Analysis and Design of Interleavers for Iterative Multiuser Receivers in Coded CDMA Systems

Alberto Tarable, Member, IEEE, Guido Montorsi, Senior Member, IEEE, and Sergio Benedetto, Fellow, IEEE

Abstract—We deal with the design of interleavers in a coded code-division multiple-access (CDMA) scenario, where at the receiver an iterative turbo-like structure to perform multiuser detection is employed. The choice of the interleavers affects both the maximum-likelihood (ML) performance and the impact of the suboptimality of the iterative receiver. First, heuristic criteria of goodness for a set of interleavers, each assigned to a given active user, are introduced and motivated. One of these criteria is based on the intersection between the equivalent codes seen after the interleavers for each user pair. The design rules are valid for any kind of channel code. In particular, when the channel code used by every user is a terminated convolutional code, a very simple design rule, in the subset of congruential interleavers, is specified.

The suitability of an interleaver set to iterative decoding is also treated. The analysis leads to a design rule which is shown to have great importance on the performance of a turbo-like receiver. Numerical results assess the validity of the derived design rules by showing that, for iterative multiuser receivers and reasonable block lengths, the suitability to iterative decoding is more important than the performance optimization.

Index Terms—Coded code-division multiple access (CDMA), interleaver design, iterative multiuser receivers, permutations.

I. INTRODUCTION

THE astonishing success of turbo decoding has brought many researchers to apply the so-called turbo principle to channel estimation [13], equalization [17], magnetic recording [18], and other fields of research. An important application is in iterative multiuser receivers for coded code-division multiple-access (CDMA) systems. These types of receivers, which promise good performance with an affordable complexity, exploit the analogy between the CDMA multiuser transmitter side and a serial concatenation of codes [11]. This analogy is extended to the receiver, in which two soft-input soft-output (SISO) blocks exchange information through several iterations. The first block is basically a multiplier detector, called hereafter SISO user separator (SISO-US). This block performs some filtering on the received signal in such a way as to separate different users, hence the name. Its outputs are sent to a bank of $K$ independent SISO channel decoders, where $K$ is the number of users. This bank of decoders can be thought of as the outer SISO in a turbo decoder for serially concatenated codes. The decoders provide a feedback to the SISO-US, used in the following iteration to improve user separation, and so on. Examples of iterative multiuser receivers can be found in [16], [20], [21], [1], [2], [10], [19].

Going further in the analogy to turbo decoding, it has been noted that generally performance improves when each user interleaves his bitstream after channel encoding. Interleavers employed in the literature for such systems are usually randomly chosen. To the authors’ knowledge, there are very few papers [4] on interleaver design for multiuser transceivers and, while there is a relatively large literature on interleavers for turbo codes, it is clear that the design is quite different in this new scenario, because, for example, there are $K$ interleavers instead of only one. The goal of this paper is to address the issue of interleaver design for iterative multiuser receivers.

Basically, like in turbo decoding, the goal of interleaving is twofold.

- Improving the maximum-likelihood (ML) bound, that is, the system performance when an ML receiver is employed.
- Minimizing the inter-iteration gain reduction (IIGR), that is, the well-known phenomenon for which, along the iterations, the gain is less and less, until there is no improvement while performing further iterations. This effect cannot be eliminated because of the intrinsic mechanism of the iterative receiver, it can only be reduced.

Most papers that deal with interleaver design for turbo codes adopt the ML point of view. In the multiuser scenario also, Bruck et al. [4] find upper bounds on the ML performance, depending on the chosen interleavers.

Our paper faces both points of view, and two heuristic rules will be given to measure the suitability of a set of interleavers to be employed in a multiuser transmitter that is coupled to an iterative turbo-like receiver. The conclusions of our paper are that, for channel codes and block lengths usually employed in conjunction with iterative multiuser receivers, the IIGR point of view (or, at least, the corresponding rule we give) seems to give more important practical results.

What is crucial to point out here is that interleaver design is independent of the SISO-US chosen, as long as the signal-to-noise ratio (SNR) is high enough for the iterative receiver to converge to ML performance.

The paper is organized in the following manner: in Section I, the system to which we refer throughout the paper is described. In Section II, design rules from the ML point of view are derived. In Section III, these rules are analyzed in depth and we
give a design recipe for the particular subset of congruential interleavers and convolutional codes. In Section IV, we analyze interleavers from the point of view of suitability to iterative decoding. In Section V, some simulation results are shown. In Section VI, finally, we draw some conclusions.

II. SYSTEM DESCRIPTION

Let us consider \( K \) bit- and frame-synchronous users transmitting simultaneously on an additive white Gaussian noise (AWGN) channel. Since encoding is made separately for each frame, we can adopt a one-shot approach, similar to the one in [14], and consider only one frame. The \( j \)th user encodes his information binary input \( d_j \) into a coded binary block \( b_j \), which enters interleaver \( \Pi_j \). The employed channel block code, possibly derived from a convolutional code by trellis termination, takes \( k \) information bits and produces \( n \) coded bits, yielding the code rate \( R_c = k/n \). The interleaved coded bit stream is then mapped onto a binary phase-shift keying (BPSK) signal constellation and multiplied by a spreading sequence. The resulting spread-spectrum signal is then sent to the channel. The upper part of Fig. 1 shows the block diagram of the transmitter.

At the receiver, a bank of matched filters (MF) produces, in the \( i \)th-bit interval \( i = 1, \ldots, n \) the following \( K \)-dimensional column vector:

\[
y[i] = RAh[i] + n[i]
\]

where

\[1\] in this paper, \( \sim \) is used on top of quantities before interleaving or after deinterleaving.

- \( R \) is the \( K \times K \) correlation matrix between the spreading sequences assigned to the users;
- \( A \) is the \( K \times K \) diagonal matrix of amplitudes;
- \( b_j[i] \in \{-1, 1\}^K \) is the vector of modulated bits sent by all users in the \( j \)th interval;
- \( n[i] \) is a vector of Gaussian noise samples, with covariance matrix \( E[n[i]n[i]^T] = \sigma^2 R \), \( \sigma^2 \) being the noise variance.

The MF output enters the SISO-US, which possibly performs a processing on \( y \), such as a minimum mean-square-error (MMSE) filtering, and sends soft output, in the form of log-likelihood ratios (LLRs) to the \( K \) SISO channel decoders. Let us suppose that the filter output for user \( k \) can be written in the following form:

\[
\hat{y}_k[i] = \hat{a}_{kk} A_k b_k[i] + \sum_{j \neq k} \hat{a}_{kj} A_j b_j[i] + \hat{n}_k[i]
\]

where \( b_j[i] \) is the \( j \)th bit of the \( i \)th user and \( \hat{a}_{kj} \) the residual correlation between users \( k \) and \( j \) after filtering \((\hat{a}_{kk} > 0)\), which depends on the SISO-US. The filtered Gaussian noise is denoted by \( \hat{n}_k[i] \). From \( \hat{y}_k[i] \), the LLRs for user \( k \) are computed, either exactly or approximately, according to the following equation:

\[
\text{LLR}_k[i] = \log \frac{\Pr(\hat{y}_k[i]|b_k[i] = 1)}{\Pr(\hat{y}_k[i]|b_k[i] = -1)}.
\]

The SISO channel decoder for user \( k \) takes as input the deinterleaved stream of all LLRs of the \( k \)th user and performs an algorithm suited for an AWGN memoryless channel. For example, if the code is convolutional, this will be the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [3]. Since the channel seen by each user is not AWGN, the decoding algorithm is not optimal,
but most of the times this problem does not harm severely the decoder performance. Moreover, the channel is not even memoryless, because the interferers’ bits satisfy the code constraints, and so there is correlation among the bits of the same interferer. Interleavers can destroy the code constraints, and make the interferers’ coded bits “as independent as possible.” In Sections II–Section VI, we will clarify this concept. See the lower part of Fig. 1 for a picture of the iterative multiuser receiver.

