation and multiple access – which are generally processed by separate elements of a modem – are now concentrated in one single entity that is derived from the typical two-dimensional array representation of the BCs.

II. Multiple Convolutional Permutors

Multiple convolutional permutors (MCPs) [11] are the fundamental elements in the constructions presented in the following sections. We define an MCP as a diagonal-type semi-infinite binary matrix $\mathbf{P}$ of the form

$$\mathbf{P} = \begin{pmatrix} p_{0,\delta} & p_{0,\delta+1} & \cdots & p_{0,\Delta} & 0 & 0 & 0 & \cdots \\ 0 & p_{1,\delta+1} & \cdots & p_{1,\Delta} & p_{1,\Delta+1} & 0 & 0 & \cdots \\ \cdots & & & \cdots & & & & \cdots \\ 0 & \cdots & 0 & p_{t,t+\delta} & \cdots & & p_{t,t+\Delta} & 0 \end{pmatrix}. \tag{1}$$

In (1), $0 \leq \delta \leq \Delta$ and each row has exactly Γ 1's and 0's elsewhere. Moreover, each of the first Δ columns have at most Γ 1's and all other columns have exactly Γ 1's. We also assume that there exist time instants $t = t_1$ and $t = t_2$ such that $p_{t_1,t_1+\delta} = 1$ and $p_{t_2,t_2+\Delta} = 1$.

In this context, Γ is called multiplicity of the MCP. The parameters δ and Δ are called *minimum* and *maximum permutor delays*, respectively. The *width* of the permutor is defined as $w = \Delta - \delta + 1$. Furthermore, we say that an MCP is *periodic* if and only if there exists a T such that $p_{t,t'} = p_{t+T,t'+T}$, $\forall t, t' \in \mathbb{Z}^+$. The minimal value of T for which this last equation still holds is called *period* of the MCP. An MCP with period T can be constructed from a $T \times T$ *multiple block permutor* using the *unwrapping procedure* presented in [4].

III. Braided Codes

An uncomplicated way to understand the BCs is by taking a look at their encoders. Fig. 1 shows a general encoder structure for BCs. In this picture, $\mathbf{P}^{(0)}$, $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ are MCPs with multiplicities Γ_0, Γ_1 and Γ_2, and widths w_0, w_1 and w_2, respectively. Moreover, $\mathbf{U}_t$ is a vector of information symbols that enter the systematic encoder at time instant t, and $\mathbf{V}_t^{(0)} = \mathbf{U}_t$, $\mathbf{V}_t^{(1)}$ and $\mathbf{V}_t^{(2)}$ are the vectors of coded symbols

[1] Some basic principles of BCDMA were formulated in [10].

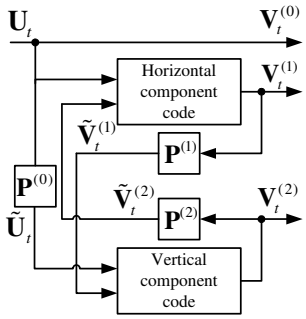

Fig. 1. Encoder structure for braided codes.

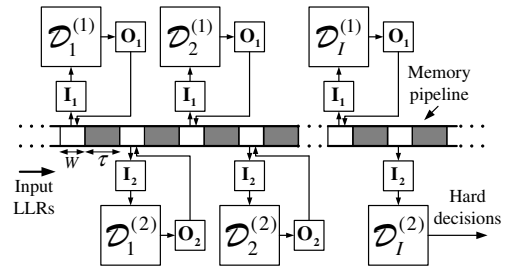

Fig. 2. Pipeline decoder for BCCs.

that leave the encoder at time instant t. The feedback of parity-bits of the horizontal encoder into the vertical one and vice versa, which can also be seen in Fig. 1, is characteristic of the BCs and results in very good error correction capabilities [5], [6].

