User-specific chip-level interleaver design for IDMA systems

H. Wu, L. Ping and A. Perotti

A user-specific interleaver design for interleave-division multiple-access (IDMA) systems is proposed. This method can solve the memory cost problem for chip-level interleavers, and reduce the amount of information exchange between mobile stations and base stations to specify the interleaver used as their identifications.

Introduction: The multiple access scheme recently studied in [1, 2] relies on interleaving as the only means for user separation, and hence it is referred to as interleave-division multiple-access (IDMA). IDMA not only inherits many advantages from conventional code-division multiple-access (CDMA), such as robustness against fading and mitigation of cross-cell interference, but also allows a very simple chip-by-chip (CBC) iterative multiuser detection (MUD) strategy while achieving impressive performance [1, 2]. In multipath channels, the per-user complexity of the IDMA MUD algorithm [2] is $O(L)$, which is independent of the user number K and linear with the channel tap number L. It is much lower than that of other alternatives, such as $O(2^K)$ for the maximum *a posteriori* (MAP)-based method [3] and $O(K^2)$ for the well-known minimum mean squared error (MMSE) method [4].

In IDMA systems, different interleavers are assigned to different users. In theory, the user-specific interleavers can be generated independently and randomly. If this is the case, the base station (BS) has to use a considerable amount of memory to store these interleavers, which may cause serious concern when the number of users is large. Also, during the initial link setting-up phase, there should be messages passing between the BS and mobile stations (MSs) to inform each other about their interleavers. Extra bandwidth resource will be consumed for this purpose if the interleavers used by the BS and MSs are long and randomly generated.

In this Letter, we examine a power-interleaver generation method to alleviate this concern. With this method, the interleaver assignment scheme is simplified and memory cost is greatly reduced without sacrificing performance.

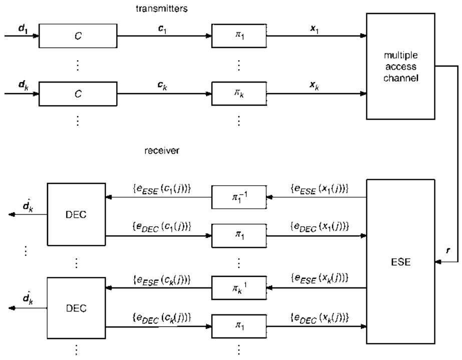

Fig. 1 *Transmitter and receiver structures of IDMA system with K simultaneous users, where π_k is the interleaver for user k*

System model: As shown in Fig. 1, we consider an IDMA system with K simultaneous users in a quasi-static single-path channel. At the transmitter, the length-N input data sequence $\boldsymbol{d}_k = [d_k(1), \ldots, d_k(i), \ldots, d_k(N)]^T$ of user k is encoded into $\boldsymbol{c}_k = [c_k(1), \ldots, c_k(j), \ldots, c_k(J)]^T$ based on a low-rate code C, where J is the frame length. We call the elements in $\boldsymbol{c}_k$ 'chips'. Then $\boldsymbol{c}_k$ is permutated by a chip-level interleaver π_k, producing the transmitted chip sequence $\boldsymbol{x}_k = [x_k(1), \ldots, x_k(j), \ldots, x_k(J)]^T$. Normally, C is constructed by serially concatenating a forward error correction (FEC) code C_{FEC} and a length-S repetition code C_{REP}. The channel observation at the receiver side $\boldsymbol{r} = [r(1), \ldots, r(j), \ldots, r(J)]^T$ is represented by

$$r(j) = \sum_{k=1}^{K} h_k x_k(j) + n(j), \quad j = 1, 2, \ldots J \tag{1}$$

where h_k is the fading coefficient related to user k, and $\{n(j)\}$ are samples of an additive white Gaussian noise (AWGN) process with zero-mean and variance $\sigma^2 = N_0/2$. For simplicity, we only consider real $\{h_k\}$, but the result can be easily extended to quadrature channels [2]. Assume that $\{h_k\}$ are known *a priori* at the receiver.

The CBC receiver consists of an elementary signal estimator (ESE) and a bank of K single-user *a posteriori* probability (APP) decoders (DECs), operating in an iterative manner. For simplicity, we assume binary phase shift keying (BPSK) signalling, i.e. $x_k(j) \in \{+1, -1\}, \forall k, j$. The outputs of the ESE and DECs are extrinsic log-likelihood ratios (LLRs) about $\{x_k(j)\}$ defined as

$$e(x_k(j)) = \log\left(\frac{\Pr(x_k(j) = +1)}{\Pr(x_k(j) = -1)}\right), \forall k, j \tag{2}$$

Further denote by $e_{ESE}(x_k(j))$ and $e_{DEC}(x_k(j))$ the LLRs generated by the ESE and DECs [5].