III. THE ROLE OF INTERLEAVERS

To show how the interleaver design affects the receiver performance, we can collect the MF output given by (1) in a $K \times n$ matrix, where $n$ is the length of the used channel code $C$

$$Y = RAB + N,$$

where $Y = \{y_1, y_2, \ldots, y_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$, and $N = \{n_1, n_2, \ldots, n_n\}$. The $j$th row of $B$ is the codeword of user $j$ and can be considered as belonging to $C^j_M$, the modulated version of $C_j$, which is the equivalent code seen after the interleaver of user $j$. The set of values that $B$ can assume when its $j$th row, $j = 1, \ldots, K$, is taken from $C^j_M$, is a (matrix) linear code, which will be called $\tilde{C}$. $\tilde{C}$ has dimension $kK$ if $k$ is the dimension of $C$.

$N$ is a matrix of correlated Gaussian variables, which can be whitened by left-multiplying $Y$ by $(Q^H)^{-1}$, where $^H$ means transpose conjugate and $Q$ is the Cholesky matrix of $R$, i.e., $R = Q^HQ$. The output of this whitening filter is then

$$Y' = (Q^H)^{-1}Y = QAB + N_w,$$

where $N_w$ is now a matrix of independent Gaussian variables. No information is lost by performing this filtering. Suppose ML decoding is performed at the receiver. The word error probability can be written as

$$P(e) \leq \frac{1}{2^K} \sum_{B \in C} \sum_{B' \in C \text{ s.t. } B \neq B'} \Pr \{B \rightarrow B'\}$$

where $\Pr \{B \rightarrow B'\}$ is the pairwise error probability (PEP), i.e., the probability that the codeword $B'$ is more likely to be the transmitted word than $B$, the actual transmitted word. The PEP is the probability that $Y'$ is closer, in the Euclidean sense, to $QAB$ than to $QAB'$

$$\Pr \{B \rightarrow B'\} = \Pr \left\{ \|QAB - B'\|_F^2 < \|QAB - B'\|_F^2 \right\}
= \Pr \left\{ \|QAB - B'\|_F^2 + 2\text{Tr} \left( QAB - B' \right)^H \right\}
= \Pr \left\{ \|QAB - B'\|_F^2 + 2\text{Tr} \left( QAB - B' \right)N_w^H \right\}$$

where $\|\cdot\|_F$ and Tr$(\cdot)$ denote Frobenius norm and matrix trace, respectively. After a little bookkeeping, the following expression can be obtained:

$$\Pr \{B \rightarrow B'\} = Q \left( \frac{\sqrt{\text{Tr} \left\{ (B - B')^T A R A(B - B') \right\}}}{2\sigma} \right).$$

Substituting (8) into (6) gives the following upper bound on the word error probability:

$$P(e) \leq \frac{1}{2^K} \sum_{B \in C} \sum_{B' \in C \text{ s.t. } B \neq B'} \Pr \{B \rightarrow B'\}$$

where $\Delta B = B - B'$.

A. The Two-User Case

When there are only two users, the upper bound to the error probability in (9) simplifies to the following expression:

$$P(e) \leq \frac{1}{2^K} \sum_{b_i, b_i' \in C^M, (b_i, b_i') \neq (b_j, b_j')} Q \left( \frac{\sqrt{\sum_{i=1}^2 A_2^2 d_H^2 + 2A_2 A_1 \rho \Delta b_1^T \Delta b_2}}{\sigma} \right).$$

where $\rho$ is the correlation between the two users, $\Delta b_i = b_i - b_i'$ and $d_H^2 \triangleq d_H(b_i, b_i')$ is the number of wrongly decoded bits of user $i$. ($d_H$ denotes Hamming distance.)

For $\sigma \to 0$, the $Q$-function with the smallest argument prevails. It is then interesting to evaluate which term of the sum in (10) has the minimum argument, or, equivalently, to solve the following combinatorial minimization problem:

$$m^*(\alpha, \rho) = \min_{b_i, b_i' \in C^M, (b_i, b_i') \neq (b_j, b_j')} \left( d_H^2 + 2\alpha d_H^2 + 2\rho \frac{\Delta b_1^T \Delta b_2}{4} \right)$$

where $\alpha \triangleq A_2^2 / A_4$ is the normalized amplitude of user 2. Since the system is completely symmetric, there is no loss of generality in supposing $\alpha \geq 1$, i.e., user 2 is stronger than user 1. Notice also that the sign of $\rho$ does not matter since, by keeping $b_1$ and $b_1'$ fixed and exchanging the roles of $b_2$ and $b_2'$, the last term in the right-hand side (RHS) of (11) changes signs. Thus, the minimum above is equivalent to the following one:

$$m^*(\alpha, \rho) = \min_{b_i, b_i' \in C^M, (b_i, b_i') \neq (b_j, b_j')} \left( d_H^2 + 2\alpha d_H^2 - 2\rho \rho \right)$$

where, for the sake of brevity,

$$d_{12} \triangleq \frac{\|\Delta b_1^T \Delta b_2\|}{4}.$$

Now, if we consider the case in which user 2 is correctly decoded, i.e., $d_H^2 = 0$, we obtain an upper bound for $m^*(\alpha, \rho)$

$$m^*(\alpha, \rho) \leq d_{\text{min}}$$

where $d_{\text{min}}$ is the minimum distance of code $C$. This bound is valid for any interleaver. If both users are incorrectly decoded,
noticing that in this case \( d_{H_i}^{(i)} \geq d_{\min} \), \( i = 1, 2 \), and \( d_{12} \leq \min\{d_{H_1}^{(1)}, d_{H_2}^{(2)}\} \), we find this lower bound

\[
\min_{b_1, b_2 \in \mathbb{C}^{2^M} \setminus \{0\}, b_1 \neq b_2} \left( d_{H_1}^{(1)} + \alpha^2 d_{H_2}^{(2)} - 2\alpha |\beta| d_{12}\right)
\geq (1 + \alpha^2 - 2\alpha |\beta|) d_{\min}. \tag{14}
\]

Combining (13) and (14), we can conclude that, if \( \alpha \geq 2|\beta| \), \( m^*(\alpha, \rho) \) is equal to \( d_{\min} \) for any interleaver. Thus, there is no asymptotic gain in choosing the interleaver properly. For this reason, in the following we will concentrate on the case \( \alpha < 2|\beta| \), which also implies \( \alpha \leq 2 \).

To proceed further, we first notice that \( m^*(\alpha, \rho) \) is achieved by choosing \( b_1 \) as the (modulated) all-zero codeword, denoted here \( 0_M \), and one between \( b_2 \) and \( b_2^\prime \) also equal to \( 0_M \). Without loss of generality, we can suppose \( \rho < 0 \). Given \( b_1, b_1^\prime, b_2, b_2^\prime \), the value of \( d_{12} \) (apart from the sign) is given by \( I_1 - I_2 \), where \( I_1 \) is the number of positions in which the errors for user 1 are concordant to those of user 2, while \( I_2 \) is the number of positions in which the errors for user 1 are discordant to those of user 2. Now substitute the 4-tuple \((b_1, b_1^\prime, b_2, b_2^\prime)\) with the 4-tuple \((0_M, b_1 + b_1^\prime, 0_M, b_2 + b_2^\prime)\). While \( d_{H_1}^{(1)} \) and \( d_{H_2}^{(2)} \) remain the same as before, the value of \( d_{12} \) becomes now \( I_1 + I_2 \), since now the errors for the two users are all concordant. Thus, the second 4-tuple will give a smaller (or, at most, equal) value of the expression in (11) than the first 4-tuple.

