

# Iterative Multiuser Joint Decoding: Unified Framework and Asymptotic Analysis

Joseph Boutros<sup>†</sup> and Giuseppe Caire<sup>\*</sup>

August 21, 2000

<sup>†</sup> Ecole Nationale Supérieure des Télécommunications de Paris,  
64 rue Barrault – 75634 Paris CEDEX, France.

Email: [boutros@com.enst.fr](mailto:boutros@com.enst.fr).

<sup>\*</sup> Institut Eurecom,  
2229 Route des Crêtes, B.P. 193  
06904 Sophia-Antipolis CEDEX, France.

Email: [caire@eurecom.fr](mailto:caire@eurecom.fr)

## Abstract

We present a framework for iterative multiuser joint decoding based on the application of the sum-product algorithm to the factor graph representation of the *a posteriori* joint probability mass function of the users information bits. Several low-complexity algorithms previously proposed based on parallel and serial hard and soft interference cancellation are derived in a simple and unified way. A wide class of these algorithms is analyzed by using the approach of *density evolution on graphs* combined with results from the theory of *large random matrices*. In the case of parallel interference cancellation and equal power users we present a simple approximated analysis able to explain qualitatively (and often quantitatively) the behavior of iterative decoding via the study of the stable fixed points of a one-dimensional non-linear dynamical system.

**Keywords:** Multiuser detection, interference cancellation, iterative decoding on graphical models.

# 1 Introduction

Multuser detection has been traditionally regarded as an ensemble of techniques to detect uncoded data in a multiple-access waveform channel (see [49] and references therein).

More recently, research has been focused on the combination and interaction of channel coding and multuser detection. From an information-theoretic point of view, all points in the capacity region of the Gaussian multiple-access channel are achievable by successive single-user decoding and interference cancellation (stripping) [16, 7]. This generalizes to the correlated-waveform channel (e.g., CDMA) as shown in [48], where the optimal stripping decoder has the structure of a decision-feedback minimum mean-square error detector where users are decoded in sequence and at each stage the decoded and re-encoded user signals are subtracted from the received signal. The key to the optimality of stripping is the use of capacity-achieving codes of rate arbitrarily close (but not larger) than the capacity of the channel obtained by removing the already decoded users. In this way, optimal spectral efficiency is achieved by simple single-user coding and decoding, with linear complexity in the number of users.

A different and perhaps more practical approach to multuser detection and decoding considers a given class of finite-complexity channel codes and investigates the achievable spectral efficiency at given target bit-error rate (BER). The number of works in this direction is overwhelming. Without the ambition of being exhaustive, we refer to [1, 2, 8, 9, 10, 14, 15, 17, 24, 26, 33, 34, 35, 41, 44, 46, 53] and references therein. In these works, joint decoding and its low-complexity iterative approximations based on *interference cancellation* (IC) have been investigated, several different algorithms have been proposed and performance has been evaluated mainly via computer simulation.

This paper contributes to the above stream of work in two ways. First, we provide a unified framework where a large class of previously proposed algorithms can be elegantly derived. Second, we provide an asymptotic performance analysis of IC-based iterative decoding enabling the quantitative and qualitative evaluation of systems for which simulation would be just impossible.

Our framework is based on the application of the *sum-product algorithm* [21] to the factor (or dependency) graph representation [45, 38, 54, 32, 21] of the *a posteriori* joint probability mass function (pmf) of the users information bits. The factor graph for the problem at hand has cycles if the number of users is larger than 1. Therefore, the resulting algorithms are intrinsically iterative. Depending on the algorithm execution scheduling we obtain classical parallel and serial iterative decoding as special cases. By making some simple approximations of the *a posteriori* pmf at the decoders output, we obtain in a direct

way several previously proposed IC-based algorithms, which were motivated mainly by heuristics (see Remark 4 of Section 3.3).

We discuss in the details some effects limiting the performance of IC-based algorithms, namely: *bias* and *mismatch*. We show that the residual interference after IC is conditionally biased if the interfering symbol estimates are obtained from the decoder *a posteriori probability* (APP) of the encoded symbols, while it is asymptotically unbiased (for large interleaver size) if these estimates are obtained from the decoder “extrinsic information”. Remarkably, several algorithms derived from heuristics [1, 9, 33, 14, 46] make use of APPs, while the straightforward application of the sum-product algorithm implies the use of extrinsic information. We show also that the *soft-in soft-out* (SISO) decoders [4] suffer from mismatch, since the variance of residual interference “assumed” by the decoders at any given iteration does not coincide in general with the true variance.

Our performance analysis is asymptotic in two ways: we consider infinite-size (ideal) interleaving and CDMA in the *large-system regime*, i.e., when the number of users and the spreading gain go to infinity while their ratio stays constant [47, 51]. The analysis holds (subject to mild conditions) because the IC-based decoders derived from the sum-product algorithm provide asymptotically unbiased residual interference at each iteration and because we are able to take the SISO mismatch into account. As a byproduct, we provide a formula for the large-system regime *asymptotic multiuser efficiency* (AME) [49] of a *linear minimum mean-square error* (LMMSE) detector with mismatch of the user received powers, which has interest in its own.

In general, all the algorithms considered in this paper can be analyzed via the probability *density evolution* (DE) of the coded symbols marginal pmf (equivalently, of symbols likelihood-ratio) on the tree representing the local neighborhood of each encoded symbol [42, 43, 52, 6, 20]. In the special case of parallel IC and equal-power users we provide an approximated analysis describing the behavior of the iterative joint decoder by a one-dimensional non-linear dynamical system. We show that this system has one or two stable fixed points for channel load (number of users per chip) smaller or larger than a threshold value. Under fairly general conditions, the single-user bound is practically achieved if the system has a single stable fixed point while the BER is much larger than the single-user bound if the system has two stable fixed points. In this respect, we say that the iterative decoder has a *threshold behavior* with respect to the channel load.

We compute the spectral efficiency achievable by parallel IC with random spreading and equal-power users with simple binary convolutional codes and BPSK or QPSK modulation and we show that at target BER equal to  $10^{-5}$  and  $E_b/N_0$  ranging from 4 to 6 dB the achievable spectral efficiency is between 1.2 to 1.8 bit/s/Hz away from the optimal

spectral efficiency achievable with random spreading and Gaussian codes.

The paper is organized as follows. Section 2 presents the system model and the main assumptions. Section 3 deals with the factor graph and the sum-product algorithm and derives iterative joint decoding schemes. In Section 3.3 we obtain low-complexity IC-based decoders. Section 4 is dedicated to the asymptotic analysis of IC decoders. Results are presented in Section 5 and in Section 6 we point out some suggestions for future work. The proofs of the main propositions are given in Appendix A.

## 2 Synchronous CDMA system model

We consider a DS-CDMA system [39] with  $K$  users, spreading gain  $L$ , synchronous transmission and frequency non-selective propagation channels. The complex baseband equivalent received signal sampled at the chip rate is represented by the linear model [49, 51, 47]

$$\mathbf{y}[n] = \mathbf{S}\mathbf{W}\mathbf{a}[n] + \boldsymbol{\nu}[n] \quad (1)$$

where  $\mathbf{y}[n] \in \mathbb{C}^L$  is the vector of chip-rate samples obtained in the  $n$ -th symbol interval,  $\boldsymbol{\nu}[n] \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I})$  is the corresponding vector of white Gaussian noise samples,<sup>1</sup>  $\mathbf{a}[n] = (a_1[n], \dots, a_K[n])^T$  is the vector of users modulation symbols transmitted at time  $n$ ,  $\mathbf{S} \in \mathbb{C}^{L \times K}$  is the matrix containing the users spreading sequences by columns and  $\mathbf{W} = \text{diag}(w_1, \dots, w_K)$  is the diagonal matrix of the users complex amplitudes.

All users modulation symbols belong to the same unit-energy  $M$ -PSK signal set  $\mathcal{A}$  ( $|a| = 1$  for all  $a \in \mathcal{A}$ ). The users spreading sequences  $\mathbf{s}_k = (s_{k,1}, \dots, s_{k,L})^T$  (where  $\mathbf{s}_k$  is the  $k$ -th column of  $\mathbf{S}$ ) have binary antipodal components  $s_{k,\ell} \in \{\pm 1/\sqrt{L}\}$ . With these normalizations, the  $k$ -th user received SNR is given by  $\gamma_k = |w_k|^2$ .

Users send independently encoded information. For simplicity, we assume that all user codes have the same code block length  $N$  and that users are frame-synchronous, i.e., their code words are aligned in time. Let  $\mathcal{C}_k$  be the code of user  $k$ , and  $\mathbf{x}_k = (x_k[1], \dots, x_k[N])^T$  denote a code word in  $\mathcal{C}_k$ . Users interleave their code words before transmission. User  $k$  interleaver is defined by a permutation  $\Pi_k$ , randomly and independently chosen with uniform probability in the set of all permutations of  $N$  elements. In the following,  $x_k[i]$  denotes the  $i$ -th code symbol of user  $k$  before interleaving (index  $i$  denotes time ordering at the encoders output), and  $a_k[n]$  denotes the  $n$ -th code symbol of user  $k$  after interleaving (index  $n$  denotes time ordering at the channel output). Then, the vector  $\mathbf{a}[n]$  is formed by

---

<sup>1</sup> $\mathcal{N}_{\mathbb{C}}(\boldsymbol{\mu}, \mathbf{R})$  denotes the circularly-symmetric complex multivariate Gaussian distribution [36] with mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\mathbf{R}$ .

the encoded symbols  $\{a_k[n] = x_k[i_k] : k = 1, \dots, K\}$  for which  $n = \Pi_k(i_k)$ . For the sake of notation simplicity, we shall write  $a_k[n] \equiv x_k[i]$  where it is understood that  $n = \Pi_k(i)$ .

Assuming that an ideal Nyquist pulse with zero excess bandwidth [39] is used for modulating the chips, the system spectral efficiency in bit/s/Hz is given by [51]

$$\rho = \frac{1}{L} \sum_{k=1}^K R_k \quad (2)$$

where  $R_k$  is the coding rate (expressed in bit per complex symbol) of user  $k$ . If the coding rate is equal to  $R$  for all users, we get  $\rho = \alpha R$ , where  $\alpha = K/L$  is the number of “users per chip”, referred to as the *channel load* [47].

### 3 Joint decoding: graph representation and iterative algorithms

In this section we apply the *factor graph* representation [32, 21, 54, 45, 44] to the problem of *maximum a posteriori* (MAP) joint decoding of the user information bits. By applying the *sum-product* algorithm to the resulting factor graph we derive a class of iterative decoding algorithms approximating optimal joint decoding. In the particular case of parallel and serial *scheduling*, we obtain the algorithms proposed in several works (see for example [35, 41, 9, 53, 10]). Finally, by making some simplifying assumptions we derive in a unified way low-complexity versions of parallel and serial IC-based iterative decoding algorithms [46, 1, 33, 53, 15, 34, 40, 44, 8, 17, 26].

#### 3.1 Factor graph representation

Throughout the paper we use the proportionality symbol  $\propto$  in order to indicate that the quantity in the RHS is defined up to a multiplicative factor chosen in order to make it a true probability density (or mass) function.

Let  $\mathbf{b}_k$  be the vector of  $\log_2 |\mathcal{C}_k|$  information bits of user  $k$ . The multiuser channel with coding is fully described by the *a posteriori* pmf of the user information bits given the received signal, denoted by  $\Omega(\mathbf{b}_1, \dots, \mathbf{b}_K | \mathbf{y}[1], \dots, \mathbf{y}[N])$ . By using the fact that the vector channel (1) is memoryless, that users code words are independently generated and that the user information bits have uniform *a priori* probability, we can write

$$\Omega(\mathbf{b}_1, \dots, \mathbf{b}_K | \mathbf{y}[1], \dots, \mathbf{y}[N]) \propto \prod_{n=1}^N q_n(\mathbf{a}[n]) \prod_{k=1}^K p_k(\mathbf{x}_k, \mathbf{b}_k) \quad (3)$$

where we define the *code constraint functions*

$$p_k(\mathbf{x}, \mathbf{b}) = \begin{cases} 1 & \text{if } \mathbf{x} = \phi_k(\mathbf{b}) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

( $\phi_k : \mathbf{b} \mapsto \mathbf{x}$  is the (deterministic) encoding function of code  $\mathcal{C}_k$ ), and the *channel transition functions*

$$q_n(\mathbf{a}) = \exp(-|\mathbf{y}[n] - \mathbf{S}\mathbf{W}\mathbf{a}|^2) \quad (5)$$

The received signal  $\mathbf{y}[n]$  appears as part of the function  $q_n(\mathbf{a})$  and not as an argument because, from the decoder point of view, it is useful to stress only the functional dependence of variables whose *a posteriori* pmf (or likelihood ratio) needs to be calculated [32, 21].

In general, the factor graph for a real-valued  $n$ -variate function  $g(v_1, \dots, v_n)$  factored in the product of  $m$  functions  $f_1, \dots, f_m$  is a bipartite graph  $G(\mathcal{V}, \mathcal{F})$  where variables are represented by nodes  $v \in \mathcal{V}$  and functions by nodes  $f \in \mathcal{F}$ . Each node  $f$  is connected to all nodes  $v$  for which the corresponding variables are arguments of the function  $f$ . In our case, we choose the coded symbols  $x_k[i]$  as variable nodes, taking into account that  $a_k[n] \equiv x_k[i]$ , and the functions  $p_k$  and  $q_n$  as function nodes. Fig. 1 represents the factor graph for  $\Omega(\mathbf{b}_1, \dots, \mathbf{b}_K | \mathbf{y}[1], \dots, \mathbf{y}[N])$  induced by the factorization (3).

**Remark 1.** Each code constraint function block can also be represented as a factor graph, depending on the code structure. For example, if code  $\mathcal{C}_k$  is a trellis code, turbo code [5], LDPC code [22] etc ..., the subgraph formed by the variable nodes  $x_k[1], \dots, x_k[N]$ , by the function node  $p_k$  and by the corresponding information bit nodes can be expanded in the well-known forms [32, 21]. Since our treatment is fully general and applies to any user code constructed on the complex alphabet  $\mathcal{A}$ , we shall not expand further the factor graph of Fig. 1.

In Tanner's terminology [45], when the factor graph represents a compound code, the function nodes are called *subcode* nodes and the variable nodes are called *bitnodes*. In our case, the bitnodes ( $x$ ) have degree 2 and the subcode nodes ( $p$  and  $q$ ) have degree  $N$  and  $K$ , respectively. The graph representation is bipartite but *irregular*, as the subcode nodes have not all the same degree.  $\diamond$

### 3.2 The sum-product algorithm

Let  $b_{k,j}$  denote the  $j$ -th information bit of user  $k$ . The optimal MAP detection rule minimizing the average BER for each user is given by

$$\hat{b}_{k,j} = \arg \max_{b \in \{0,1\}} \text{APP}_{k,j}(b) \quad \text{for all } k, j \quad (6)$$

where  $\text{APP}_{k,j}(b)$  denotes the marginal APP of bit  $b_{k,j}$ , given by

$$\text{APP}_{k,j}(b) = \sum_{\substack{\mathbf{b}_1, \dots, \mathbf{b}_K \\ b_{k,j}=b}} \Omega(\mathbf{b}_1, \dots, \mathbf{b}_K | \mathbf{y}[1], \dots, \mathbf{y}[N]) \quad (7)$$

In general, computing the APP of information symbols by brute-force (i.e., by applying (7) directly) has complexity of the order of  $\prod_{k=1}^K |\mathcal{C}_k|$ . Even for small  $K$ , this is intractable for practical user code sizes. Even in the case where all user codes are trellis codes, the complexity of joint decoding applied to the Cartesian-product trellis of  $\mathcal{C}_1 \times \dots \times \mathcal{C}_K$  is prohibitive in practice [24, 2].