For specific user k, we rewrite (1) as

$$r(j) = h_k x_k(j) + \zeta_k(j) \tag{3}$$

where

$$\zeta_k(j) = \sum_{k' \neq k} h_{k'} x_{k'}(j) + n(j) \tag{4}$$

represents a distortion term with respect to $x_k(j)$. From the central limit theorem, $\zeta_k(j)$ can be approximated by a Gaussian random variable if K is large. Denote by $\mathrm{E}(\cdot)$ and $\mathrm{Var}(\cdot)$ the mean and variance functions, respectively. We briefly list the serial CBC detection algorithm [1, 2] as follows. In a serial scheme, the ESE operations and the APP decoding are carried out user-by-user.

The Serial CBC Algorithm:

Initially, set $e_{DEC}(x_k(j)) = 0, \ \forall k, j$.

Step 1: Set $k = 1$.

Step 2: The ESE performs the following operations for user k:

$$\mathrm{E}(x_k(j)) = \tanh(e_{DEC}(x_k(j))/2), \quad \forall j, \tag{5a}$$

$$\mathrm{Var}(x_k(j)) = 1 - (\mathrm{E}(x_k(j)))^2, \quad \forall j, \tag{5b}$$

$$\mathrm{E}(\zeta_k(j)) = \sum_{k' \neq k} h_{k'} \mathrm{E}(x_{k'}(j)), \quad \forall j, \tag{5c}$$

$$\mathrm{Var}(\zeta_k(j)) = \sum_{k' \neq k} |h_{k'}|^2 \mathrm{Var}(x_{k'}(j)) + \sigma^2, \quad \forall j, \tag{5d}$$

$$\begin{aligned} e_{ESE}(x_k(j)) &= \log\left(\frac{\Pr(x_k(j) = +1 | r(j))}{\Pr(x_k(j) = -1 | r(j))}\right) \\ &= \frac{2h_k(r(j) - \mathrm{E}(\zeta_k(j)))}{\mathrm{Var}(\zeta_k(j))}, \quad \forall j. \end{aligned} \tag{5e}$$

Step 3: De-interleaving $e_{ESE}(c_k(j)) = \pi_k^{-1}(e_{ESE}(x_k(j)))$ is performed and then APP decoding [5] is performed in the DEC.

Step 4: The generated LLRs $\{e_{DEC}(c_k(j)), \forall j\}$ are re-interleaved before sent to the ESE as feedbacks. Then the mean and variance for each chip are updated. When $k < K$, we go back to step 2 for the next user with $k \Leftarrow k + 1$. When $k = K$, this iteration is finished, and we recommence at step 1 for the next iteration.

During the final iteration, the DECs produce hard decisions on information bits $\{d_k(i), \forall k, i\}$.

Power-interleaver generation method: As mentioned above, in IDMA systems, K user-specific interleavers $\{\pi_k\}$ with length J are used. Storing the indexes for these interleavers requires a considerable amount of memory. Meanwhile each MS should also inform the BS about its interleaving scheme, which may cost much bandwidth if it is not handled properly.

We propose a 'power-interleaver' method. Assume that we have a master interleaver ϕ. Then we can generate the K interleavers in Fig. 1 using $\pi_k \equiv \phi^k$. Here, $\phi^k(\boldsymbol{c})$ is defined as $\phi^1(\boldsymbol{c}) \equiv \phi(\boldsymbol{c})$, $\phi^2(\boldsymbol{c}) \equiv \phi(\phi(\boldsymbol{c}))$, $\phi^3(\boldsymbol{c}) \equiv \phi(\phi(\phi(\boldsymbol{c})))$, etc. In this way, every interleaver is a 'power' of ϕ. The rationale for this method is that if ϕ is an 'ideal' random permutation, so are all $\{\phi^k\}$, and these permutations are also approximately independent to each other. Based on this method, we simply assume that the BS assigns the power index k to each user k, and then ϕ^k will be generated at the MS for user k accordingly.

This method not only reduces the amount of information exchanged between the BS and MSs, but also greatly reduces the memory cost. Consider the storage requirement during MUD at the BS. Return to the serial CBC algorithm and define $\pi_1 \equiv \phi$, where ϕ is the master interleaver. After completing the detection cycle for user 1, the interleaver can be updated in place from $\pi_1 = \phi^1$ to $\phi(\pi) = \phi^2$ for user 2. (Here 'in place' means that ϕ^1 is discarded but the memory is used to store ϕ^2.) This procedure continues recursively. After completing the detection cycle for user k, we change the interleaver in place from $\pi_k = \phi^k$ to $\pi_{k+1} = \phi(\pi_k) = \phi^{k+1}$. With this technique, it is only necessary to store a unique master interleaver ϕ plus an intermediate interleaver π_k at any detection stage at the BS.