When \( b_1 = 0_M \) and \( b_2 = 0_M \),

\[
d_{12} = \frac{w_1^{(1)} + w_2^{(2)} - d_H(b_1, b_2)}{2}
\]

so that we can also write the following expression:

\[
m^*(\alpha, \rho) = \min_{b_1 \in \mathbb{C}^{2^M} \setminus \{0_M\}, b_1 \neq 0_M} \left( (1 - \alpha|\beta|)w_1^{(1)} + (\alpha^2 - \alpha|\beta|)w_2^{(2)} + \alpha|\beta|\delta \right)
\tag{15}
\]

where \( w_1^{(1)} \triangleq d_H(b_1, 0_M) \) is the Hamming weight of \( b_1 \) and \( \delta \triangleq d_H(b_1, b_2) \).

Now, by considering the fact that \( w_1^{(1)} - \delta \leq w_2^{(2)} \leq w_1^{(1)} + \delta \), we can bound the function to minimize

\[
(1 + \alpha^2 - 2\alpha |\beta|) \left( w_1^{(1)} - \delta \right) + \delta
\leq (1 - \alpha|\beta|)w_1^{(1)} + (\alpha^2 - \alpha|\beta|)w_2^{(2)} + \alpha|\beta|\delta
\leq (1 + \alpha^2 - 2\alpha |\beta|)w_1^{(1)} + \alpha|\beta|\delta.
\tag{16}
\]

The above bounds show the important role played by \( 1 + \alpha^2 - 2\alpha |\beta| \), which ranges from 0 to 1 when \( 2|\beta| > \alpha \). We can distinguish two limit cases.

- The case \( 1 + \alpha^2 - 2\alpha |\beta| \rightarrow 0 \), i.e., \( (\alpha, |\beta|) \rightarrow (1, 1) \). In this case, the lower and upper bound both tend to \( \delta \), no matter what the value of \( w_1^{(1)} \) is. In this limit, \( m^*(\alpha, \rho) \) tends to the minimum value of \( \delta \) for which there are two code-words in \( C_1 \) and \( C_2 \) with Hamming distance \( \delta \) between each other.

- The case \( 1 + \alpha^2 - 2\alpha |\beta| \rightarrow 1 \), i.e., \( 2|\beta| \rightarrow \alpha \), for which instead the lower bound tends to \( w_1^{(1)} \) and \( m^*(\alpha, \rho) \) tends to \( d_{\min} \), which is the minimum for \( 2|\beta| \leq \alpha \).

Between these two extreme cases, there is a region in the plane \((\rho, \alpha)\) for which the value of \( \delta \) that gives the minimum is not exactly predictable.

It is clear from the above discussion that the interleaver design becomes important when the critical situation \( (\alpha, |\beta|) = (1, 1) \) is approached. When \( (\alpha, |\beta|) = (1, 1) \), if there are nonzero codewords in the intersection between the two equivalent codes seen after the interleavers, \( m^*(\alpha, \rho) = 0 \) and, as a consequence, there is an error floor in the high-SNR region. Moreover, the larger the intersection size, the higher the error floor. If \( 1 + \alpha^2 - 2\alpha |\beta| \) is small but not zero, instead, one should consider not only the size of the intersection, but also the minimum distance of the nonzero codewords in the intersection, if any. In Section III-A1, it will be shown that the intersection between two \((n, k)\) linear codes can be empty (except for the all-zero codeword) only if the code rate \( R_c = k/n \) is at most \( 1/2 \). Then, we come up with the following design rule

\textbf{Rule 1.} If \( R_c > 1/2 \), choose the interleavers \( \Pi_1 \) and \( \Pi_2 \) such that the intersection between \( C_1 \) and \( C_2 \) be minimized and the minimum Hamming distance of a nonzero codeword in \( C_1 \cap C_2 \) be maximized. If \( R_c \leq 1/2 \), choose the interleavers \( \Pi_1 \) and \( \Pi_2 \) such that the intersection between \( C_1 \) and \( C_2 \) only contains the all-zero codeword and the minimum Hamming distance between a nonzero codeword in \( C_1 \) and a nonzero codeword in \( C_2 \) is maximized.

\textbf{Example 1:} Consider the \((7, 4)\) code with the following generator matrix:

\[
G = \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}
\tag{17}
\]

In Figs. 2–4, the union bound to the word error probability (see (10)) for the \((7, 4)\) code with generator matrix (17) is depicted in three different cases. In all three cases, there are two users, the first transmits his coded bits without interleaving, while the second interleaves his coded bits. At the receiver, ML decoding is performed. For low SNR, the bounds can be greater than 1, because the union bound becomes loose. Asymptotically, Monte Carlo simulations are in agreement with these bounds.

Fig. 2 shows different interleaving strategies for the case \( (\alpha, |\beta|) = (1, 1) \). The uppermost curve shows the performance when no interleaving is performed on the second user. The other three curves correspond to interleaving strategies in which the ratio of the intersection size to the code size is \( 1/2 \), \( 1/4 \), and \( 1/8 \), respectively, from above. As it will be shown in the next section, the last curve corresponds to the minimum possible size of the intersection. All curves show an error floor, but this floor is higher, the higher is the intersection size. This is in agreement with the analysis above, since \( R_c = 4/7 > 1/2 \).

Fig. 3 shows different interleaving strategies for the case \( (\alpha, |\beta|) = (1, 0.8) \). The uppermost curve shows the performance when no interleaving is performed on the second user.

The
Fig. 2. Union bound on the performance of the \((7, 4)\) code in a two-user scenario with different interleavers. \(\rho = 1, \alpha = 1\).

Fig. 3. Union bound on the performance of the \((7, 4)\) code in a two-user scenario with different interleavers. \(\rho = 0.8, \alpha = 1\).
second curve from above corresponds to an interleaving strategy in which the ratio of the intersection size to the code size is 1/2, and the minimum distance of the intersection is 2, which is also the minimum distance of the code. The other two curves show the performance of two different interleaving strategies, which have minimum intersection size and differ for the minimum distance of the intersection. The lower has a minimum distance of 5, which is the maximum possible, while the upper has a minimum distance of 2. As it is evident, the upper curve is practically indistinguishable from the case with a larger intersection. Also, the three upmost curves appear to have the same slope. All these behaviors are accounted for by our analysis. Note that the best strategy gains more than 2 dB over the worst one at a word error rate of $10^{-3}$.

Fig. 4 shows different interleaving strategies for the case $(\alpha, |\rho|) = (1.5, 1)$. In this case, the $x$ axis shows the SNR for the weak user, i.e., the first one. The upmost curve shows the performance when no interleaving is performed on the second user. The other curves correspond to the same strategies of Fig. 3, with the same hierarchy of the performance. Note that the gain of the best strategy is more than 3 dB at a word error rate of $10^{-3}$.

Now consider the code that is the dual of the previous code, i.e., the code that has $G$ as its parity-check matrix. This code has rate $R_c = 3/7 < 1/2$. Figs. 5-7 show the union bound to the word error probability when this $(7,3)$ code is used.