A general method for approximating (7) consists of applying the sum-product algorithm [32, 21] to the factor graph. In the sum-product algorithm the factor graph nodes exchange “messages” along the graph edges. In our case, messages are in the form of real-valued functions. If  $v \in \mathcal{V}$  and  $f \in \mathcal{F}$  are connected by the edge  $(v, f)$ , the messages passed along  $(v, f)$  in either directions are functions of  $v$ . Following [21], we let  $\mu_{v \rightarrow f}(v)$  and  $\mu_{f \rightarrow v}(v)$  indicate the messages passed in directions  $v \rightarrow f$  and  $f \rightarrow v$ , respectively. The sum-product message-passing rules are given by [21]:

- *Variable node to function node.* Let  $\mathcal{L}(v) \subseteq \mathcal{F}$  be the *local neighborhood* of the variable node  $v$ , i.e., the set of function nodes connected to  $v$ . The message passed along the edge  $(v, f)$  in direction  $v \rightarrow f$  is <sup>2</sup>

$$\mu_{v \rightarrow f}(v) \propto \prod_{h \in \mathcal{L}(v) - \{f\}} \mu_{h \rightarrow v}(v) \quad (8)$$

- *Function node to variable node.* Let  $\mathcal{L}(f) \subseteq \mathcal{V}$  be the local neighborhood of the function node  $f$ , i.e., the set of variable nodes connected to  $f$ . The message passed along the edge  $(v, f)$  in direction  $f \rightarrow v$  is<sup>3</sup>

$$\mu_{f \rightarrow v}(v) \propto \sum_{u : u \in \mathcal{L}(f) - \{v\}} f(u \in \mathcal{L}(f)) \prod_{u \in \mathcal{L}(f) - \{v\}} \mu_{u \rightarrow f}(u) \quad (9)$$

For the sake of notation simplicity, we let  $Q_{k,n}(a) = \mu_{q_n \rightarrow a_k[n]}(a_k[n] = a)$  and  $P_{k,i}(a) = \mu_{p_k \rightarrow x_k[i]}(x_k[i] = a)$  denote the messages calculated at the  $n$ -th channel transition function

<sup>2</sup>For a set  $\mathcal{S}$  and  $s \in \mathcal{S}$ , we denote by  $\mathcal{S} - \{s\}$  the set of all elements in  $\mathcal{S}$  except  $s$ .

<sup>3</sup>The short-hand notation  $\sum_{u : u \in \mathcal{L}(f) - \{v\}}$  indicates the sum over all variables  $u$  in  $\mathcal{L}(f)$  excluding  $v$ , where each variable is summed over its domain. In the language of [21], the function  $f(u \in \mathcal{L}(f)) \prod_{u \in \mathcal{L}(f) - \{v\}} \mu_{u \rightarrow f}(u)$  is “summarized” with respect to the variable  $v$ . If  $f(u \in \mathcal{L}(f)) \prod_{u \in \mathcal{L}(f) - \{v\}} \mu_{u \rightarrow f}(u)$  is proportional to a joint pmf (resp. pdf), the result of the summation is proportional to the marginal pmf (resp. pdf) of the variable  $v$  alone.

node  $q_n$  and at the  $k$ -th code constraint function node  $p_k$ , respectively, and sent to the variable node  $x_k[i] \equiv a_k[n]$  ( $a$  denotes a dummy variable taking on values in the symbol alphabet  $\mathcal{A}$ ). It is understood that both  $Q_{k,n}(a)$  and  $P_{k,i}(a)$  are pmf defined over  $\mathcal{A}$ . Therefore, the proportionality factor in (9) is chosen so that  $\sum_{a \in \mathcal{A}} Q_{k,n}(a) = \sum_{a \in \mathcal{A}} P_{k,i}(a) = 1$ .

**Computation at the variable nodes.** The local neighborhood of  $x_k[i] \equiv a_k[n]$  is the set  $\{p_k, q_n\}$ . Hence, the computation at the variable nodes is trivial and consists of passing to  $p_k$  the message coming from  $q_n$  and passing to  $q_n$  the message coming from  $p_k$ , i.e.,

$$\begin{aligned} \mu_{x_k[i] \rightarrow p_k}(x_k[i] = a) &= Q_{k,i}(a) \\ \mu_{a_k[n] \rightarrow q_n}(a_k[n] = a) &= P_{k,n}(a) \end{aligned} \quad (10)$$

**Computation at the channel transition function nodes.** The local neighborhood of  $q_n$  is the set  $\{a_1[n], \dots, a_K[n]\}$ . The message output in the direction of node  $a_k[n]$  is given by

$$Q_{k,n}(a) \propto \sum_{\substack{\mathbf{a} \in \mathcal{A}^K \\ a_k = a}} \exp(-|\mathbf{y}[n] - \mathbf{S}\mathbf{W}\mathbf{a}|^2) \prod_{j \neq k} P_{j,n}(a_j) \quad \text{for } a \in \mathcal{A} \quad (11)$$

**Computation at the code constraint function nodes.** The local neighborhood of  $p_k$  is the set  $\{x_k[1], \dots, x_k[N]\}$ . The message output in the direction of node  $x_k[i]$  is given by

$$P_{k,i}(a) \propto \sum_{\substack{\mathbf{x} \in \mathcal{C}_k \\ x[i] = a}} \prod_{j \neq i} Q_{k,j}(x[j]) \quad \text{for } a \in \mathcal{A} \quad (12)$$

In some works [5, 25, 4], the quantities defined in (12) is referred to as the “extrinsic information” of the decoder. Since  $P_{k,i}(a)$  is a pmf, it will be referred to in the following as the *extrinsic pmf* for symbol  $x_k[i]$ . It should be noticed that  $P_{k,i}(a)$  is not the APP of symbol  $x_k[i]$  given *a priori* marginal pmfs  $\{Q_{k,n}(a) : n = 1, \dots, N\}$  of the coded symbols and the code constraint of  $\mathcal{C}_k$ . The coded symbol APPs are proportional to the product  $Q_{k,i}(a)P_{k,i}(a)$  [5, 4]. In “turbo coding” terminology, the calculation of the extrinsic pmfs (12) is often referred to as *soft-in soft-out* (SISO) decoding [4].

**APP of information bits.** For the sake of MAP detection, the  $k$ -th code constraint function node computes also the APP of the information bits  $b_{k,j}$  as follows

$$\text{APP}_{k,j}(b) \propto \sum_{\substack{\mathbf{x} = \phi_k(\mathbf{b}) \\ b_{k,j} = b}} \prod_{i=1}^N Q_{k,i}(x[i]) \quad \text{for } b \in \{0, 1\} \quad (13)$$

The information bit detection is obtained by using the above APP in (6).

**Remark 2.** If the user codes admit an efficient trellis representation (e.g., they are trellis-terminated convolutional codes), the SISO computation (12) can be carried out by the forward-backward BCJR algorithm [3] with linear complexity in  $N$ . On the contrary, the CDMA vector channel (1) has no particular structure enabling efficient evaluation of (11) and the computation at the channel transition function nodes has complexity of the order of  $M^K$ , i.e., exponential in  $K$ . For large  $K$ , the iterative sum-product algorithm is still too complex for a practical implementation.  $\diamond$

**Scheduling.** It can be shown that the sum-product algorithm is able to compute exactly the marginals of the underlying multivariate function in a finite number of steps if the corresponding factor graph is cycle-free, i.e., it is a *tree* (see [38, 32, 21] and reference therein). Unfortunately, it is apparent from direct inspection of Fig. 1 that our problem yields a factor graph with cycles unless  $K = 1$ . The consequences of cycles are: 1) detection based on (13) is in general suboptimal; 2) the result is sensitive to the order in which computation is carried out through the nodes (scheduling); 3) different scheduling yields generally non-equivalent sum-product algorithms; 4) these algorithms are iterative. A scheduling is generally defined by a sequence of sets of nodes (either function or variable) to be activated. Nodes in the same set can be activated in any arbitrary order, as the value of their output messages does not depend on the activation order within the set. It is understood that when a node is activated all its output messages are calculated by using the current value of its input messages.

In our case, the simplest and most intuitive schedulings are *parallel* and *serial*. In parallel scheduling, one iteration is given by the sequence

$$\begin{aligned} &\{q_n : n = 1, \dots, N\} \rightarrow \{x_k[i] : k = 1, \dots, K, i = 1, \dots, N\} \rightarrow \\ &\rightarrow \{p_k : k = 1, \dots, K\} \rightarrow \{x_k[i] : k = 1, \dots, K, i = 1, \dots, N\} \end{aligned}$$

In serial scheduling users are considered in a given cyclic order (without loss of generality we consider the natural ordering  $k = 1, 2, \dots, K$ ). One iteration is given by the sequence

$$\begin{aligned} &\{q_n : n = 1, \dots, N\} \rightarrow \{x_1[i] : i = 1, \dots, N\} \rightarrow \{p_1\} \rightarrow \{x_1[i] : i = 1, \dots, N\} \rightarrow \\ &\rightarrow \{q_n : n = 1, \dots, N\} \rightarrow \{x_2[i] : i = 1, \dots, N\} \rightarrow \{p_2\} \rightarrow \{x_2[i] : i = 1, \dots, N\} \rightarrow \\ &\quad \vdots \\ &\rightarrow \{q_n : n = 1, \dots, N\} \rightarrow \{x_K[i] : i = 1, \dots, N\} \rightarrow \{p_K\} \rightarrow \{x_K[i] : i = 1, \dots, N\} \end{aligned}$$

In both cases, the algorithm is initialized by the uniform pmf  $P_{k,i}(a) = 1/M$  for all  $a \in \mathcal{A}$ ,  $k = 1, \dots, K$  and  $i = 1, \dots, N$ .

### 3.3 Low-complexity approximations: IC schemes

We notice that (11) consists of computing the *a posteriori* pmf of  $a_k[n]$  given the observation  $\mathbf{y}[n]$ , assuming that the interfering symbols  $a_j[n]$  are statistically independent with marginal pmf  $P_{j,n}(a)$ . The exponential complexity of (11) is due to the fact that the symbols  $a_j[n]$  take on values in the discrete set  $\mathcal{A}$ . By artificially modifying the marginal pmfs of the interfering symbols several low-complexity algorithms can be derived in a unified way.

**Hard IC.** By replacing  $P_{j,n}(a)$  with its single mass point approximation

$$\hat{P}_{j,n}(a) = \begin{cases} 1 & \text{for } a = \arg \max_{a \in \mathcal{A}} P_{j,n}(a) \\ 0 & \text{otherwise} \end{cases}$$

(11) reduces to

$$Q_{k,n}(a) \propto \exp(-\gamma_k |z_{k,n} - a|^2) \tag{14}$$

(recall that  $\gamma_k = |w_k|^2$ ), where

$$z_{k,n} = \frac{1}{w_k} \mathbf{s}_k^H (\mathbf{y}[n] - \mathbf{S}_k \mathbf{W}_k \hat{\mathbf{a}}_{k,n}) \tag{15}$$

$\mathbf{S}_k$  is obtained from  $\mathbf{S}$  by striking out the  $k$ -th column,  $\mathbf{W}_k$  is obtained from  $\mathbf{W}$  by striking out the  $k$ -th column and row, and the vector

$$\hat{\mathbf{a}}_{k,n} = (\hat{a}_{1,n}, \dots, \hat{a}_{k-1,n}, \hat{a}_{k+1,n}, \dots, \hat{a}_{K,n})^T$$

has components given by the symbol-by-symbol hard decisions  $\hat{a}_{j,n} = \arg \max_{a \in \mathcal{A}} P_{j,n}(a)$ .

The variable  $z_{k,n}$  in (15) is the result of subtracting from the output of the single-user matched filter (SUMF)  $\frac{1}{w_k} \mathbf{s}_k^H \mathbf{y}[n]$  the estimated contribution of the interfering symbols  $\sum_{j \neq k} \frac{w_j}{w_k} \mathbf{s}_k^H \mathbf{s}_j \hat{a}_{j,n}$  based on hard symbol-by-symbol detection made at the output of the SISO decoders. Depending on whether parallel or serial scheduling is adopted, we refer to this scheme as hard Parallel IC (h-PIC) or hard Serial IC (h-SIC).

**SUMF-based soft IC.** By replacing  $P_{j,n}(a)$  with the complex circularly-symmetric Gaussian pdf with the same mean and variance [14], given by

$$\tilde{P}_{j,n}(a) = \mathcal{N}_{\mathbb{C}}(\tilde{a}_{j,n}, \xi_{j,n})$$

where

$$\begin{aligned}\tilde{a}_{j,n} &= \sum_{a \in \mathcal{A}} a P_{j,n}(a) \\ \xi_{j,n} &= \sum_{a \in \mathcal{A}} |a - \tilde{a}_{j,n}|^2 P_{j,n}(a) = 1 - |\tilde{a}_{j,n}|^2\end{aligned}\tag{16}$$

$\mathbf{y}[n]$  can be treated as a Gaussian vector conditionally on  $a_k[n]$ . With the additional simplifying assumption that  $\mathbf{y}[n]$  is conditionally white, after some trivial algebra (11) reduces to

$$Q_{k,n}(a) \propto \exp(-\delta_{k,n} |z_{k,n} - a|^2)\tag{17}$$

where we let

$$\delta_{k,n} = \frac{\gamma_k}{1 + \frac{1}{L} \sum_{j \neq k} \gamma_j \xi_{j,n}}$$

and

$$z_{k,n} = \frac{1}{w_k} \mathbf{s}_k^H (\mathbf{y}[n] - \mathbf{S}_k \mathbf{W}_k \tilde{\mathbf{a}}_{k,n})\tag{18}$$

and where the vector

$$\tilde{\mathbf{a}}_{k,n} = (\tilde{a}_{1,n}, \dots, \tilde{a}_{k-1,n}, \tilde{a}_{k+1,n}, \dots, \tilde{a}_{K,n})^T$$

has components given by soft symbol estimates defined in (16).

The variable  $z_{k,n}$  in (18) is the result of subtracting from the output of the SUMF the estimated contribution of the interfering symbols  $\sum_{j \neq k} \frac{w_j}{w_k} \mathbf{s}_k^H \mathbf{s}_j \tilde{a}_{j,n}$  based on soft symbol-by-symbol estimates provided by the SISO decoders. Depending on whether parallel or serial scheduling is adopted, we refer to this scheme as SUMF-based soft Parallel IC (s-PIC-SUMF) or soft Serial IC (s-SIC-SUMF).