Moreover, the generation of interleaver π_k is also very simple at the MS of user k. When $k = 2^n$, where $n > 1$ is an integer, only $n - 1$ cycles are required with some intermediate variables generated, such as ϕ^2, $(\phi^2)^2$ and $((\phi^2)^2)^2$. Suppose that each intermediate variable can be temporarily stored. Thus when $2^{n-1} < k < 2^n$, it needs extra $n - 1$ cycles at most to 'multiply' these intermediate variables together. Therefore, the generation process of any interleaver takes a maximum of $2(n - 1)$ cycles. Let us take $k = 15$, $n = 4$ for example. Since $\pi_{15} = \phi^{15} = ((\phi^2)^2)^2 \cdots (\phi^2)^2 \cdots \phi^2 \cdots \phi$, it takes 6 cycles.

The simulation results below show that the power-interleaver performs nearly as well as the random-interleaver, regardless of whether we consider a short or long frame length scenario.

Numerical results: For simplicity, assume IDMA systems with BPSK signalling in single path AWGN channels and $h_k = 1, \forall k$. Without loss of generality, we use a uniform C_{REP} $\{+1, -1, +1, -1, \ldots\}$ with $S = 16$ for all users and 20 iterations. In Fig. 2, uncoded IDMA cases are considered, i.e. without C_{FEC} coding. The data length is 256 bits for $K = 16$, and 64 bits for $K = 28$, respectively. In Fig. 3, rate-1/2 nonsystematic convolutional code with generators $(23, 35)_8$, as C_{FEC}, are employed, and data sequences with length 256 are used for different user number cases. From these two Figures, the performances of both kinds of interleavers are almost the same, which means that the proposed power-interleaver can entirely replace the random-interleaver.

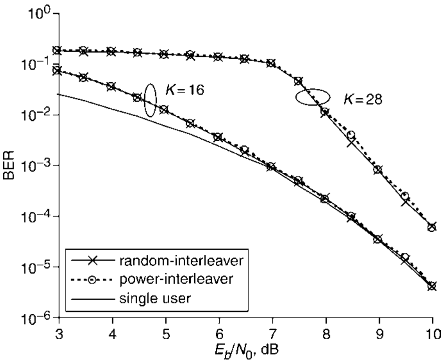

Fig. 2 *Comparison of uncoded IDMA systems in single path AWGN channels, using two kinds of interleavers*

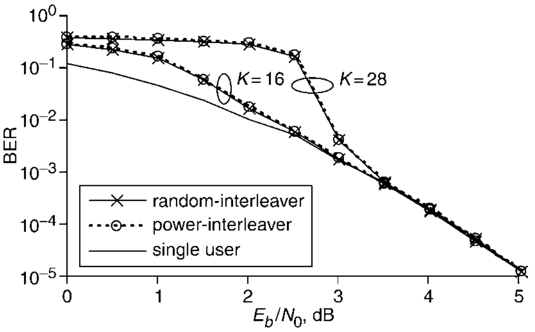

Fig. 3 *Comparison of convolutional coded IDMA systems in single path AWGN channels, using two kinds of interleavers*

Conclusion: The new interleaver generation method, so-called 'power-interleavers', can take the place of random-interleavers without performance loss. This method can greatly facilitate the design and implementation issues for IDMA systems.

Acknowledgment: This work was fully supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China [Project No. CityU 1164/03E].

25 October 2005
Electronics Letters online no: 20063770
doi: 10.1049/el:20063770

H. Wu and L. Ping (*Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong*)

E-mail: Jason.Wu@student.cityu.edu.hk

A. Perotti (*Center for Multimedia Radio Communications (CERCOM) — Politecnico di Torino, I-10129 Torino, Italy*)

References

1 Li, P., *et al.*: 'Approaching the capacity of multiple access channels using interleaved low-rate codes', *IEEE Commun. Lett.*, 2004, **8**, (1), pp. 4–6
2 Li, P., *et al.*: 'Interleave-division multiple-access', *IEEE Trans. Wirel. Commun.*, (to appear)
3 Moher, M.: 'An iterative multiuser decoder for near-capacity communications', *IEEE Trans. Commun.*, 1998, **46**, (7), pp. 870–880
4 Wang, X., and Poor, H.V.: 'Iterative (turbo) soft interference cancellation and decoding for coded CDMA', *IEEE Trans. Commun.*, 1999, **47**, (7), pp. 1046–1061
5 Berrou, C., and Glavieux, A.: 'Near optimum error correcting coding and decoding: turbo-codes', *IEEE Trans. Commun.*, 1996, **44**, (10), pp. 1261–1271