In the remainder of this paper, we will focus on BCs that are constructed using binary rate $R = 2/3$ conventional convolutional codes as component codes. The BCs resulting from this construction are called *braided convolutional codes* (BCCs) and they will have an overall rate $R = 1/3$. Moreover, we will assume symmetric MCPs, i.e., $\Gamma_0 = \Gamma_1 = \Gamma_2 = \Gamma$ and $w_0 = w_1 = w_2 = w$. In this sense, an implementation for a BCC encoder is exemplarily shown in Fig. 3, where the MCPs $\mathbf{P}^{(0)}$, $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ with $\Gamma = 5$ and $w = 10$ correspond to the white cells of the array regions indicated by $\mathbb{P}^{(0)}$, $\mathbb{P}^{(1)}$ and $[\mathbb{P}^{(2)}]^{\mathrm{T}}$, respectively. As we can note, this implementation is based on a two-dimensional memory array, which is processed in a sliding window fashion. The encoding of the information bits is then performed as follows. At the time instant t, the vector of information symbols $\mathbf{U}_t = (\mathbf{u}_{t,0}, \cdots, \mathbf{u}_{t,\Gamma-1})$ enters the BCC encoder and its elements are stored in their corresponding positions in $\mathbb{P}^{(0)}$. Then, the pairs of symbols $(\mathbf{u}_{t,j}, \tilde{\mathbf{v}}_{t,j}^{(2)})$, $j = 0, \cdots, \Gamma - 1$, are read from the memory array and given as inputs to the horizontal component code that produce as output the vector of symbols $\mathbf{V}_t^{(1)} = (\mathbf{v}_{t,0}^{(1)}, \cdots, \mathbf{v}_{t,\Gamma-1}^{(1)})$ whose elements are stored in their corresponding positions in $\mathbb{P}^{(1)}$. A similar procedure is done for the vertical encoding. In this case, the pairs of symbols $(\tilde{\mathbf{u}}_{t,j}, \tilde{\mathbf{v}}_{t,j}^{(1)})$, $j = 0, \cdots, \Gamma - 1$, are given as inputs to the vertical component code to produce $\mathbf{V}_t^{(2)} = (\mathbf{v}_{t,0}^{(2)}, \cdots, \mathbf{v}_{t,\Gamma-1}^{(2)})$ whose elements are stored in $[\mathbb{P}^{(2)}]^{\mathrm{T}}$. By doing so, in each time instant t the output of the BCC encoder is the vector of symbols $\mathbf{V}_t = (\mathbf{V}_t^{(2)}, \mathbf{U}_t, \mathbf{V}_t^{(1)}) = (\mathbf{v}_{t,0}^{(2)}, \mathbf{v}_{t,1}^{(2)}, \mathbf{v}_{t,2}^{(2)}, \mathbf{v}_{t,3}^{(2)}, \mathbf{v}_{t,4}^{(2)}, \mathbf{u}_{t,0}, \mathbf{u}_{t,1}, \mathbf{u}_{t,2}, \mathbf{u}_{t,3}, \mathbf{u}_{t,4}, \mathbf{v}_{t,0}^{(1)}, \mathbf{v}_{t,1}^{(1)}, \mathbf{v}_{t,2}^{(1)}, \mathbf{v}_{t,3}^{(1)}, \mathbf{v}_{t,4}^{(1)})$, as it is shown by the dashed box in Fig. 3.