Fig. 5 shows different interleaving strategies for the case $(\alpha, |\rho|) = (1, 1)$. The interleaving strategies correspond, in the same order, to those of Fig. 2. As it can be seen, there is an error floor for all strategies except the best one, which corresponds to an intersection containing the all-zero codeword only. The minimum Hamming distance between codewords in the two codes in this case is 1, which is also the largest possible value with these parameters.

Fig. 6 shows different interleaving strategies for the case $(\alpha, |\rho|) = (1.0, 8)$. The interleaving strategies correspond, in the same order, to those of Fig. 3. It is worth noting that, in this case, the size of the intersection is still important, for the difference between the two curves in the middle is quite appreciable. Note also that the two best curves exhibit the same slope. The gain for using the best strategy is more than 2 dB at a word error rate of $10^{-3}$.

Fig. 7 shows different interleaving strategies for the case $(\alpha, |\rho|) = (1.5, 1)$. The interleaving strategies correspond, in the same order, to those of Fig. 3. The behavior of the curves is similar to that depicted in Fig. 6. There is a gain of more than 3 dB at a word error rate of $10^{-3}$.

### B. The General $K$-User Case

When there are $K$ users, $K > 2$, as can be expected, things are more complicated. In fact, there are $K - 1$ normalized amplitudes and $K(K - 1)/2$ correlation coefficients, for a total of $(K - 1)(K + 2)/2$ free parameters, and an ad hoc interleaving
Fig. 5. Union bound on the performance of the $(7,3)$ code in a two-user scenario with different interleavers. $\rho = 1, \alpha = 1$.

Fig. 6. Union bound on the performance of the $(7,3)$ code in a two-user scenario with different interleavers. $\rho = 0.8, \alpha = 1$. 
strategy will depend on all of them. From (9), the minimization problem that should be solved can be formulated in the following way:

\[
m^*(\mathbf{A}, \mathbf{R}) = \min_{\{b_i, b'_i \in \mathcal{C}_i\}_{i=1}^K, (b_i, b'_i) \neq (b'_i, b_i)} \left( \sum_{j=1}^{K} A_{j}^T \mathbf{f}_{H} \right) + 2 \sum_{j=1}^{K} \sum_{j' = j+1}^{K} \rho_{j,j'} A_{j} A_{j'} \frac{\Delta b_j \cdot \Delta b_{j'}}{4} \right) \tag{18}
\]

where \(\rho_{j,j'}\) is the \((j,j')\) element of the correlation matrix \(\mathbf{R}\) and \(\Delta b_j = b_j - b'_j\). The above combinatorial minimization problem is intractable as it is formulated. However, we can have a hint to a sensible interleaver design rule by analyzing the function to minimize.

Suppose that we have found the following nonnegative quantity:

\[
f^*(\mathbf{A}, \mathbf{R}) = \min_{\mathbf{x} \in \{-1,0,1\}^K, \mathbf{x} \neq \mathbf{0}} \mathbf{x}^T \mathbf{A} \mathbf{R} \mathbf{A} \mathbf{x}. \tag{19}
\]

Also suppose that the vector \(\mathbf{x}^*\) that achieves this minimum is unique (apart from a change of sign) and let \(\mathcal{I} \subset \{1, \ldots, K\}\) be the set of indices for which \(\mathbf{x}^*\) is nonzero. If there is a nonzero codeword \(\mathbf{c}\) in the intersection among codes \(\mathcal{C}_{\mathcal{I}} = \mathcal{C}_{\bar{\mathcal{I}}}\), where \((j_1, \ldots, j_{|\mathcal{I}|}) = \mathcal{I}\) and \(|\mathcal{I}|\) is the cardinality of \(\mathcal{I}\), we can find a set of transmitted and received codewords that gives the following upper bound on \(m^*(\mathbf{A}, \mathbf{R})\):

\[
m^*(\mathbf{A}, \mathbf{R}) \leq f^*(\mathbf{A}, \mathbf{R}) w_H(\mathbf{c}). \tag{20}
\]

(Set \(\Delta b_i = 0\) when \(c_i = 0\) and \(\Delta b_i = \pm 2 c_i\) when \(c_i = 1\).)

The value of \(f^*\) depends only on \(\mathbf{A}\) and \(\mathbf{R}\), and not on the interleaver choice. The above bound is influenced by \(\mathbf{A}\) and \(\mathbf{R}\) only through the set \(\mathcal{I}\), the users erroneously decoded. To make interleaver design independent from the amplitude and correlation scenario, we come up with the following design rule, which is the extension to the \(K\)-user case of Rule 1 of the previous subsection.

**Rule 2.** Choose the interleavers such that for every subset of users \(\mathcal{I}\), with \(|\mathcal{I}| > 1\), the intersection among the codes of users in \(\mathcal{I}\) be minimized. If the intersection is not empty, maximize the minimum Hamming distance of the intersection. Otherwise, maximize the minimum Hamming distance of the linear closure of the codes of users in \(\mathcal{I}\).

More particularly, the size of the intersection must be minimized when \(f^*(\mathbf{A}, \mathbf{R}) = 0\). When \(f^*(\mathbf{A}, \mathbf{R}) > 0\), the weight of the codewords in the intersection becomes important (see (20)).

We remark that Rule 2 holds if the vector achieving \(f^*(\mathbf{A}, \mathbf{R})\) is unique. If, instead, more than one vectors give the minimum, which can happen when there is some symmetry in \(\mathbf{A}\) and in \(\mathbf{R}\), then it is more difficult to derive a design rule. A particular case in which this happens will be dealt with hereafter as an example.

---

Fig. 7. Union bound on the performance of the (7,3) code in a two-user scenario with different interleavers. \(\rho = 1, \alpha = 1.5\).
Example 2: Consider a case in which there are $K_1$ first-class users, $K_1 > 0$, with amplitude 1, and $K_2$ second-class users, $K_2 \geq 0$, with amplitude $\alpha > 1$. Also suppose that $\rho_{ij} = \rho$ for all user pairs. The minimization problem in (18) can be written as

$$m^*(\alpha, \rho) = \min \sum_{i=1}^{n} f_i(\alpha, \rho)$$

where

$$f_i(\alpha, \rho) = \frac{\rho}{4} \left( \sum_{j \in I_1} \Delta b_j[i] + \alpha \sum_{j \in I_2} \Delta b_j[i] \right)^2 + (1 - \rho) \left( |I_1| + \alpha^2 |I_2| \right)$$

$I_1$ and $I_2$ being the sets of users in the first and second class, respectively, that are erroneously decoded in the $i$th bit interval. The minimum of (22) can be found by distinguishing two cases, according to the sign of $\rho$.

If $\rho < 0$, the first term in (22) is negative, then its absolute value must be maximized. This is done by letting all terms in parenthesis be concordant, thus obtaining

$$f_i(\alpha, \rho) \geq \rho (|I_1| + \alpha |I_2|)^2 + (1 - \rho) \left( |I_1| + \alpha^2 |I_2| \right).$$

Analyzing the preceding expression as a function of $|I_1|$ and $|I_2|$, we find that the minimum is obtained either for $|I_1| = 1$ and $|I_2| = 0$, or for $|I_1| = K_1$ and $|I_2| = K_2$. Thus, a design rule targeted for this scenario would impose to minimize the size and maximize the minimum distance of the intersection among all users’ codes. Rule 2 above contains this one as a subcase.

