Abstract—We consider a low-cost code shift division multiple-access (CSDMA) scheme, in which user-specific shifting is used to replace user-specific interleaving in interleave division multiple access (IDMA). We also outline a low-cost Gaussian approximation (GA)-based linear minimum mean square error (LMMSE) message passing detection technique for CSDMA. We show that CSDMA can offer almost the same performance as the original IDMA in low-density parity-check (LDPC) or turbo coded systems, but with considerably lower implementation cost.

Index Terms—Multiple-access, IDMA, shifting, short cycle problem

I. INTRODUCTION

Inspired by turbo and low-density parity-check (LDPC) codes [1], [2], interleave-division multiple-access (IDMA) was proposed as a multiple-access (MA) scheme based on user-specific interleaving [3]. IDMA can outperform conventional direct-sequence code-division multiple-access (DS-CDMA) under iterative detection [4], as explained in Sect. II below. IDMA is more convenient for high rate systems since interleaving does not incur any rate loss. IDMA has also other attractive features such as timing acquisition [5], channel estimation [6] and frequency offset compensation [7].

However, user-specific interleaving may raise hardware cost. (Recall that interleaving represents a considerable part of hardware cost in an LDPC decoder.) Low-cost interleaver design based on shifting is discussed in [8]. Separate interleavers for MA and forward error control (FEC) coding are still assumed in [8]. An MA scheme is discussed in [9] based on user-specific repeat-accumulate (RA) codes. Such codes can be designed by combining interleaving for MA in IDMA with interleaving in underlying coding. (See Sect. II-A below.)

In this letter, we propose a scheme combining works in [8] and [9]. We show that user-specific interleaving in IDMA can be equivalently replaced by user-specific cyclic shifting of an underlying LDPC code, which can potentially reduce hardware cost. For convenience of discussion, we name this equivalent scheme of IDMA as code shift division multiple-access (CSDMA) scheme.

We will also briefly discuss a low cost detection technique for CSDMA based on Gaussian approximation (GA) and linear minimum mean square error (LMMSE) principles. We will demonstrate excellent performance of CSDMA in LDPC and turbo coded systems.

Recently, IDMA has been discussed for its potential applications [10], [11] in the 5th generation (5G) cellular systems. This letter provides more evidence for the advantages of IDMA. After the submission of the first version of this letter, we also observed that shifting based MA has been recently mentioned in a 5G proposal [12].

II. CODE SHIFT DIVISION MULTIPLE-ACCESS (CSDMA)

A. IDMA Principles

We start with a brief review of IDMA. Consider a system with \(K\) single-antenna users transmitting data to an \(M\)-antenna base station. Let \(\{c_k(n), n = 1, 2, ..., N\}\) be a length-\(N\) codeword generated by user \(k\). The transmitted symbols \(\{x_k(j
shown in Fig. 2(a). This reduces short cycles statistically. For example, compared with Fig. 1, \{c_1(1), c_1(3), c_2(1), c_2(3)\} no longer form a size-4 cycle after interleaving in Fig. 2(a).

Two types of interleavers are used in Fig. 2(a), one for MA and one for LDPC coding. We can combine them, resulting in Fig. 2(b). In this way, only one interleaver is used by each user. Note that the indexes of \{c_k(j)\} are shuffled after such combining. This scheme is equivalent to that in [9] where an RA code is used.

**B. Realization by shifting**

The scheme in Fig. 2(b) involves user-specific interleaving in LDPC coding. The complexity in designing and implementing a separate interleaver structure for each individual user can be a concern in practice. We consider a simple solution below.

The shifted version of an ideal random interleaver results in another independent random interleaver. This holds approximately true for an approximately random interleaver. Based on this intuition, Fig. 2 can actually be realized by user-specific cyclic-shifting of the same code. This results in the CSDMA transmitter structure in Fig. 3, where the codeword of user \(k\) is shifted by a user-specific amount \(s_k\) before transmission. For convenience of discussion, we will refer to such operation as a shifting interleaver (SI), as opposed to a general interleaver (GI). From implementation point of view, SI is much simpler than GI. An SI is defined by an integer, while a GI by a length-\(N\) vector. A GI may occupy a considerable portion of chip area (for example, 18.89% in the design of an LDPC code [13]) and may suffer from a memory conflict problem in parallel implementation [14]. These are not problems for the simple structure of an SI.