A. BCCs and M-ary Modulation Schemes

BCCs are essentially binary codes. However, we can use the array representation of the BCCs also to encode symbols of M-ary modulation schemes. In this case, each used cell of the memory array will store $q = \log_2(M)$ bits. In other words, the symbols $\mathbf{u}_{t,j}$, $\tilde{\mathbf{u}}_{t,j}$, $\mathbf{v}_{t,j}^{(1)}$, $\tilde{\mathbf{v}}_{t,j}^{(1)}$, $\mathbf{v}_{t,j}^{(2)}$ and $\tilde{\mathbf{v}}_{t,j}^{(2)}$ will be binary vectors of length q. The encoding is then performed similar to the binary case, however, each parity symbol is obtained by encoding the q bits of each input symbol sequentially. For instance, in order to obtain the parity symbols $\mathbf{v}_{t,j}^{(1)} = (v_{t,j,0}^{(1)}, \cdots, v_{t,j,q-1}^{(1)})$ from the symbols $\mathbf{u}_{t,j} = (u_{t,j,0}, \cdots, u_{t,j,q-1})$ and $\tilde{\mathbf{v}}_{t,j}^{(2)} = (\tilde{v}_{t,j,0}^{(2)}, \cdots, \tilde{v}_{t,j,q-1}^{(2)})$, the bits $u_{t,j,l}$ and $\tilde{v}_{t,j,l}^{(2)}$ enter the rate $R = 2/3$ component convolutional encoder to produce the bit $v_{t,j,l}^{(1)}$ for $l = 0$ to $q - 1$.

B. Decoders for BCCs

The BCCs can be decoded with a pipelined decoder architecture that enables continuous, high-speed decoding. As it can be observed in Fig. 2, the main characteristic of this kind of decoder is the fact that all I decoding iterations can be performed in parallel by $2I$ concatenated identical *processors* $\mathcal{D}_i^{(e)}$, where $i = 1, \cdots, I$ and $e = 1, 2$. Thus, after an initial decoding delay, the estimated values for the code symbols are output by the pipeline decoder at each processing cycle in a continuous manner. For the particular case of BCCs, the $2I$ parallel processors are implementing an *windowed* version of the BCJR algorithm [12], [13]. In this case, each processor operates in a finite window of length W and calculates the *a posteriori probabilities* (APP) of all code symbols (i.e., information and parity symbols).

More specifically, the decoding of BCCs using the pipeline decoder occurs as follows. At the t-th time instant, a vector of *log-likelihood ratios* (LLRs) $\mathbf{R}_t$ corresponding to the vector of code symbols $\mathbf{V}_t$ enters the memory pipeline as *a priori* information. Then, the *read-logic* $\mathbf{I}_1$ transfers the necessary data from the memory pipeline to the horizontal component code decoder $\mathcal{D}_1^{(1)}$. On its turn, the processor $\mathcal{D}_1^{(1)}$ performs the windowed BCJR decoding on the data and transfer the obtained extrinsic LLRs to the memory pipeline through the *write-logic* $\mathbf{O}_1$.[2] After this, the processor $\mathcal{D}_1^{(1)}$ starts processing new data. Observe in Fig. 2 that the current memory region used by $\mathcal{D}_1^{(1)}$ is separated from the vertical decoder $\mathcal{D}_1^{(2)}$ by τ positions. This is to avoid that different decoders operate in overlapping sets of coded bits at the same time. When the extrinsic LLRs produced by $\mathcal{D}_1^{(1)}$ reaches the decoder $\mathcal{D}_1^{(2)}$, vertical decoding is performed using the extrinsic LLRs produced by $\mathcal{D}_1^{(1)}$ as *a priori* information. The new extrinsic LLRs obtained by $\mathcal{D}_1^{(2)}$ are written into the memory pipeline and $\mathcal{D}_1^{(2)}$ starts to process new data. This process repeats

[2] $\mathbf{I}_i$ and $\mathbf{O}_i$, with $i \in \{1, 2\}$, are determined by the MCPs $\mathbf{P}^{(0)}$, $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$.