If $\rho > 0$, instead, both terms in (22) are positive and must be minimized. As can be computed, the minimum is given for $|I_1| = 2$ and $|I_2| = 0$, which means that at every time instant only two users of the first class are incorrectly decoded, and with discordant errors. However, if $K_1 > 2$, the two users with decoding errors can change from time interval to time interval. Thus, we have the case cited earlier of many different vectors achieving the minimum, and finding a design rule becomes more involved. However, for the particular case of this example, we will show that Rule 2 still works well.

Consider the $(7, 3)$ code whose parity-check matrix is given in (17). In Figs. 8–10, we show the union bound in (9) for different values of the parameters $K_1$, $K_2$, $\rho$, $\alpha$, and different choice of the interleavers. In all cases, we have $K = 3$ users. User 1 is not interleaved.

In Fig. 8, the performance for $K_1 = 3$, $K_2 = 0$, and $\rho = 1$ is shown. From top downwards, we have the performance bound when no interleaving is performed (crosses), when user 2 is not interleaved while the interleaver of user 3 minimizes the intersection size with the code of user 1 (triangles down), when user 2 is interleaved such that half of the codewords are common to users 1 and 2 while the interleaver of user 3 is the same as in the previous case (circles), when interleavers are chosen such that every code pair has minimum-size intersection (squares),

4Here, we implicitly suppose that $K_1 \geq 2$. It is possible to generalize to the case $K_1 = 1$, but it does not add any new ingredient to this discussion.
finally, when interleavers are chosen such that the linear closure of the resulting codes has maximum size, according to Rule 2 (triangles up). The performance hierarchy agrees with the results of the above analysis. Notice also that all curves show an error floor, as it should be.

In Fig. 9, the performance for $K_1 = 3$, $K_2 = 0$, and $\rho = -0.5$ is depicted. The five curves shown correspond to the same interleaving strategies of Fig. 8. As can be seen, only the first exhibits an error floor, while the two best strategies have exactly the same performance. This means that for this case, the minimization of the pairwise intersections is a good design rule.

In Fig. 10, finally, we show the performance for $K_1 = 2$, $K_2 = 1$, $\rho = 1$, and $\alpha = 1.5$. The strong user is the third one. The five curves shown correspond to the same interleaving strategies of Fig. 8. Practically, in this case, the performance depends mostly on user 1 and user 2 codes. The two worst curves, both corresponding to a scenario in which users 1 and 2 have the same code, perform asymptotically the same. The difference between the two best curves, which is present also if user 3 is removed, can be imputed to higher order effects.

IV. THEORY OF INTERLEAVERS FOR BINARY LINEAR BLOCK CODES

In the previous section, we have given design rules for the channel codes assigned to the $K$ synchronous users of a CDMA multiuser system. These design rules involve minimization of the size and maximization of the minimum distance of the intersection between any subset of the codes seen after the interleave. In this section, we will give some theoretical results about the size of the intersection of two equivalent codes. To do this, we define the concept of $\mathcal{C}$-optimality that will be extended to $K \geq 2$ users to give a simpler rule than Rule 2 in the previous section. We will not give any results about the minimum distance of the intersection.

Let us consider a binary linear block code $\mathcal{C}$ with rate $R_c = k/n$. Let $G$ and $H$ be its generator and parity-check matrix, respectively. Consider interleavers acting on codewords, defined by permutations of the integers $\{1, \ldots, n\}$. Formally, the interleaver of the $j$th user $\Pi_j$ permutes the input codeword according to the bijective correspondence $\pi_j : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$. Notice that upper and lower case Greek letters are used for interleavers and for the permutations that define them, respectively. If there are only two users, numbered user 1 and 2, they will see, after their interleavers, two equivalent codes, which we denote, with a slight abuse of notation, $\Pi_1(\mathcal{C})$ and $\Pi_2(\mathcal{C})$. Since we are interested in a mutual property, we can suppose without loss of generality that user 1 has the identity interleaver, defined by $\pi_1(i) = i$, so that $\Pi_1(\mathcal{C}) = \mathcal{C}$. We want to study the properties of $\Pi_2 = \Pi$ within this assumption.

Let us define $\mathcal{C}' \triangleq \Pi(\mathcal{C}) \cap \mathcal{C}$, i.e., the intersection between the equivalent codes seen after the interleavers. It is easy to show that $\mathcal{C}'$ is a linear subcode of $\mathcal{C}$. Following the considerations of Section III, we define the concept of $\mathcal{C}$-optimal interleaver in the following manner.

**Definition 1:** The interleaver $\Pi$ is $\mathcal{C}$-optimal if and only if

$$\dim(\mathcal{C}') = k_0 \triangleq k - \min\{k, n-k\}. \quad (24)$$
The above definition is in agreement with the conclusions of Section III. In fact, if $\Pi(G)$ is the matrix obtained from $G$ by permuting its columns according to $\pi$, the product $P \triangleq \Pi(G)H^T$ will have a rank equal to $k - \dim(C')$. Keeping in mind that $P$ has $k$ rows and $n-k$ columns, the maximum value of its rank is $\min\{k, n-k\}$. Then, if $\Pi$ is $C$-optimal, the intersection between $C$ and $\Pi(C)$ is actually the minimum possible.

Definition 1 can be justified also from another perspective, which fits the consideration that interleavers in a multiuser receiver should make the interferers’ bits “as memoriless as possible.” Let us call $B^n$ the space of the $n$-length binary vectors. Each of the $2^n$ elements in $B^n$ will have the same probability of being output by a binary memoriless source (BMS). Then, the probability of belonging to $C$, for an $n$-dimensional word output by a BMS, equals $2^{k-n}$. If $k \geq n-k$ and $\Pi$ is $C$-optimal, the fraction of codewords in $\Pi(C)$ that are also in $C$ is $2^{k-n-k} = 2^{k-n}$, by definition. So, $\Pi(C)$ is a fair sampling of $B^n$ with respect to the belonging to $C$. Fig. 11 shows graphically this concept. If $k \leq n-k$, then $C'$ contains only the all-zero codeword, which is always present in a linear code and obviously resistant to all permutations. (We say that a codeword $c \in C$ is resistant to $\Pi$ iff $\Pi(c) \in C$.)

The following proposition relates an interleaver with its inverse.

Proposition 1: A given interleaver $\Pi$ has the same effect on $C'$ as its inverse $\Pi^{-1}$. Precisely, if $C' \triangleq \Pi^{-1}(C) \cap C$, then $\dim(C') = \dim(C')$.

Proof: $\Pi(C') = \Pi((\Pi^{-1}(C) \cap C) = C \cap \Pi(C) = C'$. □

Then the dimension of $(C')^\perp$ can also be written as $2(n-k) - \dim((C')^\perp)$. Comparing the two expressions, we obtain $\dim((C')^\perp) = n - k - m$. □

From the last proposition, we deduce that, if $\Pi$ is $C$-optimal, at least one between $C$ and $C^\perp$, and precisely the one with the least dimension, has only one codeword resistant to $\Pi$. This only codeword is obviously the all-zero word. So, we can conclude that not all codes have optimal interleavers. In fact, the codes that have $R_c \leq 1/2$ and possess the all-one word, together with
their dual codes, do not have optimal permutations, because also the all-one word is resistant to every interleaver. At the moment, we do not know if this is the only case of codes without optimal interleavers.

Now, let us consider the role of the automorphism group of $C$, which we will call $U_C$ (from “useless”), in our problem.