**LMMSE-based soft IC.** We make the same conditional Gaussian approximation of  $\mathbf{y}[n]$  as for the SUMF-based schemes, but instead of treating  $\mathbf{y}[n]$  as conditionally white, we take into account its covariance matrix resulting from the assumption that the symbols  $a_j[n]$  are circularly-symmetric complex Gaussian and independent, with marginal pdf  $\tilde{P}_{j,n}(a)$ . The resulting covariance matrix is given by

$$\tilde{\Sigma}_{k,n} = \mathbf{S}_k \mathbf{W}_k \Xi_{k,n} \mathbf{W}_k^H \mathbf{S}_k^H + \mathbf{I}\tag{19}$$

where

$$\Xi_{k,n} = \text{diag}(\xi_{1,n}, \dots, \xi_{k-1,n}, \xi_{k+1,n}, \dots, \xi_{K,n})\tag{20}$$

With the above assumption, after some simple algebra (11) reduces to

$$Q_{k,n}(a) \propto \exp(-\beta_{k,n} |z_{k,n} - a|^2) \quad (21)$$

where we let

$$\beta_{k,n} = \gamma_k \mathbf{s}_k^H \tilde{\Sigma}_{k,n}^{-1} \mathbf{s}_k$$

and

$$z_{k,n} = \frac{w_k^*}{\beta_{k,n}} \mathbf{s}_k^H \tilde{\Sigma}_{k,n}^{-1} (\mathbf{y}[n] - \mathbf{S}_k \mathbf{W}_k \tilde{\mathbf{a}}_{k,n}) \quad (22)$$

The variable  $z_{k,n}$  in (22) is the result of subtracting from the received signal vector  $\mathbf{y}[n]$  the estimated interference vector  $\sum_{j \neq k} w_j \mathbf{s}_j \tilde{a}_{j,n}$  based on soft symbol-by-symbol estimates provided by the SISO decoders, and then filtering the resulting difference vector by the *estimated* unbiased LMMSE filter defined by the filter coefficients vector

$$\mathbf{h}_{k,n} = \frac{w_k}{\beta_{k,n}} \tilde{\Sigma}_{k,n}^{-1} \mathbf{s}_k \quad (23)$$

Depending on whether parallel or serial scheduling is adopted, we refer to this scheme as LMMSE-based soft Parallel IC (s-PIC-LMMSE) or soft Serial IC (s-SIC-LMMSE).

**Remark 3: on the use of APPs versus extrinsic pmfs.** In several papers (e.g., [1, 9, 33, 46, 15, 14]),  $\tilde{a}_{j,n}$  in (16) is calculated by from the APP at the SISO output and not from the exstrinsic pmf  $P_{j,n}(a)$ . As a consequence, the residual interference plus noise at the decoder input,  $\zeta_{k,n} = z_{k,n} - a_k[n]$ , where  $z_{k,n}$  is given either by (18) or by (22)), is biased conditionally on  $a_k[n]$  [11, 18]. In order to illustrate the bias problem we consider the simple case of s-PIC-SUMF with BPSK modulation. In this case,  $\tilde{a}_{j,n}$  can be written as

$$\tilde{a}_{j,n} = \tanh(\mathcal{L}_{j,n}/2) \quad (24)$$

where  $\mathcal{L}_{j,n} = \log \frac{P_{j,n}(+1)}{P_{j,n}(-1)}$  is the extrinsic log-likelihood ratio (LLR) of symbol  $a_j[n]$ . Recalling that the APP at the SISO output is proportional to the product  $Q_{j,n}(a)P_{j,n}(a)$  and by using (17), the *a posteriori* LLR of symbol  $a_j[n]$  is given by  $4\delta_{j,n} \text{Re}\{z_{j,n}\} + \mathcal{L}_{j,n}$  and the soft symbol estimate calculated from the APP is given by

$$\tilde{a}_{j,n} = \tanh(2\delta_{j,n} \text{Re}\{z_{j,n}\} + \mathcal{L}_{j,n}/2) \quad (25)$$

Now, we focus on the real part of the residual interference variable for the  $n$ -th input of the  $k$ -th decoder at iteration  $m$  (superscripts are used for the iteration index). We let

$\nu_{k,n} = \text{Re}\{\mathbf{s}^H \boldsymbol{\nu}[n]/w_k\}$ ,  $r_{k,j} = \mathbf{s}_k^H \mathbf{s}_j$  (this is real since the spreading sequences are real) and  $A_{k,j} \cos(\theta_{k,j}) = \text{Re}\{w_j/w_k\}$  and, by using (18) and (25), we can write

$$\begin{aligned}
 \text{Re}\left\{\zeta_{k,n}^{(m)}\right\} &= \sum_{j \neq k} A_{k,j} r_{k,j} \cos(\theta_{k,j}) \left(a_j[n] - \tilde{a}_{j,n}^{(m)}\right) + \nu_{k,n} \\
 &= \sum_{j \neq k} A_{k,j} r_{k,j} \cos(\theta_{k,j}) \left(a_j[n] - \tanh\left(2\delta_{j,n}^{(m-1)} \text{Re}\{z_{j,n}^{(m-1)}\} + \mathcal{L}_{j,n}^{(m)}/2\right)\right) + \nu_{k,n} \\
 &\stackrel{(a)}{\approx} \sum_{j \neq k} A_{k,j} r_{k,j} \cos(\theta_{k,j}) \left(a_j[n] - 2\delta_{j,n}^{(m-1)} \text{Re}\{z_{j,n}^{(m-1)}\} - \mathcal{L}_{j,n}^{(m)}/2\right) + \nu_{k,n} \\
 &\stackrel{(b)}{=} \sum_{j \neq k} A_{k,j} r_{k,j} \cos(\theta_{k,j}) \left(a_j[n] - 2\delta_{j,n}^{(m-1)} \left(A_{j,k} r_{j,k} \cos(\theta_{j,k}) a_k[n] + v_{j,n}^{(m-1)}\right) - \mathcal{L}_{j,n}^{(m)}/2\right) \\
 &\quad + \nu_{k,n} \\
 &\stackrel{(c)}{=} -2 \left(\sum_{j \neq k} \delta_{j,n}^{(m-1)} r_{k,j}^2 \cos^2(\phi_{k,j})\right) a_k[n] + v_{k,n}^{(m)} \tag{26}
 \end{aligned}$$

where (a) follows by assuming that the signal-to-interference plus noise ratio (SINR) at the SISO decoders inputs of iteration  $(m - 1)$  is small, so that both  $|\mathcal{L}_{j,n}^{(m)}|$  and  $\delta_{j,n}^{(m-1)}$  are small and we can approximate  $\tanh(x) \approx x$ , where (b) follows by collecting the terms proportional to  $a_k[n]$  in  $z_{j,n}^{(m-1)}$  ( $v_{j,n}^{(m-1)}$  denotes the rest), and (c) follows by collecting the terms proportional to  $a_k[n]$  ( $v_{k,n}^{(m)}$  denotes the rest). From (26) we observe that, by using APPs instead of extrinsic pmfs,  $\zeta_{k,n}^{(m)}$  is conditionally biased given  $a_k[n]$ , and the bias has always sign opposite to  $a_k[n]$  (i.e., it decreases the useful signal term). As a matter of fact,  $v_{k,n}^{(m)}$  in the last line of (26) depends on  $a_k[n]$  as well (unless  $m = 1$  [11]) and tends to compensate for the bias if the reliability of the SISO decoders outputs increase with  $m$ . In fact, in the limit for perfect decisions the interference is totally removed and  $\zeta_{k,n}^{(m)}$  is obviously unbiased, as it contains only noise. However, for large load the SISO decoder output in the first iterations is very unreliable and the bias might cause the iterative decoder to get stuck.

From the derivation of Section 3 we notice that the bias problem is naturally avoided by the rigorous application of the sum-product algorithm.<sup>4</sup> As an example, Fig. 2 shows the conditional cumulative distribution function (cdf) of  $\text{Re}\{\zeta_{k,n}^{(m)}\}$  for  $m = 1$  (i.e., after one IC stage) given  $a_k[n] = +1$  and  $a_k[n] = -1$  when  $\tilde{a}_{j,n}^{(m)}$  are calculated from the APPs or from extrinsic pmfs. The cdfs are obtained by simulating the s-PIC-SUMF with  $L = 20$ ,  $K = 40$ , randomly generated spreading sequences and interleavers,  $E_b/N_0 = 6$  dB, BPSK modulation and convolutional code with 4 states, rate 1/2 and generators (5, 7) (octal

---

<sup>4</sup>Strictly speaking, this statement is correct only for block length  $N \rightarrow \infty$  and by using ideal interleaving. This is precisely the assumption made in the analysis of next section.

notation [39]) and block length  $N = 1000$  for all users. The bias of the cdfs resulting from the use of APP is clearly visible, while if extrinsic pmf is used the bias is practically absent already for such a small finite-dimensional system.

Interestingly, since IC schemes are obtained from the sum-product algorithm by quite crude approximations, using APPs instead of extrinsic pmfs might provide better results despite the bias problem, especially for small load. In fact, as it is evident from the example of Fig. 2, APPs provide a generally smaller variance of residual interference. The trade-off between residual interference variance and bias can be optimized by using *partial IC*, i.e., by introducing iteration-dependent coefficients in order to weight optimally the estimated interference (see [11, 18] and references therein). Optimizing particular partial IC schemes is out of the scope of this paper. In the following, we shall focus on the basic algorithms obtained *directly* by the application of the sum-product algorithm (i.e., making use of extrinsic pmfs), without any claim about their optimality.

**Remark 4: relation to previous work.** The h-PIC and h-SIC schemes have been proposed by several authors (see [49] and references therein).

The s-PIC-SUMF has been proposed in [1], where it is motivated as an application of the EM algorithm. Unfortunately, the rigorous application of the EM algorithm does not yield the term  $\delta_{k,n}$  in (17) and would involve hard symbol-by-symbol decisions in the maximization step (missing in the SUMF-based s-PIC/s-SIC). As a matter of fact, the EM method is applied rigorously to multiuser detection in [37] and the resulting algorithm is not equivalent to the s-PIC-SUMF of [1].

The s-PIC-LMMSE has been proposed in [53], where the algorithm is motivated by an heuristic argument. Our unified treatment puts in evidence the relations among all these algorithms. In particular, the LMMSE-based s-PIC/s-SIC follow from the same Gaussian approximation yielding the SUMF-based s-PIC/s-SIC by taking into account that the residual interference vector is not white. In the case of  $L = 1$  (no spreading), the SUMF-based and LMMSE-based algorithms obviously coincide, yielding the soft IC schemes of [46, 14, 15].  $\diamond$

**Remark 5: complexity issues.** Both the h-PIC/h-SIC and the SUMF-based s-PIC/s-SIC algorithms perform IC *after* SUMF filtering. Then, the SUMF outputs for all users can be calculated before iterative decoding and no re-spreading and filtering is needed during the iterations. On the contrary, the LMMSE-based s-PIC/s-SIC algorithms perform IC *before* LMMSE filtering, since the filter vector given by (23) must be recalculated for all symbols at every iteration. Therefore, the LMMSE-based algorithms are much more

complex than the SUMF-based (or hard-decision based) algorithms. As an alternative, the LMMSE filters can be calculated adaptively, as proposed in [34].  $\diamond$

**Remark 6: some important warnings.** It is worthwhile to point out that  $P_{j,n}(a)$  is *not* the true APP of  $a_j[n]$ , even after an arbitrarily large number of iterations.<sup>5</sup> This simple observation has some important consequences:

1) The soft symbol estimates  $\tilde{a}_{j,n}$  are *not* the non-linear MMSE symbol estimates given by the conditional mean  $E[a_j[n]|\mathbf{y}[1], \dots, \mathbf{y}[N]]$ .

2) The *estimated* SINRs  $\gamma_{k,n}$ ,  $\delta_{k,n}$  and  $\beta_{k,n}$  in (14), (17) and (21), respectively, are *not* the true SINR at the SISO decoder input. Actually, they are the SINRs “assumed” by the iterative decoder at the current iteration, based on the *a priori* information provided by the previous iteration. The true SINRs are unknown to the decoder since they depend on the joint statistics of  $a_j[n]$  and  $\tilde{a}_{j,n}$  (or  $\hat{a}_{j,n}$ , in the case of h-PIC/h-SIC). We shall refer to  $\gamma_k$ ,  $\delta_{k,n}$  and  $\beta_{k,n}$  as the *nominal SINR* at the SISO decoder input. Treating the nominal SINRs as the true SINRs (as in [53]) provides optimistic results. In order to see this, notice that  $\xi_{j,n} \in [0, 1]$  while  $E[|a_j[n] - \tilde{a}_{j,n}|^2] \in [0, 4]$ .

3) The filter  $\mathbf{h}_{k,n}$  given by (23) is *not* the true solution of the minimum MSE problem

$$\min_{\mathbf{h}} E \left[ |a_k[n] - \mathbf{h}^H (\mathbf{y}[n] - \mathbf{S}_k \mathbf{W}_k \tilde{\mathbf{a}}_{k,n})|^2 \right]$$

On the contrary, it is the LMMSE filter subject to the assumption  $E[|a_j[n] - \tilde{a}_{j,n}|^2] = \xi_{j,n}$ , which does not hold in general. We refer to  $\mathbf{h}_{k,n}$  as the *nominal LMMSE* filter at the current iteration, based on the *a priori* information provided by the previous iteration.  $\diamond$

From the above discussion we observe that the IC-based decoders previously derived are affected by mismatch. Namely, the SISO decoders works with a nominal input SINR different from the true input SINR. Moreover, the s-PIC/s-SIC LMMSE-based algorithms are mismatched also because the estimated LMMSE filters used at each IC stage do not coincide with the true LMMSE filters. These two types of mismatch can be reduced by suitably modifying the basic algorithms. For example, the true decoder input SINR can be estimated at each iteration and used to weight the SISO decoder metrics [1], and the true value of the residual interference symbol variances  $E[|a_k[n] - \tilde{a}_{k,n}|^2]$  can be estimated at each iteration and used to compute the filters  $\mathbf{h}_{k,n}$ .

---

<sup>5</sup>This is true even if the SISO output APPs instead of the extrinsic pmfs are used, since the factor graph has cycles.

## 4 Asymptotic performance analysis

There are two main obstacles to the performance analysis of the IC algorithms derived in the previous section: 1) The factor graph contains cycles; 2) The output statistics of  $z_{k,n}$  produced by IC and filtering depends in a complicated way on the interferers residual power and spreading sequences. The first obstacle is removed by assuming  $N \rightarrow \infty$  [8]. This approach is common in the analysis of coding with iterative decoding [42, 43, 52]. The basic idea is that in the infinite factor graph representing the system for  $N \rightarrow \infty$  the local neighborhood of any variable up to an arbitrary depth looks like a tree, and the sum-product algorithm provides locally exact marginal computation. The second obstacle is removed by assuming random spreading and a *large system* regime, i.e.,  $K \rightarrow \infty$  while the load  $K/L$  converges to a given value  $\alpha$  [47, 51, 31, 28, 50].

Our analysis technique is based on the probability *density evolution* (DE) [42, 43]. Briefly, the messages  $\{P_{k,i}(a) : a \in \mathcal{A}\}$  and  $\{Q_{k,n}(a) : a \in \mathcal{A}\}$  are random vectors<sup>6</sup> and their pdfs evolve with the iterations by propagating along the factor graph. The probability measure on the messages is induced by the conditional probability measure of the received signals  $\mathbf{y}[1], \dots, \mathbf{y}[N]$  given the transmitted code words  $\mathbf{x}_1, \dots, \mathbf{x}_K$ . The DE can be obtained by iterating the pdf transformation corresponding to a single iteration step of the iterative decoding algorithm. Computation takes place at the channel transition function nodes (corresponding to the IC stage) and at the code constraint function nodes (corresponding to SISO decoding). Next, we use large-system analysis to characterize the IC stage. Then, we give some results characterizing the SISO stage. Finally, as an example, we provide the complete DE algorithm for the s-PIC-LMMSE decoder.