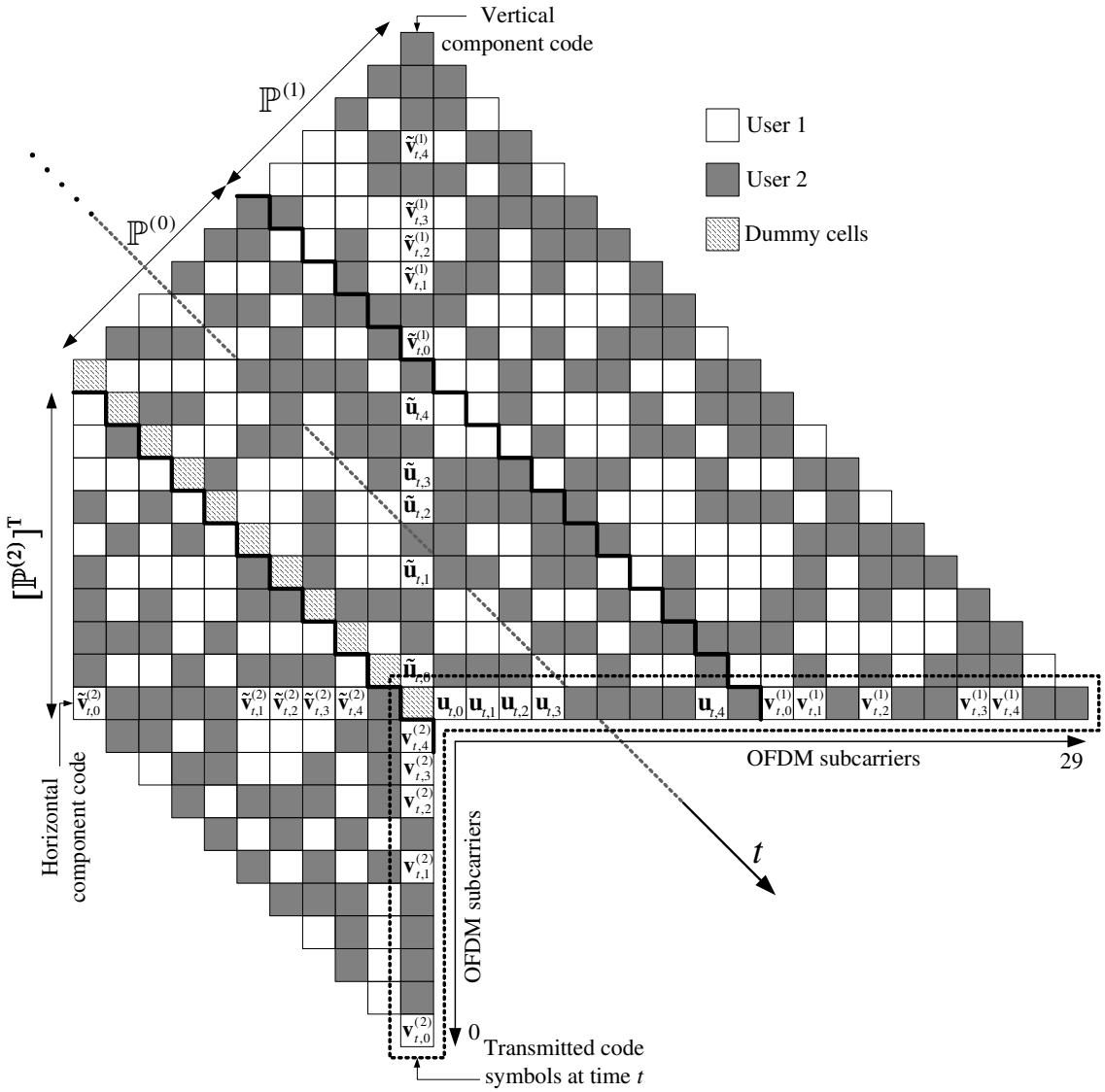

Fig. 3. Array representation of a braided convolutional code (BCC) and of a braided code division multiple access (BCDMA) scheme for two users.

through all horizontal and vertical processors until the data reaches the vertical decoder $\mathcal{D}_I^{(2)}$. This last processor outputs the final LLRs and hard decision is done to determine the values of the decoded bits.