**Proposition 3:** Permutations belonging to the same left (right) coset of $U_C$ have the same effect on the code.

**Proof:** Let $\pi'$ be a permutation belonging to the same left coset of $U_C$ as $\pi$. Then, there will be $\pi_u \in U_C$ such that $\pi' = \pi \pi_u$. $C$-useless interleavers transform $C$ into itself, so, except for a reordering, $\Pi$ and $\Pi'$ have the same effect. Now let $\pi'$ be a permutation belonging to the same right coset of $U_C$ as $\pi$. Then, there exists $\pi_u \in U_C$ such that $\pi' = \pi \pi_u$. Let $\pi_u(C') \cap C = \emptyset$. $\Pi_u$ maps elements in $C$ into elements in $C$ and elements out of $C$ into elements out of $C$, so $\Pi_u(C') \cap C = \emptyset$. Then $\Pi$ and $\Pi'$ have the same effect on $C$. \hfill \square

Thanks to the last proposition, we can talk of $C$-optimal left (right) cosets of $U_C$, instead of $C$-optimal interleavers.

Now we generalize to $K$ users the concept of $C$-optimality.

**Definition 2:** A set of $K$ interleavers $\Pi_1, \Pi_2, \ldots, \Pi_K$ is called $C$-optimal if the interleavers are pairwise $C$-optimal, that is, if $\Pi_i^{-1} \Pi_j$ is $C$-optimal for every choice of $i$ and $j$ in $\{1, \ldots, K\}$, $i \neq j$.

The reason for this definition is that user $i$ sees the coded bits of user $j$ through the compound interleaver $\Pi_i^{-1} \Pi_j$. Thanks to Proposition 1, only interleavers $\Pi_i^{-1} \Pi_j$ with $i < j$, in number of $K(K - 1)/2$, have to be considered. Notice that $C$-optimality is strictly contained in Rule 2 of Section III, because Rule 2 forces us to look at every subset of users, not only pairs.

In general, we do not have any recipe to construct deterministically the $C$-optimal interleaver set. However, we do have a recipe in the particular subset of congruential interleavers, as explained in Section IV-A.

### A. Interleavers for Convolutional Codes

There are several ways to obtain a block code from an $(n_0, k_0, N)$ convolutional code $C$. For example, if all paths in the trellis start and end in the zero state, we obtain a block code $C_{\text{ZT}}$, called zero-termination convolutional code, with parameters $n = n_0 L$ and $k = k_0 L - \nu$, where $L$ is the number of trellis steps from zero state to zero state and $\nu = (N - 1)k_0$ is the memory. Another possibility is to make paths starting and ending in the same state, thus obtaining a block code $C_{\text{TBD}}$, called tail-biting convolutional code, with parameters $n = n_0 L$ and $k = k_0 L$. The theory of the previous section, then, holds also for a zero-termination or tail-biting convolutional codes.

For convolutional codes, it is relevant to consider symbol interleavers, i.e., interleavers that take the $n_0$ bits output in a single trellis step as a symbol drawn from GF$(2^{n_0})$ and do permutations at the symbol level. The following lemma will be useful afterwards.

---

5As usual in the literature [6], $k_0$ and $n_0$ are the number of input bits and of output bits, respectively, at each trellis step. $N$ is the constraint length.
of indices through a $C$-useless permutation is still an $N$-length run of indices.

The above proposition has the following corollary, whose proof is immediate.

**Corollary 5:** For a convolutional code $C$, $U_C$ can contain only two symbol interleavers: the identity permutation (always) and the symbol time-reversal permutation, defined by

$$\Pi_{TR}(i) = L - i + 1, \quad i = 1, \ldots, L. \quad (27)$$

This second permutation will belong to $U_C$ only if the code is a symbol time-reversal.

In the remainder of this section, we will prove a theorem which provides a method of constructing a set of $K$ $C$-optimal interleavers, but, for this purpose, it is necessary to reduce the scope of the analysis. First, we concentrate on symbol interleavers. Since a frame contains $L$ symbols, symbol interleavers are permutations acting on the set of integers $\{0, \ldots, L - 1\}$. Moreover, we limit our analysis to the particular set of congruential permutations [4], defined as

$$\pi_g(i) = ig \mod L \equiv (ig)_L \quad (28)$$

where $g$ is an integer and $g$ and $L$ are relatively prime. This particular set of interleavers is simple to treat and has also the good property that it can be easily implemented and stored, since it depends only on the parameter $g$.

The main result is the following.

**Theorem 6:** Consider an $(n_0, k_0, N)$ zero-termination convolutional code $C_{ZT}$, with $k_0 = 1$ and $n_0 \geq 2$, and symbol interleavers. Denote with $N'$ the constraint length of the dual code. If the frame length $L$ satisfies the following conditions:

1. **Condition 1.** $L > NN'$, and
2. **Condition 2.** $\sum_{i=0}^{L-1} z^i = \frac{1 + z^L}{1 + z}$ is an irreducible polynomial over GF (2),

then $\Pi_g$, defined by the permutation in (28) is $C_{ZT}$-optimal for $1 < g < L - 1$.

Before proving Theorem 6, we remark that the values of $L$ that satisfy Condition 2 of Theorem 6 are prime. In fact, if $L = L_1 L_2$, with $L_1 > 1$, $L_2 > 1$, then

$$\sum_{i=0}^{L-1} z^i = \left( \sum_{i=0}^{L_1-1} z^i \right) \left( \sum_{i=0}^{L_2-1} z^{iL_1} \right).$$

Then, for those values of $L$, every $1 \leq g < L$, defines an interleaver. Also, it is known [15, Ch. 7, Sec. 5] that, for a given $L$, if the values

$$\{(2^i)_L, i = 0, \ldots, L - 2\} \quad (29)$$

are all distinct, then Condition 2 is satisfied by $L$. So it is easy to find the “good” values of $L$, as defined in Theorem 6.

**Proof:** Let the generator matrix of the corresponding tail-biting convolutional code $C_{TB}$ be as in (30) (see the bottom of the page), where the elements $m_i$ are drawn from GF $(2^{n_0})$. It has $l_0 L$ rows and $L$ columns. Its parity-check matrix $H_{TB}$ can be written as in (31) (also at the bottom of the page), where the elements $p_i = (p_{i0}, \ldots, p_{i, n_0 - k_0})^T$ are drawn from GF $(2^{n_0})^{(n_0 - k_0)}$. It has $(n_0 - k_0) L$ rows and $L$ columns.

We can write the columns of $H_{TB}$ in a polynomial form, by multiplying the element in the $i$th row, $i = 0, \ldots, (n_0 - k_0) L - 1$, by $D^i$, $D$ being a dummy variable. So the $j$th column of $H$, $j = 0, \ldots, L - 1$, can be written

$$h_j(D) = \left( \sum_{i=0}^{N'-1} p_i(D) z^{j+N'-i-1} \right)_{1+Z}. \quad (32)$$

where $p_i(D) = \sum_{i=0}^{n_0 - k_0 - 1} p_{i, i} D^i$ and $Z \triangleq D^{n_0 - k_0}$.