### 4.1 Interference cancellation stage

We assume that  $K/L = \alpha$ , and that the user spreading waveforms are generated with i.i.d. random entries. In the case of hard IC, we define  $V_{k,n} = \gamma_k E[|a_k[n] - \hat{a}_{k,n}|^2]$  and  $U_{k,n} = 0$ . In the case of soft IC, we define  $V_{k,n} = \gamma_k E[|a_k[n] - \tilde{a}_{k,n}|^2]$  and  $U_{k,n} = \gamma_k (1 - |\tilde{a}_{k,n}|^2) = \gamma_k \xi_{k,n}$  (see (16)). At any given iteration of the iterative decoding algorithm, we assume that the following conditions hold:

- C.1 The empirical joint cdf of the pairs  $\{(U_{k,n}, V_{k,n}) : k = 1, \dots, K\}$ , for all  $n$ , converges weakly as  $K \rightarrow \infty$  to a given non-random cdf  $F_{U,V}(u, v)$ , dependent on the iteration step but independent of  $n$ .

---

<sup>6</sup>In general, a  $M$ -valued pmf  $\{p(a) : a \in \mathcal{A}\}$  is a  $M$ -dimensional vector lying in the  $(M-1)$ -dimensional simplex defined by the equality  $\sum_{a \in \mathcal{A}} p(a) = 1$  and by the inequalities  $p(a) \geq 0$  for all  $a \in \mathcal{A}$ .

- C.2 The  $V_{k,n}$ 's are uniformly bounded above and, in the case of soft IC, the  $U_{k,n}$  are uniformly bounded below by a positive number.
- C.3 The empirical cdf of the phase of  $w_k$  converges weakly as  $K \rightarrow \infty$  to the uniform distribution over  $[0, 2\pi]$ .

Notice that  $V_{k,n}$  and  $U_{k,n}$  are the residual and nominal interference power provided by user  $k$  at time  $n$ , respectively. Intuitively, it is reasonable to expect that the performance of IC schemes depends on the joint cdf of the true and nominal residual interference power, and that this is required to converge to some fixed deterministic cdf at every iteration when the number of users gets large.

Since the pmf/pdf of the estimated symbols depends on the pdf of the SISO output messages, the validity of condition C.1 will be addressed later (see Proposition 6). Condition C.2 is somewhat technical (see [28]). The assumption that  $\gamma_k$  is uniformly bounded above and below is not very restrictive in any practical power controlled system. Then, C.2 reduces to assuming that  $E[|a_k[n] - \tilde{a}_{k,n}|^2]$  (or  $E[|a_k[n] - \hat{a}_{k,n}|^2]$ ) is uniformly bounded above and  $(1 - |\tilde{a}_{k,n}|^2)$  is uniformly bounded below. The first statement is always verified, since  $E[|a_k[n]|^2] = E[|\hat{a}_{k,n}|^2] = 1$  and  $E[|\tilde{a}_{k,n}|^2] \leq 1$ . The second statement is more critical, since  $(1 - |\tilde{a}_{k,n}|^2)$  might be smaller than any given constant with positive probability. However, since by definition of  $\tilde{a}_{k,n}$  (see (16)) we have  $|\tilde{a}_{k,n}| \leq 1$  with probability 1, we can multiply  $\tilde{a}_{k,n}$  by  $(1 - \epsilon)$  for a given small  $\epsilon \in (0, 1)$  and use  $(1 - \epsilon)\tilde{a}_{k,n}$  instead of  $\tilde{a}_{k,n}$  in the IC stage. For sufficiently small  $\epsilon$  this incurs in negligible performance loss and C.2 is verified since  $1 - (1 - \epsilon)^2|\tilde{a}_{k,n}|^2 > \epsilon$  with probability 1. Condition C.3 ensures that the statistics of the received signal has circular symmetry (i.e., it is invariant with respect to phase rotations [36]). It is not particularly restrictive since in the uplink of practical systems no phase coherence among the users is assumed. Subject to conditions C.1, C.2 and C.3 we have the following results:

**Proposition 1.** The nominal SINRs  $\delta_{k,n}$  and  $\beta_{k,n}$  converge as  $K \rightarrow \infty$  almost surely to the values  $\gamma_k \eta^{\text{sumf}}$  and  $\gamma_k \eta^{\text{mmse}}$ , respectively, where

$$\eta^{\text{sumf}} = \frac{1}{1 + \alpha \int u dF_{U,V}(u, v)} \tag{27}$$

and  $\eta^{\text{mmse}}$  is the unique non-negative solution of the equation

$$\eta = \frac{1}{1 + \alpha \int \frac{u}{1 + \eta u} dF_{U,V}(u, v)} \tag{28}$$

**Sketch of proof.** This proposition follows immediately from the definition of  $\delta_{k,n}$  and  $\beta_{k,n}$  by applying Theorem 3.1 and Proposition 3.3 of [47] with the strengthening made in [28].  $\square$

**Proposition 2.** The true SINR at the SISO decoder input of the h-PIC/h-SIC and of the SUMF-based s-PIC/s-SIC algorithms converges as  $K \rightarrow \infty$  almost surely to the values  $\gamma_k \kappa^h$  and  $\gamma_k \kappa^{\text{sumf}}$ , respectively, where both  $\kappa^h$  and  $\kappa^{\text{sumf}}$  are given by

$$\kappa = \frac{1}{1 + \alpha \int v dF_{U,V}(u, v)} \tag{29}$$

(Notice that, in general,  $\kappa^h$  and  $\kappa^{\text{sumf}}$  are different since the underlying limiting distribution  $F_{U,V}(u, v)$  is different in the two cases, as the variables  $V_{k,n}$  are defined differently for hard and soft IC).

**Sketch of proof.** This proposition follows immediately by writing the SINR at the output of the hard or SUMF-based soft IC stage as

$$\text{SINR}_{k,n} = \frac{\gamma_k}{1 + \sum_{j \neq k} V_{j,n} |\mathbf{s}_k^H \mathbf{s}_j|^2}$$

and by applying Proposition 3.3 of [47] with the strengthening made in [28].  $\square$

**Proposition 3.** The true SINR at the SISO decoder input of LMMSE-based s-PIC/s-SIC converges as  $K \rightarrow \infty$  almost surely to the value  $\gamma_k \kappa^{\text{mmse}}$ , where

$$\kappa^{\text{mmse}} = \frac{1 - \alpha |\eta^{\text{mmse}}|^2 \int \frac{u^2}{(1+u\eta^{\text{mmse}})^2} dF_{U,V}(u, v)}{1 + \alpha \int \frac{v}{(1+u\eta^{\text{mmse}})^2} dF_{U,V}(u, v)} \tag{30}$$

**Proof.** See Appendix A.  $\square$

**Proposition 4.** The residual interference variable  $\zeta_{k,n} = z_{k,n} - a_k[n]$ , where  $z_{k,n}$  is defined in (15), (18) and (22), is asymptotically circularly symmetric complex Gaussian with mean zero and variance given by  $1/(\gamma_k \kappa^h)$ ,  $1/(\gamma_k \kappa^{\text{sumf}})$  and  $1/(\gamma_k \kappa^{\text{mmse}})$ , in the case of h-PIC/h-SIC, SUMF-based s-PIC/s-SIC and LMMSE-based s-PIC/s-SIC, respectively.

**Sketch of proof.** This proposition follows immediately from Proposition IV.1 of [51] and Theorem 3.2 of [28], after noticing that since the hard and soft symbol estimates  $\hat{a}_{j,n}$  and  $\tilde{a}_{j,n}$  are calculated from the extrinsic pmf  $P_{j,n}(a)$  and interleaving is ideal they are independent of  $a_k[n]$  (i.e., the residual interference after IC is conditionally unbiased given  $a_k[n]$ ).  $\square$

**Remark 7.** The true SINR of user  $k$  in the absence of the other users is equal to its SNR  $\gamma_k$ . Therefore,  $\kappa^h$ ,  $\kappa^{\text{sumf}}$  and  $\kappa^{\text{mmse}}$  are the AME of hard IC, SUMF-based and LMMSE-based soft IC at the current iteration step, in the large system regime. Also,  $\eta^{\text{sumf}}$  and  $\eta^{\text{mmse}}$  are the *nominal* AME of SUMF-based and LMMSE-based soft IC. From (14) we see that the nominal AME of hard IC is equal to 1 (corresponding to error-free IC).  $\diamond$

**Remark 8.** Proposition 3 has some interest on its own. Namely, it provides the AME of a mismatched LMMSE receiver with imperfect knowledge of the interferers powers, in the large system regime.  $\diamond$

## 4.2 SISO decoding stage

In this section we define a class of codes for which the evaluation of the pdf of the SISO output messages  $\{P_{k,i}(a) : a \in \mathcal{A}\}$  from a given pdf of the SISO input messages  $\{Q_{k,j}(a) : a \in \mathcal{A}\}$  is particularly simple. Consider a code  $\mathcal{C}$  over a complex signal set  $\mathcal{A}$ , with transmission in AWGN. Then, we have the following definitions: <sup>7</sup>

**Definition: Uniformity.** Let  $f_{P_i|\mathbf{x}}(p_1, \dots, p_M)$  denote the  $M$ -variate pdf of the SISO output message  $\{P_i(a) : a \in \mathcal{A}\}$  corresponding to the  $i$ -th coded symbol, induced by the conditional probability measure of the received signal  $\mathbf{y}$  given the transmitted signal  $\mathbf{x}$ . The code  $\mathcal{C}$  is said to be *uniform* if, for all  $i = 1, \dots, N$ ,  $f_{P_i|\mathbf{x}}(p_1, \dots, p_M) = f_{P_i|\mathbf{x}'}(p_1, \dots, p_M)$  for all  $\mathbf{x}' \in \mathcal{C}$  such that  $x'_i = x_i$ . In words,  $f_{P_i|\mathbf{x}}(p_1, \dots, p_M)$  depends only on the symbol  $x_i$  transmitted in position  $i$  and not on the whole code word  $\mathbf{x}$ .  $\diamond$

For uniform codes, we shall write  $f_{P_i|x}(p_1, \dots, p_M)$  for  $f_{P_i|\mathbf{x}}(p_1, \dots, p_M)$  where  $x_i = x$  is the symbol transmitted in position  $i$ .

**Definition: symbol-by-symbol symmetry.** The code  $\mathcal{C}$  is said to be *symbol-by-symbol symmetric* if it is uniform and, for all  $i = 1, \dots, N$  and any two symbols  $x, x' \in \mathcal{A}$  there exists a permutation  $\pi$  of the integers  $1, \dots, M$  such that

$$f_{P_i|x'}(p_1, \dots, p_M) = f_{P_i|x}(p_{\pi(1)}, \dots, p_{\pi(M)})$$

$\diamond$

---

<sup>7</sup>These definitions are stated as properties of the code alone, since the channel here is fixed to be complex circularly-symmetric AWGN. In a more general setting, these are properties of the code *and* the channel together.

**Definition: Isotropy.** A code  $\mathcal{C}$  is said to have *unisotropy degree*  $d$  if it is symbol-by-symbol symmetric and if as  $i$  ranges from 1 to  $N$  the pdf  $f_{P_i|x}(p_1, \dots, p_M)$  ranges in a set of  $d$  possible pdfs. If  $d = 1$ , the code is said to be *isotropic*.  $\diamond$

The symmetry conditions given here are related to those given in [43] (and references therein) for binary-input symmetric-output channels. Code isotropy is discussed in [52]. In particular, it is shown that linear cyclic block codes are isotropic and that time-invariant trellis codes with  $s$  symbols per trellis branch have unisotropy degree  $d \leq s$ . The following proposition establishes symbol-by-symbol symmetry for a wide class of codes.

**Proposition 5.** Let  $\mathcal{C}$  be a group code with Abelian generating group [29, 27], then  $\mathcal{C}$  is symbol-by-symbol symmetric.

**Proof.** See Appendix A.  $\square$

In particular, linear codes mapped onto BPSK modulation and geometrically uniform  $M$ -PSK codes constructed from cyclic groups [12] satisfy Proposition 5.

### 4.3 Density evolution

The following proposition provides a sufficient condition (met in several cases of interest) for which conditions C.1 holds at every iteration step of the iterative decoding algorithm.

**Proposition 6.** Assume that the users are partitioned into a finite number  $c$  of classes. Users belonging to the same class have the same SNR and use the same channel code. Moreover, user codes are symbol-by-symbol symmetric and have bounded unisotropy degree  $\leq d$ . Then, condition C.1 holds.

**Sketch of proof.** This proposition follows by noticing that at each iteration the SISO output messages are distributed in at most  $D = cd < \infty$  possible ways. Therefore, for each symbol interval  $n$ , there are  $K/D \rightarrow \infty$  SISO output messages distributed according to the same pdf. This implies that the empirical cdf of the SISO output messages converges to a convex combination of the  $D$  possible message cdfs (see [28] for a similar argument).

$\square$

Subject to the assumptions of propositions 1 – 6, we are now ready to consider DE. Fig. 3 shows the local neighborhood of nodes  $q_n$  and  $p_k$  connected through the variable node  $x_k[i]$ . The basic DE step is as follows: from the pdf of the “ $P$ ” messages (SISO

decoding output), determine the pdf of the “ $Q$ ” messages (IC stage output). Then, from the pdf of the “ $Q$ ” messages determines the pdf of the “ $P$ ” messages at next iteration. From propositions 1–4 it follows that the BER for user  $k$  at the current iteration is a one-to-one function of the true SINR  $\gamma_k \kappa$  (where the AME  $\kappa$  is equal to  $\kappa^h$ ,  $\kappa^{\text{sumf}}$  or  $\kappa^{\text{mmse}}$  depending on the algorithm). Therefore, for all algorithms considered here the DE is described by the evolution of the single real non-negative parameter  $\kappa$  versus the number of iterations. If  $\kappa \rightarrow 1$  as the number of iterations increases, the so called “single-user bound” is achieved, i.e., each user is able to achieve the same BER as if it was alone in the system.

In the following, we give explicitly the DE recursion for the case of s-PIC-LMMSE. Analogous DEs can be easily obtained for the other iterative decoding algorithms considered in this paper, and are not included because of space limitations. Throughout this section we denote by  $c$  the number of user classes, we let  $\alpha_\ell$  for  $\ell = 1, \dots, c$  denote the fraction of users per chip in class  $\ell$ , such that  $\sum_{\ell=1}^c \alpha_\ell = \alpha$ , and we let  $g_\ell$  for  $\ell = 1, \dots, c$  denote the values of received SNR for class  $\ell$  (i.e.,  $\gamma_k = g_\ell$  for all users  $k$  belonging to class  $\ell$ ).