IV. Braided Code Division Multiple Access

In addition to the array representation of a BCC, the example shown in Fig. 3 can also be interpreted as the *subcarrier allocation pattern* of a BCDMA system for two users. In this case, we map each of the array cells at time t (i.e., cells inside the dashed box) to a subcarrier of an OFDM system. For instance, the BCDMA scheme depicted in Fig. 3 would need as basis for its realization an OFDM system with at least $N_{\text{OFDM}} = 30$ subcarriers. We assume that the white cells are associated with User 1 and gray cells are associated with User 2. The striped cells are dummy cells, they do not contain any information. Moreover, in the Fig. 3, only the code symbols of User 1 are depicted in the array. If we number the OFDM subcarriers from the left to right, the symbols transmitted by User 1 at time instant t, $\mathbb{S}_1(t) = \{\mathbf{v}_{t,0}^{(2)}, \mathbf{v}_{t,1}^{(2)}, \mathbf{v}_{t,2}^{(2)}, \mathbf{v}_{t,3}^{(2)}, \mathbf{v}_{t,4}^{(2)}, \mathbf{u}_{t,0}, \mathbf{u}_{t,1}, \mathbf{u}_{t,2}, \mathbf{u}_{t,3}, \mathbf{u}_{t,4}, \mathbf{v}_{t,0}^{(1)}, \mathbf{v}_{t,1}^{(1)}, \mathbf{v}_{t,2}^{(1)}, \mathbf{v}_{t,3}^{(1)}, \mathbf{v}_{t,4}^{(1)}\}$, will be mapped to the OFDM subcarriers $\mathbb{C}_1(t) = \{0, 5, 7, 8, 9, 10, 11, 12, 13, 18, 20, 21, 23, 26, 27\}$, respectively. Accordingly, the symbols of User 2 at time t, $\mathbb{S}_2(t)$, will be mapped to the subcarriers $\mathbb{C}_2(t) = \{1, 2, 3, 4, 6, 14, 15, 16, 17, 19, 22, 24, 25, 28, 29\}$.

We can observe that the MCPs are the key components of our construction. They are responsible for defining the BCCs, as well as, the subcarrier allocation pattern of the users. In Fig. 3, we denote $\mathbb{P}^{(0)}$, $\mathbb{P}^{(1)}$ and $[\mathbb{P}^{(2)}]^{\mathrm{T}}$ as being the sets containing the MCPs $\mathbf{P}_k^{(i)}$, $i \in \{0, 1, 2\}$, $k \in \{0, \cdots, K-1\}$, of the $K = 2$ users of the system.[3] The generalization for $K \geq 3$ is straightforward. However, a fundamental condition that the sets $\mathbb{P}^{(i)}$, $i \in \{0, 1, 2\}$, must fulfill in order to define a BCDMA system is that their elements must be orthogonal to

[3] $[\mathbb{P}^{(2)}]^{\mathrm{T}}$ in the set containing the inverse MCPs $[\mathbf{P}_k^{(2)}]^{\mathrm{T}}$, $k \in \{0, \cdots, K-1\}$.

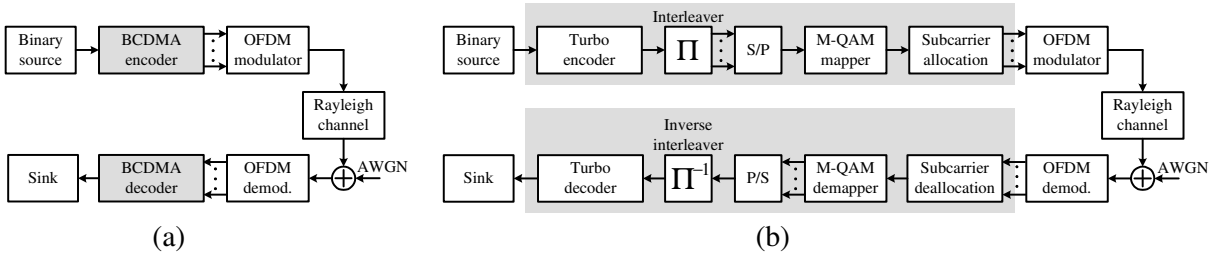

Fig. 4. Systems under study. **(a)** BCDMA system. **(b)** Bit-interleaved coded modulation (BICM) system using turbo codes.