In the above discussions, we assume that the underlying FEC code involves random (or approximately random) interleaving. An LDPC code naturally meets this requirement. The situation is different for a convolutional code. For example, Fig. 4 shows the factor graph of a rate-1/2 convolutional code. A convolutional coded multi-user system can be obtained by replacing the LDPC code graph for each user in Fig. 1 by
that in Fig. 4. In such a system, no matter how to shift, there will be a lot of size-4 cycles related to the regular local constraints in Fig. 4. In this case, an extra full interleaving becomes necessary to ensure good performance.

A turbo code is more complicated. A component convolutional code in a turbo code also involves strong local correlation. However, the interleaver between two component codes in a turbo code results in global information flow during iterative detection, which may relieve the short cycle problem. This issue is difficult to analyze, but we observed good performance for turbo coded CSDMA via simulation. A more rigorous analysis is still required for this issue.

III. GA-LMMSE RECEIVER

We consider a GA-based LMMSE message passing detection technique. Denoted by DEC $k$ the decoder for user $k$. We define two messages: $LLR_{\text{DEC}}(x_k(j))$ and $LLR_{\text{ESE}}(x_k(j))$ (here ESE means elementary signal estimation) that are, respectively log-likelihood ratios (LLRs), generated by DEC $k$ and the $j$th MA node, as shown in Fig. 2(b). The discussions for $LLR_{\text{DEC}}(x_k(j))$ follow the standard LDPC decoding principles [2] and so will be omitted. We will focus on the computation for $LLR_{\text{ESE}}(x_k(j))$ below.

We rewrite (1) as

$$y(j) = Hx(j) + \eta(j),$$

(2)

where $H = [h_1, h_2, \ldots, h_K]$ and $x = [x_1(j), x_2(j), \ldots, x_K(j)]^T$. In the following part of this section, we assume each $x_k(j)$ a binary phase shift keying (BPSK) signal. Quadrature phase shift keying (QPSK) and other modulations can be processed in a similar way.

Based on $LLR_{\text{DEC}}(x_k(j))$ from DEC $k$, the a priori mean of $x_k(j)$ is

$$\hat{x}_k(j) \equiv \mathbb{E}(x_k(j)) = \Pr(x_k(j) = +1) - \Pr(x_k(j) = -1).$$

(3)

In the beginning of the iterative detection, the outputs of DEC $k$ are not available. Then we set all $\hat{x}_k(j) = 0$ (assuming that the mean of the signal constellation is zero). Following the LMMSE principle [15], we compute

$$\hat{V} = (\sigma^2 - 2H^TH + V^{-1})^{-1}$$

or equivalently, (which has lower cost if the row number of $H$ is smaller than its column number),

$$\hat{V} = V - VH^H(2\sigma^2I + HVH^H)^{-1}HV$$

and

$$\hat{x}(j) = \hat{V}(\sigma^2 - 2H^Ty(j) + V^{-1}\hat{x}(j)),$$

(4c)

where $V = \text{diag}(\hat{v}_1(j), \hat{v}_2(j), \ldots, \hat{v}_K(j))$ and $x(j) = [\hat{x}_1(j), \hat{x}_2(j), \ldots, \hat{x}_K(j)]$. In the above, $\hat{v}_k(j) \equiv \text{Var}(x_k(j))$ is the a priori variance for $x_k(j)$. The exact value of $\hat{v}_k(j)$ can be computed as

$$\hat{v}_k(j) = \text{Var}(x_k(j)) = 1 - (\hat{x}_k(j))^2,$$

(5)

which is a function of $j$. We observed that performance is insensitive to $\hat{v}_k(j)$. To reduce complexity, we take the approximation

$$\hat{v}_k(j) \approx v_k, \forall j.$$  

(6)

To estimate $v_k$, we can take the average of some samples of $\hat{v}_k(j)$. We observed that a small number of samples (such as 20) are usually sufficient. We obtain the output messages from (4) as follows:

- The a posteriori mean for $x_k(j)$ is given by $\hat{x}_k(j)$ where $\hat{x}_k(j)$ is the $k$th element of $\hat{x}(j)$.
- The a posteriori variance for $v_k(j)$ is given by $\hat{v}_k(j)$ where $\hat{v}_k(j)$ is the $k$th diagonal element of $\hat{V}$.