**DE for the s-PIC-LMMSE.** Before any interference cancellation, we have that  $U_{k,n} = V_{k,n} = \gamma_k$ . Hence,  $F_{U,V}(u, v)$  must be initialized to the cdf of the received SNRs

$$F_{U,V}(u, v) = \frac{1}{\alpha} \sum_{\ell=1}^c \alpha_\ell \mathbf{1}\{g_\ell \leq u, g_\ell \leq v\} \tag{31}$$

Let  $\eta_0$  and  $\kappa_0$  be the nominal and true AME of s-PIC-LMMSE computed according to Proposition 1 and Proposition 3, respectively, from the cdf  $F_{U,V}(u, v)$  given above (notice that  $\eta_0 = \kappa_0$ ). Let  $m$  denote the iteration index. At each iteration  $m = 0, 1, 2, \dots$  the DE algorithm must evaluate the new empirical joint cdf of  $U_{k,n}$  and  $V_{k,n}$ , and compute the new nominal and true AMEs according to the following recipe:

1. For  $\ell = 1, \dots, c$ , repeat a sufficiently large number of times the steps:
  - Generate the SISO input signal  $\mathbf{z} = \mathbf{x} + \boldsymbol{\zeta}$  where  $\mathbf{x}$  is selected with uniform probability over the code words of class  $\ell$  code and  $\boldsymbol{\zeta}$  is random with i.i.d. components  $\sim \mathcal{N}_{\mathbb{C}}(0, 1/(g_\ell \kappa_m))$ .
  - Calculate the corresponding sequence of SISO input messages

$$Q_i(a) \propto \exp(-g_\ell \eta_m |z[i] - a|^2) \quad a \in \mathcal{A} \tag{32}$$

- Apply the SISO decoder to the sequence of messages (32), calculate the encoded symbol soft estimates  $\tilde{a}_i$  according to (16) and store the values  $u_{\ell,i} = g_\ell(1-|\tilde{a}_i|^2)$  and  $v_{\ell,i} = g_\ell|x_i - \tilde{a}_i|^2$  in a buffer.
2. For  $\ell = 1, \dots, c$ , compute the empirical mean  $V_\ell = \frac{1}{B} \sum_{i=1}^B v_{\ell,i}$ , where  $B$  is the size of the random sample generated above.
  3. Obtain the new empirical cdf  $F_{U,V}(u, v)$  as

$$F_{U,V}(u, v) = \frac{1}{\alpha} \sum_{\ell=1}^c \frac{\alpha_\ell}{B} \sum_{i=1}^B 1\{u_{\ell,i} \leq u, V_\ell \leq v\} \quad (33)$$

4. Compute  $\eta_{m+1}$  and  $\kappa_{m+1}$  according to Proposition 1 and 3, respectively, from the cdf  $F_{U,V}(u, v)$  given in (33) and go to the next iteration step.

Ideally matched SISO decoding can be taken into account by replacing  $\eta_m$  with  $\kappa_m$  in (32). Also, ideally matched LMMSE filters can be taken into account by calculating  $\kappa^{\text{mmse}}$  as the unique non-negative solution of

$$\kappa = \frac{1}{1 + \alpha \int \frac{v}{1+\kappa v} dF_{U,V}(u, v)} \quad (34)$$

instead of using Proposition 3.

**Remark 9.** Our asymptotic analysis is based on the claim that as  $N \rightarrow \infty$  the performance of systems with finite-length random interleavers “concentrates” around the performance of the infinite-dimensional system whose factor graph is cycle-free. This claim holds for trellis codes and SISO decoders operating on a finite sliding window [43]. Since the sliding-window BCJR algorithm provides practically the same performance of the BCJR algorithm applied to the whole code block length [4], the DE performance in the examples of Section 5 (based on convolutional codes) is actually approached with arbitrarily large probability by finite-dimensional random systems as  $N \rightarrow \infty$ . This conclusion is supported by the simulation results of Section 5. Establishing rigorous *concentration theorems* (see [43] and reference therein) for more general classes of codes is of great theoretical and practical interest.  $\diamond$

#### 4.4 Approximated analysis of iterative PIC

In this section we present an approximated analysis allowing the description of PIC algorithms in terms of a one-dimensional non-linear dynamical system. Throughout this

section we assume that all users are received with the same SNR  $\gamma$  and use the same channel code. Subject to these assumptions, with PIC decoding the system is completely symmetric with respect to any user. Therefore, at any iteration all users have the same AME at their SISO decoder input and symbol error rate (SER) at their SISO decoder output. Let  $\epsilon = \epsilon(\gamma)$  denote the SER (of the encoded symbols) at the output of the SISO decoder as a function of the input SNR  $\gamma$  and suppose that we are able to find a function  $\kappa = \kappa(\epsilon)$  relating the true AME at the input of any SISO decoder at iteration  $m + 1$  to the SER  $\epsilon$  at the output of the SISO decoders at iteration  $m$ . Then, evolution of the AME vs. the number iterations is described by the dynamical system

$$\begin{aligned} \epsilon_m &= \epsilon(\kappa_m \gamma) \\ \kappa_{m+1} &= \kappa(\epsilon_m) \end{aligned} \tag{35}$$

for  $m = 0, 1, 2, \dots$ , with the initial condition

$$\kappa_0 = \begin{cases} 1/(1 + \alpha\gamma) & \text{h-PIC and s-PIC-SUMF} \\ \frac{\gamma(1-\alpha)-1+\sqrt{(\alpha\gamma+1)^2+2\gamma+(1-2\alpha)\gamma^2}}{2\gamma} & \text{s-PIC-LMMSE} \end{cases} \tag{36}$$

(the expression of  $\kappa_0$  for the s-PIC-LMMSE follows since at the first iteration the nominal and the true AMEs are equal to the AME of the standard linear MMSE detector, and the closed-form result of [51, 47] applies).

The SER function  $\epsilon = \epsilon(\gamma)$  must be obtained by simulation, since it is the SER of symbol-by-symbol decisions based on the extrinsic pmf only, and the usual union bounding techniques developed for ML decoding do not apply. However, this is much simpler than DE and (above all) than simulation of the whole iterative decoder.

In the case of h-PIC, the AME function is obtained by writing

$$\begin{aligned} V_{k,n} &= \gamma E[|a_k[n] - \hat{a}_{k,n}|^2] \\ &= 2\gamma (1 - \text{Re}\{E[a_k[n]\hat{a}_{k,n}^*]\}) \\ &\approx 2\gamma \left( 1 - (1 - \epsilon) - \frac{\epsilon}{M-1} \text{Re} \left\{ \sum_{i=1}^{M-1} e^{j2\pi i/M} \right\} \right) \\ &= 2\gamma \epsilon \frac{M}{M-1} \end{aligned} \tag{37}$$

where we assumed that, if a symbol is in error, any other symbol appears with uniform probability  $\epsilon/(M-1)$ , and we used the fact that  $\sum_{i=1}^{M-1} e^{j2\pi i/M} = -1$ . By using (37) in (29) we obtain

$$\kappa^h(\epsilon) = \frac{1}{1 + \alpha 2\gamma \epsilon \frac{M}{M-1}} \tag{38}$$

In order to obtain the AME function in the case of soft IC we use a *Gaussian approximation* (GA) of the SISO output [23, 55]. We restrict to BPSK and Gray-labeled QPSK modulations and consider the real binary-input AWGN channel  $y = a + \nu$ , where  $a \in \{\pm 1\}$  and  $\nu \sim \mathcal{N}(0, 1/\lambda)$ . The LLR for the ML detection of  $a$  from the observation  $y$  is given by

$$\mathcal{L}(y) = \log \frac{p(y|a = +1)}{p(y|a = -1)} = 2\lambda y$$

The conditional distribution of  $\mathcal{L}(y)$  given  $a$  is  $\mathcal{N}(2\lambda a, 4\lambda)$ , and the error probability is

$$\epsilon = \Pr(\mathcal{L}(y) < 0 | a = +1) = Q(\sqrt{\lambda})$$

where  $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$  is the Gaussian tail function.

We approximate the output of the SISO decoder with SER  $\epsilon$  by the output of a “virtual” uncoded binary-input AWGN channel with SNR

$$\lambda(\epsilon) = [Q^{-1}(\epsilon)]^2 \tag{39}$$

Therefore, the *a posteriori* LLR at the SISO decoder output corresponding to the (coded) symbol  $a$  is approximated by a Gaussian random variable  $\sim \mathcal{N}(2\lambda(\epsilon)a, 4\lambda(\epsilon))$ . Without loss of generality, we assume  $a_k[n] = +1$ . Then, the corresponding SISO output messages are approximated by

$$\begin{aligned} P_{k,n}(+1) &= \frac{e^{\mathcal{L}_{k,n}}}{e^{\mathcal{L}_{k,n}} + 1} \\ P_{k,n}(-1) &= \frac{1}{e^{\mathcal{L}_{k,n}} + 1} \end{aligned} \tag{40}$$

where  $\mathcal{L}_{k,n} \sim \mathcal{N}(2\lambda(\epsilon), 4\lambda(\epsilon))$ . The soft symbol estimate  $\tilde{a}_{k,n}$  is given by

$$\tilde{a}_{k,n} = P_{k,n}(+1) - P_{k,n}(-1) = \frac{e^{\mathcal{L}_{k,n}} - 1}{e^{\mathcal{L}_{k,n}} + 1}$$

By using this in the expression for  $V_{k,n}$  and  $U_{k,n}$  we obtain

$$\begin{aligned} V_{k,n} &= \gamma E \left[ \frac{4}{(e^{\mathcal{L}_{k,n}} + 1)^2} \right] \\ U_{k,n} &= \gamma \left( 1 - \left| \frac{e^{\mathcal{L}_{k,n}} - 1}{e^{\mathcal{L}_{k,n}} + 1} \right|^2 \right) \end{aligned} \tag{41}$$

The  $V_{k,n}$ 's are again constant (as in (37)) and the cdf of the  $U_{k,n}$ 's is easily found in closed form as

$$F_U(u) = \begin{cases} 0 & u < 0 \\ Q \left( \frac{\log \left( \frac{1 + \sqrt{1 - u/\gamma}}{1 - \sqrt{1 - u/\gamma}} \right) - 2\lambda}{\sqrt{4\lambda}} \right) + Q \left( \frac{\log \left( \frac{1 + \sqrt{1 - u/\gamma}}{1 - \sqrt{1 - u/\gamma}} \right) + 2\lambda}{\sqrt{4\lambda}} \right) & u \in [0, \gamma] \\ 1 & u > \gamma \end{cases} \tag{42}$$

where  $\lambda = \lambda(\epsilon)$  is given by (39). Although it is not possible to express  $\kappa = \kappa(\epsilon)$  in closed form as in the case of hard IC (see (38)), the AME function can be obtained numerically by using (42) in Propositions 1, 2 and 3 and letting  $\epsilon$  vary in the interval  $[0, 1/2]$ .

For Gray-labeled QPSK modulation we assume that each user encodes the in-phase and quadrature symbols independently by using the same binary linear code. The same approach followed for BPSK carries through immediately (we omit the details for the sake of space limitation).

**Remark 10.** Since the analysis based on GA derives the approximated SISO output statistics from its output SER only, this type of analysis is not able to catch the effect of the SISO decoding SINR mismatch. Hence, the GA-based analysis describes the behavior of iterative soft IC with ideally matched SISO, aware of the true SINR at its input. We can also modify the GA-based analysis to describe a s-PIC-LMMSE with ideally matched LMMSE filters by using (34) instead of Proposition 3 to calculate the AME function.  $\diamond$

## 5 Results for PIC with equal-power users

In this section we present some results for PIC with equal-power users. First, we compare the DE of infinite-dimensional systems with the performance of finite-dimensional systems. Then, we compare the GA-based approximated analysis with DE and we discuss the threshold behavior of the decoder with respect to the channel load. Finally, we use the GA-based analysis to determine the system spectral efficiency for simple convolutional codes and PSK/QPSK modulation.

**Finite-dimensional vs. asymptotic systems.** Fig. 4 shows the AME vs. the number of iterations of a finite-dimensional system with  $L = 32$  and  $K = 64$  and 80 users (corresponding to points “+” and “o”, respectively), with s-PIC-SUMF decoding, randomly generated spreading sequences, (5, 7) convolutional code, BPSK and  $E_b/N_0 = 6$  db. Decoding is implemented by the BCJR algorithm with mismatched input (17). For each iteration step a cloud of points corresponding to the AMEs of all  $K$  users is plotted. The AME is calculated by averaging the residual interference plus noise power over the code word block length. In this example, the block length is  $N = 1000$ . The AME of DE (infinite-dimensional system) with the same load  $\alpha = 2.0$  and 2.5 is shown for comparison. The random finite-dimensional system shows good agreement with its infinite-dimensional limit. Unfortunately, we could not simulate systems with larger  $L$  and  $K$  or with LMMSE-based decoders since the complexity is just too large.

**Comparison of DE and GA-based analysis.** Since the GA-based analysis is not able to take into account the SISO mismatch, from now on we consider only algorithms making use of ideally matched SISO. As observed before, these algorithms can be approximated in practice by estimating the true SINR at the SISO decoder inputs for each iteration.

Fig. 5 shows the true AME vs. the number of iterations of h-PIC where all users have  $E_b/N_0 = 6$  dB and use the (5,7) convolutional code with BPSK modulation. Points refer to DE and lines to GA. Curves for  $\alpha = 1.5, 1.8$  and  $2.5$  are shown. GA shows good agreement with DE for  $\alpha = 1.5$  and  $\alpha = 2.5$ , while for  $\alpha = 1.8$  it is still able to predict exactly the limiting AME value but not exactly the whole trajectory.

When the AME converges to 0 dB, the single-user bound is attained. The decoder shows a sort of *threshold effect* with respect to  $\alpha$ . For small  $\alpha$  it converges to the single-user performance, while for large  $\alpha$  it does not improve with iterations and remains far from the single-user bound. The threshold effect is explained by looking at the trajectory of system (35). Figs. 6, 7 and 8 show the functions  $\text{SER} = \epsilon(\gamma)$  and  $\text{SINR} = \gamma \kappa^h(\epsilon)$  in the (SER, SINR) plane (log-log scale), and the trajectory of (35) for  $\alpha = 1.5, 1.8$  and  $2.5$ , respectively, in the same conditions of Fig. 5. For  $\alpha = 1.5$  and  $1.8$  the two curves intersect in a single point, which is the unique (stable) fixed point of (35). Notice that for  $\alpha = 1.8$  the curves are closer than for  $\alpha = 1.5$ , and more iterations are needed to converge. For  $\alpha = 2.5$  the two curves intersect in three points. The leftmost and the rightmost are stable fixed points and the middle is an unstable fixed point. It turns out that with initial condition (36) the iterative decoder converges to the leftmost fixed point, characterized by a very large SER. We refer to this behavior as “above threshold” (the threshold in this case is  $\alpha = 2.01$ ).

The behavior observed in the example holds in general for all the IC-based decoders considered in this paper provided that the operating  $E_b/N_0$  is sufficiently large. The SINR curve has a very steep “waterfall” behavior and has a vertical asymptote for  $\text{SINR} = \gamma$ . In fact, for SER equal to zero the interference can be completely removed and the channel behaves as a single-user AWGN channel with  $\text{SINR} = \text{SNR} = \gamma$  for all users. The SER curve of usual convolutional codes has a much less steep waterfall behavior. Therefore the SINR and SER curves intersect either in one or in three points. The limiting case, determining the threshold of  $\alpha$ , is when the curves intersect in one point and are tangent in another point. Because of the steepness of the SINR curve, for  $E_b/N_0$  large enough (i.e., such that the corresponding single-user BER is small), when the decoder converges to the rightmost fixed point the single-user performance is almost achieved (this behavior has been shown experimentally by simulation of finite-dimensional systems in several works [8, 15, 33, 1, 46, 35, 53]). On the contrary, if the system has two stable fixed

point, the initial condition (36), which is also a function of the load  $\alpha$ , is such that the decoder converges to the leftmost fixed point and its performance is very far from the single-user bound. Fig. 9 shows the SER curve of the (5, 7) convolutional code and the SINR curves for  $E_b/N_0 = 6$  dB for h-PIC, s-PIC-SUMF, s-PIC-LMMSE and s-PIC-LMMSE with ideally matched MMSE filter, for  $\alpha = 1.8$ .