each other. Considering the expression (1), this orthogonality condition can be formulated as

$$p_{t,t'}^{(i),k_1} \cdot p_{t,t'}^{(i),k_2} = 0, \ \forall t, t' \in \mathbb{Z}^+ \text{ and } k_1 \neq k_2, \qquad (2)$$

where $p_{t,t'}^{(i),k}$ is an element of the MCP $\mathbf{P}_k^{(i)}$, $i \in \{0, 1, 2\}$, associated with the k-th user of the system. In words, the condition imposed by (2) means that the different users of the BCDMA system do not share OFDM subcarriers to transmit their symbols at the time instant t. Thus, they do not cause interference to each other.

A. Spectrum Management for BCDMA Systems

In spite of having the subcarrier allocation pattern determined by the error control coding scheme, the BCDMA system is still very flexible concerning the spectrum management for the several users. Considering that the available spectrum is given by the number of OFDM subcarriers N_{OFDM}, some straightforward strategies for spectrum management are listed below:

a) If $\lfloor N_{\text{OFDM}}/3w \rfloor < 2$ the throughput of the users can increased or reduced by increasing or reducing the multiplicities Γ, respectively. Alternatively, different rows of the MCPs can be overlapped into a single OFDM symbol to increase the throughput, or the elements of the rows of the MCPs can be spread across different OFDM symbols to decrease the throughput.

b) If $\lfloor N_{\text{OFDM}}/3w \rfloor \geq 2$ the above mentioned techniques can also be applied. In addition, different rows of the MCPs can be placed into orthogonal sub-bands of a single OFDM symbol to increase the throughput.

Observe that by applying the spectrum managements techniques from above, the orthogonality condition in (2) must be redefined.

B. Further Remarks

There are some points that are worth discussing when considering the constructions presented in this section. As we can observe in Fig. 3, the sub-carriers belonging to a particular user are spread in frequency domain. This is a very important feature that enables the BCDMA system to exploit the *frequency diversity* of the channel. Another very important property that belongs the BCDMA system is the *frequency hopping* experienced by the users in the succession of the time instants, i.e., the subcarriers used by a particular user are different for the succession of time instants. This last property implies that good and also bad parts of the spectrum are equally (or almost equally) distributed among the users. In other words, it means that the BCDMA system is *fair*. Additionally, if BCDMA is supposed to be deployed in a cellular system with *frequency reuse*, the frequency hopping is very important for averaging the inter-cell interference.

V. Simulation Results

In this section, we evaluate the performance of the BCDMA system depicted in Fig. 4(a) by means of computer simulations. As a reference system, we use the arrangement shown in Fig. 4(b), which employs the *bit-interleaved coded modulation* (BICM) ideas presented in [14] and analyzed in [15] to improve the code diversity of coded modulation on a Rayleigh channel.

For the BCDMA system, the component convolutional codes have rate $R = 2/3$ and 4 states. Their generator matrices are given by

$$\mathbf{G}(D) = \begin{pmatrix} 1 & 0 & 1/(1+D+D^2) \\ 0 & 1 & (1+D^2)/(1+D+D^2) \end{pmatrix}. \qquad (3)$$

On the other hand, the component codes of the turbo code of the BICM system have rate $R = 1/2$ and also 4 states. Their generator matrices are given by

$$\mathbf{G}(D) = \begin{pmatrix} 1 & (1+D^2)/(1+D+D^2) \end{pmatrix}. \qquad (4)$$

Moreover, all MCPs, interleavers and the subcarrier allocation pattern of the BICM system have been randomly generated.