We then compute the extrinsic mean and variance according to the Gaussian message combining rule [16] as follows:

$$v_k^\text{ex} = (\hat{v}_k - v_k)^{-1},$$

(7a)

$$x_k^\text{ex}(j) = v_k^\text{ex}(\hat{x}_k(j) - \hat{x}_k(j)/v_k).$$

(7b)

Finally, the decoder inputs are given in LLR values as

$$LLR_{\text{ESE}}(x_k(j)) = 2x_k^\text{ex}(j)/v_k^\text{ex}.$$  

(8)

IV. NUMERICAL RESULTS

The simulation results in Figs. 5-6 compare CSDMA and standard IDMA based on, respectively, user-specific shifting and user-specific MA interleaving. The two schemes are different only in the choices of interleavers if shifting is viewed as a special case of interleaving. The following common settings are used in Figs. 5-6: independent identically distributed (IID) quasi-static Rayleigh block fading, QPSK, and $M = 4$. The shift of user $k$ in CSDMA is $\tau_k = k \times |N/K|$ where $N$ is the codeword length. No spreading is used.

Fig. 5 compares the bit error rates (BERs) of CSDMA and IDMA with rate-1/2 LDPC coding. We can see that, for all $K$, the two schemes have almost the same performance. This is consistent with the discussion in Sect. II-B that shifting an LDPC code has similar effect as random interleaving. Note that a very high overall rate of 8 is achieved in Fig. 5. This represents very high overloading [17].

Fig. 6 compares CSDMA and IDMA with rate-1/2 convolutional coding and rate-1/2 turbo coding respectively. We
can see an error floor with convolutional coding for CSDMA, which is due to the short cycle problem discussed in Sect. II-B. For turbo coding, the two schemes perform similarly. This is related to the discussion in Sect. II-B that the global information flow may overcome the effect of correlations caused by some short cycles.

It is interesting to compare different coding methods in Fig. 6. Although a turbo code generally outperforms a convolutional one in single-user detection, the situation is different in multi-user detection (MUD). It can be shown that MUD performance can be improved by matching the extrinsic information transfer (EXIT) functions [18] for the two types of messages discussed in Section III, i.e., \( \{LLR_{\text{DEC}}(x_k(j))\} \) and \( \{LLR_{\text{ESE}}(x_k(j))\} \). It can also be shown that convolutional coding results in better matching than turbo coding in Fig. 6. Some discussions on such matching can be found in [19]. The details are beyond scope of this letter.

We observed similar performance comparisons between CSDMA and IDMA for other \( M \) values. We also observed that only 8-10 iterations (between ESE operations and DECs) and 20 samples for variance approximation are sufficient for a good performance.

There is a subtle issue if repetition coding (equivalent to spreading) is used to support a large \( K \). Let \( c_k \) be the LDPC codeword of user \( k \). Several replicas of \( c_k \) are generated after repetition coding. In this case, we should assign different amounts of shifting to different replicas, which should be user-specific as well replica-specific. Otherwise short-cycles can become a problem due to the regular structure related to repetition. We observed good performance by properly choosing different amounts of shifting for such case.

V. CONCLUSION

We discussed low-cost implementation techniques for IDMA involving user-specific shifting at the transmitter and GA-LMMSE detection at the receiver. Some of the software used in this letter are available in the following site: http://www.ee.cityu.edu.hk/~z7eling/Research/Simulationpackage/

REFERENCES


[12] Transmitter side signal processing schemes for NOMA, document R1-1804396, Samsung, 3GPP TSG RAN WG1 Meeting #92Bis, Sanya,China, Apr. 2018.