Fig. 10 shows the true AME vs. the number of iterations of s-PIC-SUMF in the same conditions as before with  $\alpha = 2.5, 2.8, 3.5$ , and Fig. 11 shows analogous curves for s-PIC-LMMSE with  $\alpha = 3.0, 3.3, 3.7$ . Curves for the s-PIC-LMMSE with ideally matched LMMSE filters (denoted by “id.”) are also shown for  $\alpha = 3.7, 4.0$ . This can be regarded as an upper bound on the performance of a decoder that calculates an approximation of the LMMSE filters by estimating the residual interference variance at every iteration.

In order to illustrate the SISO mismatch problem, in Fig. 12 we plot true and nominal AME of s-PIC-LMMSE with mismatched SISO decoders in the same conditions of Fig. 11 for  $\alpha = 2.5$  and  $3.0$ . For  $\alpha = 2.5$  the true and nominal AME are very close and both converge rapidly to 1. For  $\alpha = 3.0$  the nominal AME grows and converges to a large value, while the true AME converges to a small value. Thus, the SISO decoder works assuming an input SINR much larger than the true SINR. This makes the decoder give large reliability weights to wrong decisions, and the benefit of soft interference cancellation is reduced. In passing, we notice that analyzing these detectors on the basis of the nominal SINR (as done in [53]) is very dangerous and provides accurate results only for small load  $\alpha$ , i.e., far from threshold.

**System spectral efficiency.** We use the GA-based analysis as a *rule of thumb* for a quick evaluation of the spectral efficiency achievable with given codes and PIC iterative decoders. Spectral efficiency is calculated as follows. We fix the target BER to be achieved by all users and we compute  $E_b/N_0$  necessary to achieve the target BER in the single-user case. If the target BER is sufficiently small, because of the threshold behavior explained above, it can be achieved only if the decoder works below threshold, i.e., if the single-user bound is attained. Then, the resulting spectral efficiency is given by  $\rho = \alpha^* R$ , where  $\alpha^*$  is the threshold load for the value of  $E_b/N_0$  obtained and  $R$  is the coding rate.

Figs. 13, 14 and 15 show the spectral efficiency for target BER equal to  $10^{-5}$  achieved by convolutional codes of rate  $1/2$ ,  $1/3$  and  $1/4$ , respectively, with BPSK and Gray-labeled QPSK modulation with h-PIC, s-PIC-SUMF and s-PIC-LMMSE with ideally matched SISO. We considered also s-PIC-LMMSE with ideally matched LMMSE filters (curves denoted by “id.”). For each coding rate we considered the optimal binary convolutional codes with 4, 8, 16, 32 and 64 states given in [39]. In all figures coded systems are

represented by points in the  $(E_b/N_0, \rho)$  plane. For each set of results the number of trellis states increases from right to left (i.e., the rightmost point corresponds to the 4-state code and the leftmost point to the 64-state code). Larger spectral efficiency is achieved by simpler codes, at the price of a larger required  $E_b/N_0$  to achieve the target BER.

The QPSK system is obtained by independently encoding the in-phase and quadrature symbols. Therefore, QPSK yields exactly the same spectral efficiency of BPSK with h-PIC and s-PIC-SUMF (the user coding rate is doubled, but the threshold load  $\alpha^*$  is halved with respect to the corresponding BPSK system, analogously to what observed for uncoded linear detectors [13]). With s-PIC-LMMSE things are different. In fact, the LMMSE performance is sensitive to the dimensionality of the space spanned by interference and better performance is achieved for the QPSK system (less users with larger coding rate), in agreement with what found for uncoded linear detection [13]. For the sake of comparison, the spectral efficiency with random spreading in a large-system regime achieved by linear detectors (SUMF and LMMSE) with single-user decoding and by the optimal joint decoder with ideal Gaussian random codes is also shown (adapted from [51]).

## 6 Open issues

We would like to conclude this work by pointing out some suggestions for future work.

1. The framework developed in this paper can be extended to more general CDMA models, involving asynchronous transmission, multipath propagation and joint data and channel estimation. From the analysis point of view, the DE approach may be extended by using the results of [31] to include asynchronous CDMA and the results of [19] to include multipath propagation and the effect of channel estimation errors.
2. If compound codes suitable for iterative decoding are used (e.g., turbo-codes or, more in general, Tanner codes [45]), each code constraint function in the basic factor graph of Fig. 1 can be expanded into the factor graph for the corresponding user code. Then, different scheduling options in addition to the plain parallel and serial ones considered in this work might prove to be useful. In particular, partial decoding of each user might yield better results than full decoding. For example, in [14, 15] a system based on s-PIC-SUMF and turbo codes is proposed where at each iteration of the joint decoder only one iteration of the user turbo decoders is performed.
3. The convergence speed and threshold load of the various algorithms might be im-

proved by allocating non-uniform user powers (e.g., in [14, 15] a soft “onion-peeling” structure is created by a geometrically increasing power allocation).

4. The GA-based analysis should be extended to the case of unequal received powers and to SIC algorithms. In both cases, the SER at the decoder outputs is not the same for all users and the AME is a function of all different SERs. Then, the system behavior is described by a multidimensional non-linear dynamical system.
5. Better IC-based multiuser joint decoders could be obtained by seeking an optimal trade-off between the bias and the variance of residual interference (especially at the first iteration steps, which dominate the threshold load). Our asymptotic analysis might provide a useful tool for the optimization of the weights of partial IC.

## Appendix

### A Proofs

#### A.1 Proof of Proposition 3

At any given iteration of the LMMSE-based s-PIC/s-SIC, for given  $k$  and  $n$ , let

$$\mathbf{y}_{k,n} = \mathbf{S}_k \mathbf{W}_k (\mathbf{a}_k[n] - \tilde{\mathbf{a}}_{k,n}) + \boldsymbol{\nu}[n] \quad (43)$$

denote the residual interference plus noise vector after IC, where we define  $\mathbf{a}_k[n] = (a_1[n], \dots, a_{k-1}[n], a_{k+1}[n], \dots, a_K[n])^T$ . Subject to the assumption of infinite interleaving, the true covariance matrix of  $\mathbf{y}_{k,n}$  is given by

$$\boldsymbol{\Sigma}_{k,n} = \mathbf{S}_k \mathbf{V}_{k,n} \mathbf{S}_k^H + \mathbf{I} \quad (44)$$

while the nominal covariance matrix (i.e., the matrix used by the decoder to construct the approximated LMMSE filter  $\mathbf{h}_{k,n}$  in (23)) is given by (see (19))

$$\tilde{\boldsymbol{\Sigma}}_{k,n} = \mathbf{S}_k \mathbf{U}_{k,n} \mathbf{S}_k^H + \mathbf{I} \quad (45)$$

where we have defined

$$\begin{aligned} \mathbf{V}_{k,n} &= \text{diag}(V_{1,n}, \dots, V_{k-1,n}, V_{k+1,n}, \dots, V_{K,n}) \\ \mathbf{U}_{k,n} &= \text{diag}(U_{1,n}, \dots, U_{k-1,n}, U_{k+1,n}, \dots, U_{K,n}) \end{aligned}$$

The SINR at the SISO input is given by

$$\text{SINR}_{k,n} = \gamma_k \frac{|\mathbf{s}_k^H \tilde{\boldsymbol{\Sigma}}_{k,n}^{-1} \mathbf{s}_k|^2}{\mathbf{s}_k^H \tilde{\boldsymbol{\Sigma}}_{k,n}^{-1} \boldsymbol{\Sigma}_{k,n} \tilde{\boldsymbol{\Sigma}}_{k,n}^{-1} \mathbf{s}_k} \quad (46)$$

By noticing that  $\beta_{k,n} = \gamma_k \mathbf{s}_k^H \tilde{\boldsymbol{\Sigma}}_{k,n}^{-1} \mathbf{s}_k$  and by applying Proposition 1 we get immediately that the numerator of the fraction in the RHS of (46) converges almost surely to  $|\eta^{\text{mmse}}|^2$ . By following the same steps of the proof of Theorem 3.2 in [28] we obtain that the denominator of the fraction in the RHS of (46) converges almost surely to the quantity

$$\Delta = \alpha E_{U,V} \left[ \frac{V \int \frac{1}{(\lambda+1)^2} dG(\lambda)}{\left(1 + U \int \frac{1}{\lambda+1} dG(\lambda)\right)^2} \right] + \int \frac{1}{(\lambda+1)^2} dG(\lambda) \quad (47)$$

where  $E_{U,V}[\cdot]$  denotes expectation with respect to the limiting joint statistics of  $U_{k,n}$  and  $V_{k,n}$  (which is well-defined by condition C.1) and where  $G(\lambda)$  is the limiting eigenvalue cdf

of the covariance matrix  $\tilde{\Sigma}_{k,n}$ , which is guaranteed to converge (see [47, 28] and references therein). Let  $m_G(z)$  be the Stieltjes transform of  $G(\lambda)$ , given by

$$m_G(z) = \int \frac{1}{\lambda - z} dG(\lambda) \quad (\text{Im}\{z\} > 0)$$

Then,  $m_G(z)$  satisfies the equation [47]

$$m_G(z) = \frac{1}{-z + \alpha \int \frac{u}{1 + um_G(z)} dF_{U,V}(u, v)} \quad (48)$$

By comparing this with (28), we get that  $\eta^{\text{mmse}} = m_G(-1)$  and that

$$\int \frac{1}{(\lambda + 1)^2} dG(\lambda) = \left. \frac{d}{dz} m_G(z) \right|_{z=-1} = m'_G(-1)$$

By using this into (47) we obtain

$$\Delta = m'_G(-1) \left( 1 + \alpha \int \frac{v}{(1 + u\eta^{\text{mmse}})^2} dF_{U,V}(u, v) \right) \quad (49)$$

Finally, we apply the implicit function derivative theorem (see [30, Sect. 2.10]) to (48) and get explicitly

$$m'_G(-1) = \left[ \left( 1 + \alpha \int \frac{u}{1 + u\eta^{\text{mmse}}} dF_{U,V}(u, v) \right)^2 - \alpha \int \frac{u^2}{(1 + u\eta^{\text{mmse}})^2} dF_{U,V}(u, v) \right]^{-1}$$

By using this and (49) into (46) we obtain (30), as desired.  $\square$

## A.2 Proof of Proposition 5

Let  $\mathcal{C} \subseteq \mathbb{C}^N$  be a group code generated by an abstract finite Abelian group  $\mathcal{G}$  [29]. Then, there exist a representation  $G$  of  $\mathcal{G}$  in terms of unitary matrices such that  $\mathcal{C}$  is the orbit of any of its code words under the action of  $G$ , i.e., for any  $\mathbf{x}_0 \in \mathcal{C}$  we have

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{C}^N : \mathbf{x} = \mathbf{U}\mathbf{x}_0, \mathbf{U} \in G\}$$

As a consequence of the general structure of Abelian groups, there exists a system of coordinates in which the projection of  $\mathcal{C}$  in any of its components is a PSK constellation [27] (possibly of different sizes and amplitudes) and the elements of  $G$  are diagonal matrices. For an arbitrary  $1 \leq i \leq N$  let  $\mathcal{A}$  be the projection of  $\mathcal{C}$  on the  $i$ -th component (we let  $M = |\mathcal{A}|$ ). Let  $\hat{\mathbf{x}}$  denote the transmitted reference code word and  $\mathbf{y} = \hat{\mathbf{x}} + \boldsymbol{\nu}$  be the corresponding received signal after transmission on a complex circularly-symmetric

AWGN channel with SNR  $\gamma$ . The output pmf for symbol  $i$  produced by the SISO decoder with input  $\mathbf{y}$  is given by

$$P_i(a) \propto \sum_{\substack{\mathbf{x} \in \mathcal{C} \\ x_i = a}} \prod_{j \neq i} \exp(-\gamma |y_j - x_j|^2) \quad (50)$$

Let  $f_{P_i|\hat{\mathbf{x}}}(p_1, \dots, p_M)$  be the joint density of the vector  $\{P_i(a) : a \in \mathcal{A}\}$  induced by the conditional probability measure of  $\mathbf{y}$  given  $\hat{\mathbf{x}}$ . In order to show symbol-by-symbol symmetry we have to show that  $f_{P_i|\hat{\mathbf{x}}}(p_1, \dots, p_M)$  depends only on the  $i$ -th transmitted symbol  $\hat{x}_i$  (uniformity), and that for all  $b, b' \in \mathcal{A}$  there exists a permutation  $\pi$  of the arguments  $p_1, \dots, p_M$  such that

$$f_{P_i|b'}(p_1, \dots, p_M) = f_{P_i|b}(p_{\pi(1)}, \dots, p_{\pi(M)}) \quad (51)$$

Without loss of generality we can assume that  $1 \in \mathcal{A}$ , since because of the circular symmetry of the noise the channel is invariant with respect to a scaling and rotation of the output. Consider the subgroup  $H \subseteq G$  of all matrices  $\mathbf{U}$  having 1 as  $i$ -th element of the diagonal. This induces the coset partition  $G = \cup_{a \in \mathcal{A}} \mathbf{U}_a H$ , where  $\mathbf{U}_a$  is the ‘‘coset leader’’ for the coset of matrices having  $a$  as  $i$ -th diagonal element. Define the subcode  $\mathcal{C}_a^i = \{\mathbf{x} \in \mathcal{C} : x_i = a\}$ . Then,  $\mathcal{C}_a^i = \{\mathbf{x} = \mathbf{U}_a \mathbf{U} \mathbf{x}_0 : \mathbf{U} \in H\}$ . We can rewrite (50) as

$$P_i(a) \propto \sum_{\mathbf{U} \in H} \exp(-\gamma (|\mathbf{y} - \mathbf{U}_a \mathbf{U} \mathbf{x}_0|^2 - |y_i - a|^2)) \quad (52)$$

Let  $\hat{\mathbf{x}} = \hat{\mathbf{U}} \mathbf{x}_0$  and assume that  $\hat{\mathbf{U}} \in \mathbf{U}_b H$ , for some  $b \in \mathcal{A}$  where  $\hat{x}_i = b$  is the symbol transmitted in position  $i$ . For any arbitrary  $\tilde{\mathbf{U}} \in H$  we have

$$\begin{aligned} |\mathbf{y} - \mathbf{U}_a \mathbf{U} \mathbf{x}_0|^2 &= |\boldsymbol{\nu} + \hat{\mathbf{x}} - \mathbf{U}_a \mathbf{U} \mathbf{x}_0|^2 \\ &= \left| \boldsymbol{\nu} + \hat{\mathbf{U}} \tilde{\mathbf{U}}^{-1} \tilde{\mathbf{U}} \mathbf{x}_0 - \mathbf{U}_a \mathbf{U} \mathbf{x}_0 \right|^2 \\ &= \left| \tilde{\mathbf{U}}^{-1} \left( \tilde{\boldsymbol{\nu}} + \hat{\mathbf{U}} \tilde{\mathbf{U}} \mathbf{x}_0 - \mathbf{U}_a \tilde{\mathbf{U}} \mathbf{x}_0 \right) \right|^2 \\ &= \left| \tilde{\boldsymbol{\nu}} + \hat{\mathbf{U}} \tilde{\mathbf{x}} - \mathbf{U}_a \mathbf{U} \tilde{\mathbf{x}} \right|^2 \end{aligned}$$

where  $\tilde{\mathbf{x}} = \tilde{\mathbf{U}} \mathbf{x}_0$  and where  $\tilde{\boldsymbol{\nu}} = \tilde{\mathbf{U}} \boldsymbol{\nu}$  has the same statistics of  $\boldsymbol{\nu}$  since  $\tilde{\mathbf{U}}$  is unitary. As  $\mathbf{U}$  varies in  $H$ ,  $\mathbf{x} = \mathbf{U}_a \mathbf{U} \tilde{\mathbf{x}}$  varies in the subcode  $\mathcal{C}_a^i$ . Moreover, the term  $|y_i - a|^2 = |\nu_i + b - a|^2$  in the exponent of (52) depends only on  $b$  and not on the whole  $\hat{\mathbf{x}}$ . Since  $\tilde{\mathbf{U}}$  is arbitrary, then  $\hat{\mathbf{U}} \tilde{\mathbf{x}}$  is an arbitrary code word in the subcode  $\mathcal{C}_b^i$ . We conclude that  $f_{P_i|\hat{\mathbf{x}}}(p_1, \dots, p_M)$  depends only on  $b$  and not on the particular transmitted code word  $\hat{\mathbf{x}} \in \mathcal{C}_b^i$ , so that  $\mathcal{C}$  is uniform.