For both systems, the underlying OFDM system consists of $N_{\text{OFDM}} = 1024$ subcarriers and a *cyclic prefix* of length $N_{\text{CP}} = 16$, which is appended to the beginning of each transmitted OFDM symbol. Additionally, we use a Rayleigh flat fading channel in our simulations that is constant over the duration of one OFDM symbol but varies independently from symbol to symbol. Moreover, in the OFDM demodulators of both systems from Fig. 4 a linear zero-forcing equalizer with perfect channel knowledge is used to mitigate the channel distortions.

In our simulations, the BCDMA system was observed in six different configurations, which are listed below:

1. $w = 50$, $\Gamma = 5$, QPSK modulation, block length $L = 63800$ bits.

2. $w = 50$, $\Gamma = 5$, 16-QAM modulation, block length $L = 62800$ bits.

3. $w = 100$, $\Gamma = 10$, QPSK modulation, block length $L = 255200$ bits.

4. $w = 100$, $\Gamma = 10$, 16-QAM modulation, block length $L = 251200$ bits.

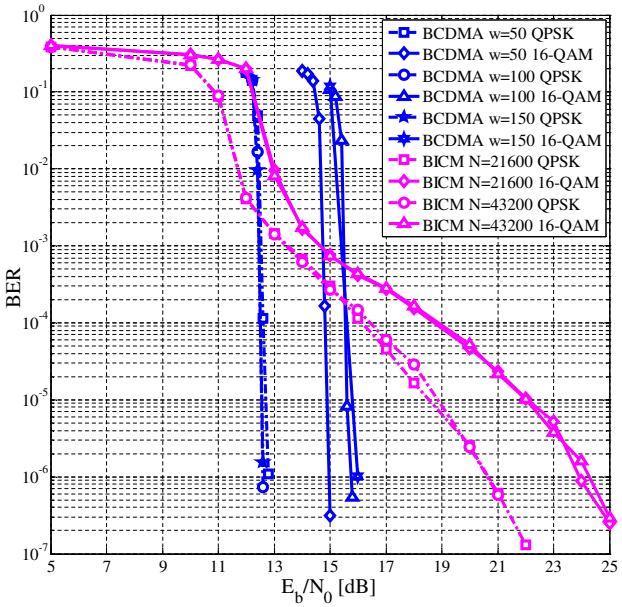

Fig. 5. BER performance of the BCDMA system

5. $w = 150$, $\Gamma = 15$, QPSK modulation, block length $L = 574200$ bits.
6. $w = 150$, $\Gamma = 15$, 16-QAM modulation, block length $L = 565200$ bits.

For the BCDMA configurations from above, the BCC encoders have been terminated with $2w\Gamma \log_2(M)$ zero bits, resulting in coding rates of $R \approx 0.323$ for QPSK and $R \approx 0.312$ for 16-QAM. Moreover, the strategy **b)** of Section IV-A has been used such that $N_{\mathrm{U}} = 90$ OFDM subcarriers are used in each OFDM symbol to transport user data.

The BICM system has also been examined in different configurations. Here, interleaver lengths of $N = 21600$ and $N = 43200$ (resulting in terminated blocks of lengths $L = 64808$ and $L = 129608$, respectively) have been combined with QPSK and 16-QAM modulations. As in the case of the BCDMA system, the subcarrier allocation module of the BICM system assigns $N_{\mathrm{U}} = 90$ subcarriers to user data in each OFDM symbol.

Fig. 5 shows the BER performance of the BCDMA system compared against the BICM system. In the simulations, a maximum of 100 decoding iterations were allowed. Moreover, at least 500 blocks were transmitted and at least 20 block errors have occurred in each simulation point. Astonishingly, the performance of the BCDMA system is about 8 dB better than the performance of the BICM system with similar parameters at a BER of 10^{-6}. This is quite an impressive result, since the component codes of both systems have almost the same trellis complexity and the interleavers of the turbo codes and of the BICM scheme are orders of magnitude bigger than the widths of the MCPs used in the BCCs. Furthermore, the steepness shown by the BER curves of the BCDMA system is an indication of the very good minimum distances of the underlying BCCs.