$$G_{TB} = \begin{pmatrix}
    m_{N-1} & m_0 & m_1 & \cdots & m_{N-2} \\
    m_{N-2} & m_{N-1} & \vdots & \ddots & \vdots \\
    \vdots & \ddots & \ddots & \ddots & \vdots \\
    m_1 & m_2 & \cdots & m_{N-1} & m_0 \\
    m_0 & m_1 & \cdots & m_{N-1} & m_{N-2} \\
    \vdots & \ddots & \ddots & \ddots & \vdots \\
    m_0 & m_1 & \cdots & m_{N-1} & m_0 \\
    \vdots & \ddots & \ddots & \ddots & \vdots \\
    m_0 & m_1 & \cdots & m_{N-1} & m_0 \\
\end{pmatrix} \quad (30)$$

$$H_{TB} = \begin{pmatrix}
p_{N'-1} & p_0 & p_1 & \cdots & p_{N'-2} \\
p_{N'-2} & p_{N'-1} & \vdots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
p_1 & p_2 & \cdots & p_{N'-1} & p_0 \\
p_0 & p_1 & \cdots & p_{N'-1} & p_{N'-1} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
p_0 & p_1 & \cdots & p_{N'-1} & p_{N'-1} \\
\end{pmatrix} \quad (31)$$
Define, for a given $i$

$$q_i(D) = m_i \cdot h_0(D) = \left( \sum_{l=0}^{N-1} m_i \cdot p_l(D) Z^{N-1-l} \right)_{1+ZL}$$

(33)

where $m_i \cdot p_l(D) = \sum_j m_{ij} p_{ij}^l D^l$, $m_i$, and $p_{ij}$ being the representations of $m_i$ and $p_{ij}$ as binary vectors, is a polynomial with binary coefficients.

By straightforward computation, one can show that the $j$th row of the product

$$P_{TB} = \Pi_g(G_{TB}) \cdot H_{TB}^T, \quad 0 \leq j < k_0 L$$

can be written in polynomial form as

$$r_j(D) = \left( \sum_{l=0}^{N-1} Z^{((j+1)g)_l} q_i(D) \right)_{1+ZL}$$

(34)

In the following, conditions on the rank of $P_{TB}$ are derived.

There are no more rows than columns, so that, if $P_{TB}$ is not full rank, there exists a linear dependence between rows. Every row of $P_{TB}$ is the cyclic shift of the first row $r_0(D)$

$$r_j(D) = (Z^{(jg)_l} r_0(D))_{1+ZL}$$

Thus, if the rows of $P_{TB}$ are linearly dependent, there exists a binary polynomial $g(Z)$, with maximum degree $L-1$, such that

$$(g(Z) r_0(D))_{1+ZL} = 0$$

(35)

or, in equivalent form

$$g(Z) r_0(D) = c(D)(1 + Z^L).$$

(36)

Now, if $p(Z) \equiv \sum_{l=0}^{L-1} Z^l$, a factor of $1 + Z^L$ is irreducible, then it must divide either $g(Z)$ or $r_0(D)$. The maximum degree of $g(Z)$ is $L - 1$, so, if $p(Z)$ divides $g(Z)$, then it must be $g(Z) = p(Z)$. Actually, we know that the sum of all rows of $G_{TB}$ in (30) is resistant to all symbol permutations because it is composed of all equal symbols. Thus, the last row of $P_{TB}$ is equal to the sum of the other rows.

We would like to know under which condition the first $k_0 L - 1$ of $P_{TB}$ rows are linearly independent. Suppose they are dependent. Then, $p(Z)$ must be a factor of $r_0(D)$. From (34) it can be seen that $r_0(D)$ is composed by a sum of $N$ polynomials, each of length $(n_0 - k_0) N'$, a multiple of $n_0 - k_0$, and starting from an offset which is a multiple of $n_0 - k_0$. Then, dividing the columns of $P_{TB}$ in strips of $n_0 - k_0$, there are at most $NN'$ such strips covered by $r_0(D)$ (the maximum is reached if the $N$ polynomials do not overlap). There are $L$ strips in $P_{TB}$, so if $L > NN'$ (Condition 1 of the theorem), at least one is entirely composed of zeros. If $r_0(D)$ is divisible by $p(Z)$, then we have

$$r_0(D) = \beta(D) p(Z)$$

(37)

where $\deg(p(Z)) < n_0 - k_0$, because $\deg(r_0(D)) < (n_0 - k_0)L$. Then, $r_0(D)$ can be viewed as the periodic repetition of $\beta(D)$, every $n_0 - k_0$ columns. If Condition 1 is met, at least one of these repetitions is zero, and then $\beta(D) = 0$. Thus, we conclude that $P_{TB} = 0$.

V. INTERLEAVER SUITABILITY TO ITERATIVE DECODING

As mentioned in the Introduction, the role of the interleavers is two-fold. Interleavers change the constellation of possible transmitted signal, so they can influence the distance spectrum (see Section III). But they also have a strong impact on the performance of iterative decoding. In this section, we will analyze interleaver characteristics from this point of view, when convolutional codes are used. The basis of the following analysis is a paper by Hokfelt et al. [12].

Let us consider the first iteration of an iterative multiuser receiver for two users. The SISO-US outputs LLRs for both users, which are correlated with each other. In the synchronous case, correlations are different from zero only between observables in the same bit interval. If LLRs are extracted from $\hat{y}$, see (2)), there will be correlation between $LLR_{1}[j]$, $LLR_{2}[j]$, $\bar{y}_1[j]$, and $\bar{y}_2[j]$, where $LLR_{i}[j]$ is the $j$th LLR of the $i$th user and $\bar{y}_i[j]$ is the $j$th component of $\hat{y}$ in the $j$th bit interval. The amount of correlation depends on the users’ amplitudes and residual correlations after the SISO-US.

The LLRs are then deinterleaved according to both users’ interleavers. If $LLR_{j}[\cdot]$ represents the deinterleaved LLR for user $j$, we can write

$$\begin{bmatrix}
LLR_{1}[1] \\
LLR_{2}[1]
\end{bmatrix} = \begin{bmatrix}
\llbracket LLR_{1}[\pi_1^{-1}(y_1)] \\
\llbracket LLR_{2}[\pi_2^{-1}(y_2)]
\end{bmatrix} = \begin{bmatrix}
LLR_{1}[\pi_1(1)] & \cdots & LLR_{1}[\pi_1(n)] \\
LLR_{2}[\pi_2(1)] & \cdots & LLR_{2}[\pi_2(n)]
\end{bmatrix}.$$  (38)

Then, there is correlation between $LLR_{1}[\pi_1^{-1}(y_1)]$, $LLR_{2}[\pi_2^{-1}(y_2)]$, $\bar{y}_1[j]$, and $\bar{y}_2[j]$. The deinterleaved LLRs enter the SISO channel decoders. According to [12], the autocorrelation between the decoder output and the cross correlation

$$is divisible by$$

$h_0(D)$, then we have $$h_0(D). The deinterleaved LLRs enter$$

strips in$$

$s$ of $i_f$ LLR or divides $h_0$, then it must be 8 in the$$

optimal interleavers. The$$

resistant to$$

and$$

Thus, for$$

rows are linearly independent. Suppose they are depen$$

has then dimension$$

LLR$$

is$$

is composed of all equal symbols. Thus, the last row of$$

dent. Then,$$

such strips covered by$$

can be written in polynomial form as$$

TARABLE et al. [12].