Now, for any  $b' \in \mathcal{A}$  consider the matrix  $\mathbf{U}_{b'}\widehat{\mathbf{U}}^{-1}$  belonging to the coset of  $H$  of all matrices having the element  $b'b^*$  in the  $i$ -th diagonal position. We have

$$|\mathbf{y} - \mathbf{U}_a\mathbf{U}\mathbf{x}_0|^2 = \left| \boldsymbol{\nu}' + \mathbf{U}_{b'}\widehat{\mathbf{U}}^{-1}\widehat{\mathbf{x}} - \mathbf{U}_{b'}\widehat{\mathbf{U}}^{-1}\mathbf{U}_a\mathbf{U}\mathbf{x}_0 \right|^2$$

where  $\boldsymbol{\nu}' = \mathbf{U}_{b'}\widehat{\mathbf{U}}^{-1}\boldsymbol{\nu}$  has the same statistics of  $\boldsymbol{\nu}$ , and  $|\nu_i + b - a|^2 = |\nu'_i + b' - (b'b^*)a|^2$ . The ordered sequences of symbols  $\{a : a \in \mathcal{A}\}$  and  $\{b'b^*a : a \in \mathcal{A}\}$  differ only by a cyclic shift, namely, by the rotation that brings  $b$  into  $b'$  (an isomorphism of the generating group of  $\mathcal{A}$  [29]). Also the ordered sequences of cosets  $\{\mathbf{U}_aH : a \in \mathcal{A}\}$  and  $\{\mathbf{U}_{b'}\widehat{\mathbf{U}}^{-1}\mathbf{U}_aH : a \in \mathcal{A}\}$  differ by the same cyclic shift. Therefore, by letting  $\pi$  be the permutation corresponding to that cyclic shift we have immediately that (51) is verified.  $\square$

## References

- [1] P. Alexander and M. Reed. Iterative detection in code-division multiple-access with error control coding. *to appear on European Trans. On Telecomm.*, 2000.
- [2] T. Aulin and R. Espineira. Trellis coded multiple access (TCMA). In *Proc. ICC 1999*, Vancouver, June 1999.
- [3] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. on Inform. Theory*, 20(3):284–287, March 1974.
- [4] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. soft-input soft-output building blocks for the construction of distributed iterative decoding of code networks. *European Trans. on Commun.*, April 1998.
- [5] C. Berrou and A. Glavieux. Near optimum error-correcting coding and decoding: Turbo codes. *IEEE Trans. on Commun.*, 44(10), October 1996.
- [6] J. Boutros and O. Pothier. Convergence analysis of turbo decoding. In *Canadian Workshop on Inform. Theory*, Toronto, June 1997.
- [7] B. Rimoldi and R. Urbanke. A rate splitting approach to the gaussian multiple-access channel. *IEEE Trans. on Inform. Theory*, 42(2):364–375, March 1996.
- [8] L. Brunel. *Algorithmes de decodage de canal pour l' acces multiple à etalement de spectre*. PhD thesis, Ph.D Thesis, ENST Paris, 1999.

- [9] L. Brunel and J. Boutros. Code division multiple access based on independent codes and turbo decoding. *Annales des Télécommunications*, 54(7-8):401–410, July 1999.
- [10] L. Brunel and J. Boutros. Performance limits of CDMA with independent codes and turbo decoding. submitted to IEEE Trans. on Commun., June 2000.
- [11] M. Buehrer, S. Nicoloso, and S. Gollamudi. Linear versus non-linear interference cancellation. *Journal of Communications and Networks*, 1(2):118–133, June 1999.
- [12] G. Caire and E. Biglieri. Linear codes over cyclic groups. *IEEE Trans. on Inform. Theory*, 41(5):1246–1256, September 1995.
- [13] G. Caire, E. Biglieri, and G. Taricco. CDMA system design through asymptotic analysis. to appear on IEEE Trans. on Commun., 2000.
- [14] N. Chayat and S. Shamai. Iterative soft onion peeling for multi-access and broadcast channels. In *Proc. PIMRC'98*, Boston, September 1998.
- [15] N. Chayat and S. Shamai. Convergence properties of iterative soft onion peeling. In *Proc. ITW 1999*, page 9, Kruger national park, South Africa, June 1999.
- [16] T. Cover and J. Thomas. *Elements of information theory*. Wiley, New York, 1991.
- [17] M. Damen. *Joint coding/decoding in a multiple access system: applications to mobile communications*. PhD thesis, Ph.D Thesis, ENST Paris, 1999.
- [18] D. Divsalar, M. Simon, and D. Raphaeli. Improved parallel interference cancellation for CDMA. *IEEE Trans. on Commun.*, 46(2):258–268, February 1998.
- [19] J. Evans and D. Tse. Large system performance of linear multiuser receivers in multipath fading channels. to appear on IEEE Trans. on Inform. Theory, 2000.
- [20] A. Felstrom and K. Zigangirov. Time-varying periodic convolutional codes with low-density parity-check matrix. *IEEE Trans. on Inform. Theory*, 45(6), September 1999.
- [21] B. Frey, F. Kschischang, A. Loeliger, and N. Wiberg. Factor graphs and algorithms. In *Proc. 35th Annual Allerton Conference on Communication, Control and Computing*, pages 666–680, Allerton house, Monticello, IL, October 1997.
- [22] R. Gallager. *Low-density parity check codes*. PhD thesis, Cambridge, M.I.T. Press, 1963.

- [23] H. El Gamal and R. Hammons. Analyzing the turbo decoder using the Gaussian approximation. submitted to IEEE Trans. on Inform. Theory, June 2000.
- [24] T. Giallorenzi and S. Wilson. Multiuser ML sequence estimator for convolutional coded asynchronous DS-CDMA systems. *IEEE Trans. on Commun.*, 44(8):997–1008, August 1996.
- [25] J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary block and convolutional codes. *IEEE Trans. on Inform. Theory*, 42:429–445, 1996.
- [26] N. Ibrahim. *Codage et decodage de canal pour un système de communication à accès multiple*. PhD thesis, Ph.D Thesis, ENST Paris, 1999.
- [27] I. Ingemarsson. Commutative group codes for the Gaussian channel. *IEEE Trans. on Inform. Theory*, (2):215–219, March 1973.
- [28] E. Chong J. Zhang and D. Tse. Output MAI distribution of linear MMSE multiuser receivers in DS-CDMA systems. submitted to IEEE Trans. on Inform. Theory, May 2000.
- [29] D. Forney Jr. Geometrically uniform codes. *IEEE Trans. on Inform. Theory*, 37(5):1241–1260, September 1991.
- [30] W. Kaplan. *Advanced calculus*. Addison-Wesley, Reading, MA, 1987.
- [31] Kiran and D. Tse. Effective interference and effective bandwidth of linear multiuser receivers in asynchronous systems. submitted to IEEE Trans. on Inform. Theory, 1999.
- [32] F. Kschischang and B. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE J. Select. Areas Commun.*, 16(2):219–230, February 1998.
- [33] A. Lampe and J. Huber. On improved multiuser detection with soft decision interference cancellation. In *Proc. ICC 1999, Comm. Theory Mini-Conference*, pages 172–176, Vancouver, June 1999.
- [34] G. Woodward M. Honig and P. Alexander. Adaptive multiuser parallel decision-feedback with iterative decoding. In *Proc. ISIT 2000*, page 335, Sorrento, Italy, June 2000.

- [35] M. Moher. An iterative multiuser decoder for near-capacity communications. *IEEE Trans. on Commun.*, 46(7):870–880, July 1998.
- [36] F. Neeser and J. L. Massey. Proper complex random processes with applications to information theory. *IEEE Trans. on Inform. Theory*, 39(4):1293–1302, July 1993.
- [37] L. Nelson and V. Poor. Iterative multiuser receivers for CDMA channels: an EM-based approach. *IEEE Trans. on Commun.*, 44(12):1700–1710, December 1996.
- [38] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Francisco, 1988.
- [39] J. Proakis. *Digital communications, 3rd Ed.* McGraw-Hill, New York, 1995.
- [40] R. Ratasuk and M. Honig. Large system error probability of multiuser decision feedback receivers. In *Proc. ISIT 2000*, page 385, Sorrento, Italy, June 2000.
- [41] M. Reed, C. Schlegel, P. Alexander, and J. Asenstorfer. Iterative multiuser detection for CDMA with FEC: near single-user performance. *IEEE Trans. on Commun.*, 46(12):1693–1699, December 1998.
- [42] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message passing decoding. submitted to IEEE Trans. on Inform. Theory, 1998.
- [43] T. Richardson and R. Urbanke. An introduction to the analysis of iterative coding systems. IMA Proceedings, 2000.
- [44] C. Schlegel. Joint detection in multiuser systems via iterative processing. In *Proc. ISIT 2000*, page 274, Sorrento, Italy, June 2000.
- [45] R.M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. on Inform. Theory*, 27(5), 1981.
- [46] F. Tarkoy. Iterative multiuser decoding for asynchronous users. In *Proc. ISIT '97*, page 30, Ulm, Germany, July 1997.
- [47] D. Tse and S. Hanly. Linear multiuser receivers: Effective interference, effective bandwidth and capacity. *IEEE Trans. on Inform. Theory*, 45(2):641–675, March 1999.

- [48] M. Varanasi and T. Guess. Optimum decision feedback multiuser equalization with successive decoding achieves the total capacity of the Gaussian multiple access channel. In *Proc. Asilomar Conference*, Pacific Groove, CA, November 1997.
- [49] S. Verdu. *Multiuser detection*. Cambridge University Press, Cambridge, UK, 1998.
- [50] S. Verdu and S. Shamai. The effect of frequency-flat fading on the spectral efficiency of CDMA. submitted to IEEE Trans. on Inform. Theory, November 1999.
- [51] S. Verdu and S. Shamai. Spectral efficiency of CDMA with random spreading. *IEEE Trans. on Inform. Theory*, 45(2):622–640, March 1999.
- [52] S. Vialle and J. Boutros. Performance limits of concatenated codes with iterative decoding. In *Proc. ISIT 2000*, page 150, Sorrento, Italy, June 1997.
- [53] X. Wang and V. Poor. Iterative (Turbo) soft interference cancellation and decoding for coded CDMA. *IEEE Trans. on Commun.*, 47(7):1047–1061, July 1999.
- [54] N. Wiberg, H. Loeliger, and R. Kotter. Codes and iterative decoding on general graphs. *European Trans. on Telecomm.*, 6, September 1995.
- [55] Sae Young Chung, T. Richardson, and R. Urbanke. Analysis of sum-product decoding of low-density parity-check codes using Gaussian approximation. submitted to IEEE Trans. on Inform. Theory, 2000.

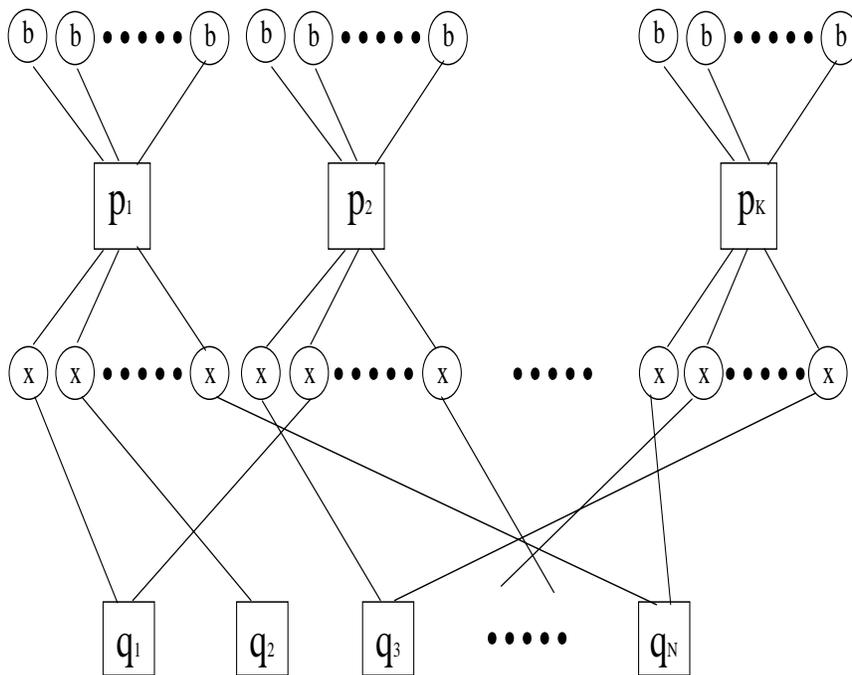


Figure 1: Factor graph for the input-output joint pdf of the multiuser coded channel. The received signal vectors  $\mathbf{y}[n]$  are “hidden” in the channel transition function nodes  $q_n$ .