VI. Conclusions

In this paper, we presented the general principles of a novel multiple access scheme called *braided code division multiple access* (BCDMA). Besides of providing a flexible and efficient medium access scheme for a large number of users, the application of the BCDMA concept results in a conceptual change in the design of multiple access modems. In this case, processing tasks that are traditionally considered in separate (i.e., error correction coding, interleaving for diversity exploitation, modulation and multiple access) are now combined in one single entity. Moreover, the pipeline decoding of the underlying *braided convolutional codes* (BCCs) enables very high speed VLSI implementations. Simulation results have also shown that the BCDMA system has a considerable gain in comparison with conventional systems based on bit-interleaved coded modulation.

We consider BCDMA to be a promising research topic. Several points regarding this new technique remain open. For instance, the design of the MCPs to simultaneously maximize the error correction capabilities of the underlying BCC, frequency diversity exploitation, fairness and interference averaging property must be studied. In addition, some other issues like decoding delay reduction, VLSI implementation, rate compatibility, synchronization and channel estimation are also very interesting.

Acknowledgment

The authors are grateful for the use of the high performance computing facilities of the ZIH at the TU Dresden.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes (1)," in *Proc. IEEE International Conference on Communications*, Geneva, Switzerland, May 1993, vol. 2, pp. 1064–1070.
[2] R. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.
[3] D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, Mar 1999.
[4] A. Jiménez Feltström and K.Sh. Zigangirov, "Periodic time-varying convolutional codes with low-density parity-check matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 2181–2190, Sept. 1999.
[5] A. Jiménez Feltström, M. Lentmaier, D.V. Truhachev, and K.Sh. Zigangirov, "Braided block codes," *IEEE Trans. Inform. Theory*, June 2006, submitted for publication. Availabe online: http://www.it.lth.se/users/kamil.
[6] W. Zhang, , M. Lentmaier, K.Sh. Zigangirov, and D.J. Costello, Jr., "Braided convolutional codes: a new class of turbo-like codes," *IEEE Trans. Inform. Theory*, May 2006, submitted for publication.
[7] M. Lentmaier, D.V. Truhachev, and K.Sh. Zigangirov, "Iterative decodable sliding codes on graphs," in *Proc. of ACCT-VIII*, St. Petersburg, Russia, Sept. 2002, pp. 190–193.
[8] D.V. Truhachev, M. Lentmaier, and K.Sh. Zigangirov, "On braided block codes," in *Proc. of IEEE International Symposium of Information Theory (ISIT'03)*, Yokohama, Japan, June 2003, p. 32.
[9] W. Zhang, M. Lentmaier, K.Sh. Zigangirov, and D.J. Costello, Jr., "Braided convolutional codes," in *Proc. of IEEE International Symposium of Information Theory (ISIT'05)*, Adelaide, Australia, Sept. 2005, pp. 592–596.
[10] K.Sh. Zigangirov and M.B.S. Tavares, "Braided code division multiple access," *Multiple access channels: theory and practice*, edited by E. Biglieri and L. Györfi, IOS Press, pp. 191-213, 2007.
[11] A. Jimenéz Feltström, *Iteratively Decodable Convolutional Codes: Analysis and Implementation Aspects*, Ph.D. thesis, Lund University, Lund, Sweden, 2006.
[12] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "A soft-input soft-output maximum a posteriori (MAP) module to decode parallel and serial concatenated codes," *JPL TDA Progress Report*, vol. 42, no. 127, pp. 1–20, Nov. 1996.
[13] A.J. Viterbi, "An intuitive justification of the MAP decoder for convolutional codes," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 260–264, Feb. 1998.
[14] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.
[15] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.