Authorized licensed use limited to: Politecnico di Torino. Downloaded on March 12, 2010 at 12:54:37 EST from IEEE Xplore. Restrictions apply.
between input and output can be both suitably modeled by exponential function. If we denote by $\overrightarrow{LLR}_j[i]$ the output of the $j$th decoder (in the order it is output), we will have an input–output cross correlation for user 1

$$E \left[ \overrightarrow{LLR}_1[i] \overrightarrow{LLR}_1[i'] \right] = ae^{-\|i-i'\|}, \quad i \neq i'$$

(39)

where $a$ and $c$ are constants depending on the considered code. Because of the correlation between $\overrightarrow{LLR}_1(\pi^{-1}_1(i))$ and $\overrightarrow{LLR}_2(\pi^{-1}_2(i))$, there is also a correlation between the output of the first decoder and the input of the second decoder

$$E \left[ \overrightarrow{LLR}_1[i] \overrightarrow{LLR}_2[i'] \right] = a'e^{-\|i-i'\|}, \quad i \neq i'$$

(40)

where $a'$ is a constant. This, in turn, causes the output of the first and second decoder to be correlated

$$E \left[ \overrightarrow{LLR}_1[i] \overrightarrow{LLR}_2[i'] \right] = \sum_{i''} a'' e^{-\|i-i''\|} \overrightarrow{LLR}_1[i''], \quad i \neq i'$$

(41)

$a''$ being a constant. An analogous correlation exists between $y_1[i]$ and $\overrightarrow{LLR}_2[i]$. After interleaving, the a priori information, which we denote $\overrightarrow{LLR}_2[i]$, enters the SISO-US. The algorithm performed by the SISO-US assumes that $\overrightarrow{LLR}_2[i]$ is independent from $y_1[i]$ and from $\overrightarrow{LLR}_1[i]$. According to (41), this suggests that, whenever $\|i-i''\|$ is small, $\pi^{-1}_1 \pi^{-1}_2(i) - \pi^{-1}_1 \pi^{-1}_2(i'')$ should be kept high, i.e., the permutation $\pi^{-1}_1 \pi^{-1}_2$ should have good triangular spread properties [8].

The generalization to $K$ users is straightforward: permutation $\pi^{-1}_j \pi^{-1}_j$, for every $j$ and $j', j \neq j'$, must have good triangular spread properties.

To quantify the goodness of an interleaver pair with respect to (w.r.t.) their use in an iterative multiuser receiver, we define an iterative decoding suitability measure, which will be called hereafter IDS, defined as the empirical variance of the decoder output correlation. We then have, for a given interleaver pair

$$IDS \triangleq \sum_{i, j=1}^{n} \left( E \left[ \overrightarrow{LLR}_1[i] \overrightarrow{LLR}_2[i'] \right] - \sum_{i', j'=1}^{n} E \left[ \overrightarrow{LLR}_1[i'] \overrightarrow{LLR}_2[i''] \right] \right)^2.$$  

(42)

The smaller the IDS, the better the interleaver pair, because the correlation between LLRs is more evenly distributed across the block length. To compute the IDS, we use the approximated analytical expression given in (41).

An example of a (symbol) interleaver set that has a low value of the IDS but does not satisfy Rule 1 in Section III is given by the pair $(\Pi_1, \Pi_2)$, where $\Pi_1$ is defined by the identity permutation with length $L = 128$, while $\Pi_2$ is defined by the congruential permutation with $g = 113$. This set has good spread properties, but the intersection between the two equivalent codes has dimension 12, while the minimum dimension is 9. However, the simulations in Section VI will show that this interleaver set does give good performance.

The extension of the IDS parameter to sets of more than two interleavers is still open.

VI. RESULTS OF SIMULATIONS

In this section, we give simulation results to assess the validity of the design rules given in the previous sections. In all cases, we consider terminated convolutional codes as channel codes. Iterative multiuser receivers are employed at the receiver to estimate the information bits.

In the performed simulations, interleavers are either congruential, according to Theorem 6, or pairwise S-random [9]. To generate a suitable set of $K$ S-random $C$-optimal interleavers, we can first generate a set of $S$-random interleavers, and then “adjust” them to fit the conditions of $C$-optimality. Although there may be more sophisticated algorithms, the “adjustment” consists of a series of transpositions that do not spoil the spread characteristics of the interleaver set. The comparison between an interleaver pair $\Pi_1$ and $\Pi_2$ is performed by concatenating $\Pi_1(G)$ and $\Pi_2(G)$

$$\left( \begin{array}{c} \Pi_1(G) \\ \Pi_2(G) \end{array} \right)$$

(43)

and computing the rank $\rho$ of the resulting $2k \times n$ matrix. The dimension of the intersection between the two equivalent codes is given by $2k - \rho$. This computation has a complexity of about $O(k^2 n)$. In general, there are “many” $C$-optimal interleavers, so it does not take long time to reach the end of the search. Indeed, when $n$ is high enough, as for a convolutional code, $K$ random interleavers, for reasonable values of $K$, are with high probability $C$-optimal.

Fig. 13 shows a two-user system in which the code is the rate-1/2, 16-state code with generators $23_{85}, 35_{88}$. The block length is equal to 128 and the first user is not interleaved. The receiver is the iterative receiver described in [19] with switching iteration set to zero. The users have the same energy and correlation. We see that, as the iterations proceed, the gap between the noninterleaved system and the interleaved system increases significantly. The $C$-optimal interleaver set has good spread properties. It gains about 1 dB at BER = $10^{-2}$ over the simulated random interleaver set. The receiver in the system with $C$-optimal interleaving converges to single-user performance for $E_b/N_0$ equal to about 4.5 dB.

Fig. 14 shows a symmetric four-user system with the same code and the same receiver as in the previous simulation. Every pair of users has correlation 0,7. Again, the iterations enhance the gap between the two extreme cases. In this case, interleavers have been designed to be $C$-optimal. Concerning the spread properties, S-random permutations have been chosen for $\pi_2, \pi_3$, and $\pi_4$, while user 1 is not interleaved. No condition has been imposed on the spread properties of $\pi_1^{-1} \pi_2, \pi_1^{-1} \pi_3, \pi_1^{-1} \pi_4$, and $\pi_1^{-1} \pi_1$. The performance of user 1 is reported.

Fig. 15 shows the different performance at the tenth iteration, obtained by choosing different congruential interleavers with the code $23_{85}, 35_{88}$. Here we have the same system as in Fig. 13. The purpose of this simulation is to compare the importance of the two criteria, the $C$-optimality and the spread properties w.r.t. each other. The IDSs computed for these interleavers are based on (42), with parameters $a = 1$ and $c = 0,1$ experimentally found. The five curves shown are as follows:

- $L = 128, g = 63, C$-optimal, IDS = $1,6989$;
Fig. 13. Iterative receiver with different interleaving strategies: two-users, code (21g, 35g), correlation 1.

• $L = 128$, $g = 65$, not C-optimal, IDS = 1.0659;
• $L = 128$, $g = 111$, C-optimal, IDS = 0.1306;
• $L = 128$, $g = 113$, not C-optimal, IDS = 0.1313;
• $L = 107$, $g = 63$, C-optimal since $L = 107$ satisfies the conditions in Theorem 6, IDS = 0.1820.

As can be seen, the spread properties seem to have much more influence on the system performance than the C-optimality for relatively small values of the block length. We conjecture that this behavior is even more evident when the block length increases. Thus, for practical systems, we conclude that the spread
characteristics of the interleaver set is likely to have a greater impact on the iterative multiuser receiver performance.

VII. CONCLUSION

In this paper, we have introduced several criteria of goodness of a set of interleavers for a turbo-like multiuser transceiver. The definitions we have introduced are based on empirical considerations and no optimality with respect to the bit-error rate (BER) or other performance measures is claimed. The criteria found, which have a reasonable complexity, are derived by considering both the ML performance and the iterative decoding suitability point of view. Simulations performance show that, especially when the block length is typical of convolutional codes, the latter approach seems to be more important than the former one.

REFERENCES