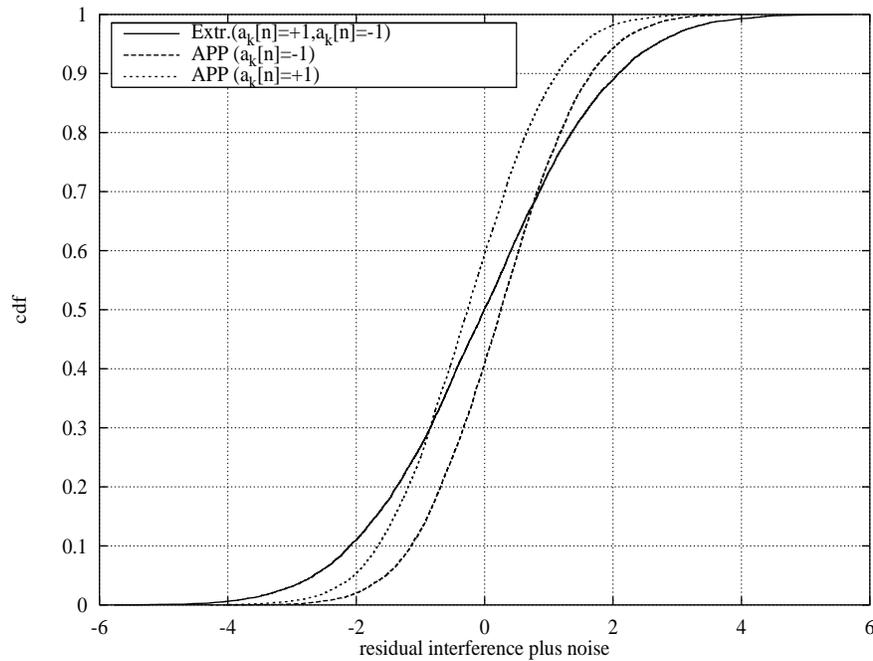


Figure 2: Conditional cdf of the residual interference  $\text{Re}\{\zeta_{k,n}\}$  given  $a_k[n] = +1$  and  $a_k[n] = -1$  after one iteration of the s-PIC-SUMF with BPSK, (5,7) binary convolutional code and  $E_b/N_0 = 6$  dB for all users, for a system randomly generated spreading sequences of length  $L = 20$  and  $K = 40$  users, and for symbol estimates obtained by APPs or by extrinsic pmfs.

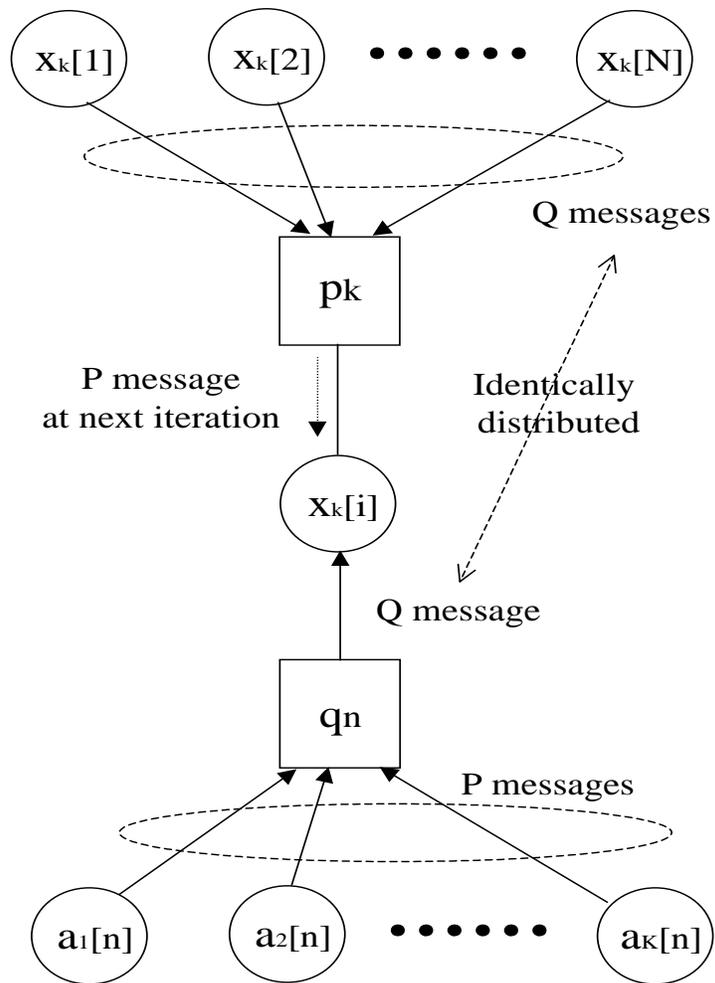


Figure 3: Local neighborhood in the infinite factor graph for visualizing density evolution.

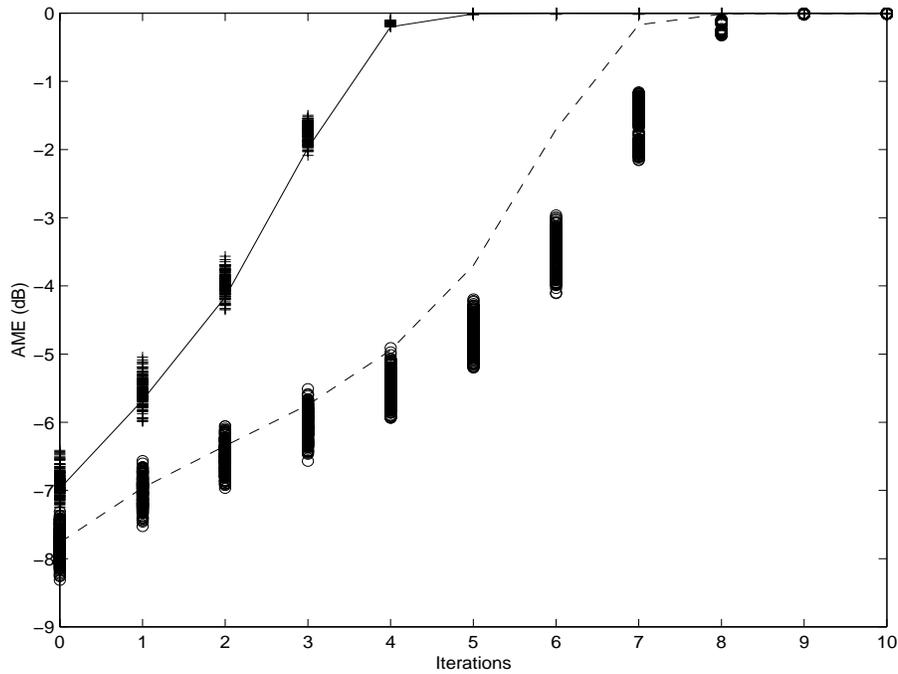


Figure 4: AME vs. number of iterations for s-PIC-SUMF, BPSK, (5,7) binary convolutional code and  $E_b/N_0 = 6$  dB. Points “+” and “o” refer to finite-dimensional systems with  $L = 32$  and  $K = 64$  and 80 users, respectively. The AME trajectory of the corresponding infinite-dimensional systems for  $\alpha = 2.0$  and 2.5 obtained by DE are indicated by the solid and the dashed lines, respectively.

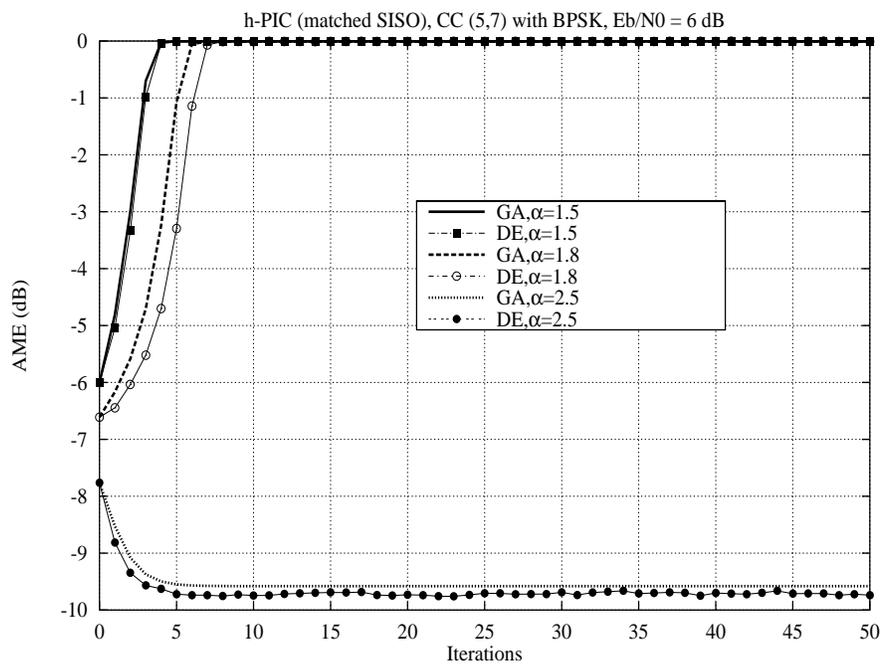


Figure 5: AME vs. number of iterations for h-PIC (with matched SISO) with BPSK, (5,7) binary convolutional code and  $E_b/N_0 = 6$  dB, obtained by DE and by GA.

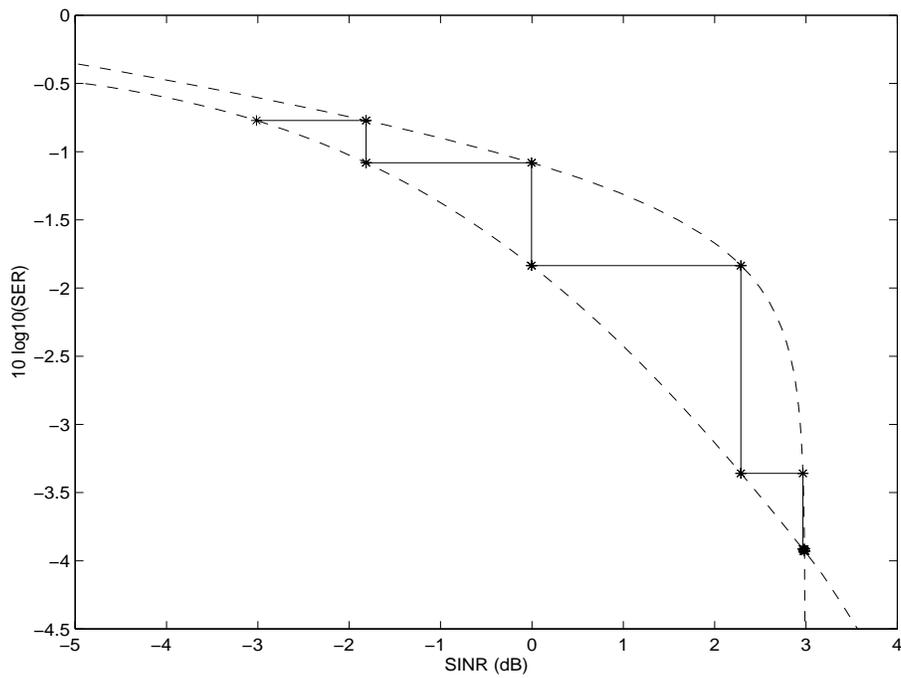


Figure 6: Trajectory of the h-PIC (obtained by GA) in the SINR-SER plane, for  $\alpha = 1.5$  and the same conditions of Fig. 5. The upper and lower dashed lines indicate the code SER and the hard IC SINR curves, respectively.

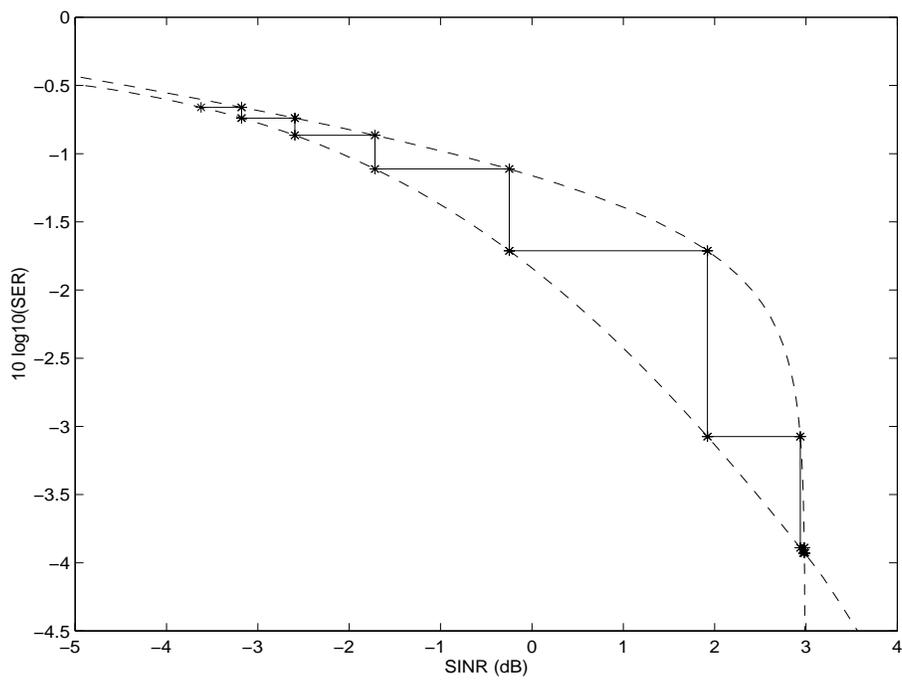


Figure 7: Trajectory of the h-PIC (obtained by GA) in the SINR-SER plane, for  $\alpha = 1.8$  and the same conditions of Fig. 5. The upper and lower dashed lines indicate the code SER and the hard IC SINR curves, respectively.

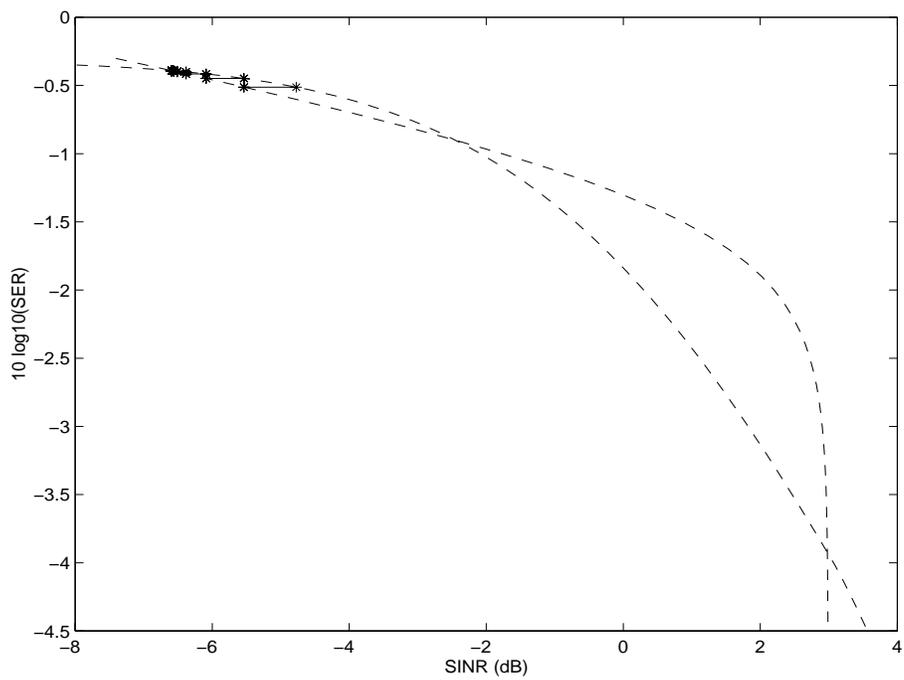


Figure 8: Trajectory of the h-PIC (obtained by GA) in the SINR-SER plane, for  $\alpha = 2.5$  and the same conditions of Fig. 5. The upper and lower dashed lines indicate the code SER and the hard IC SINR curves, respectively.

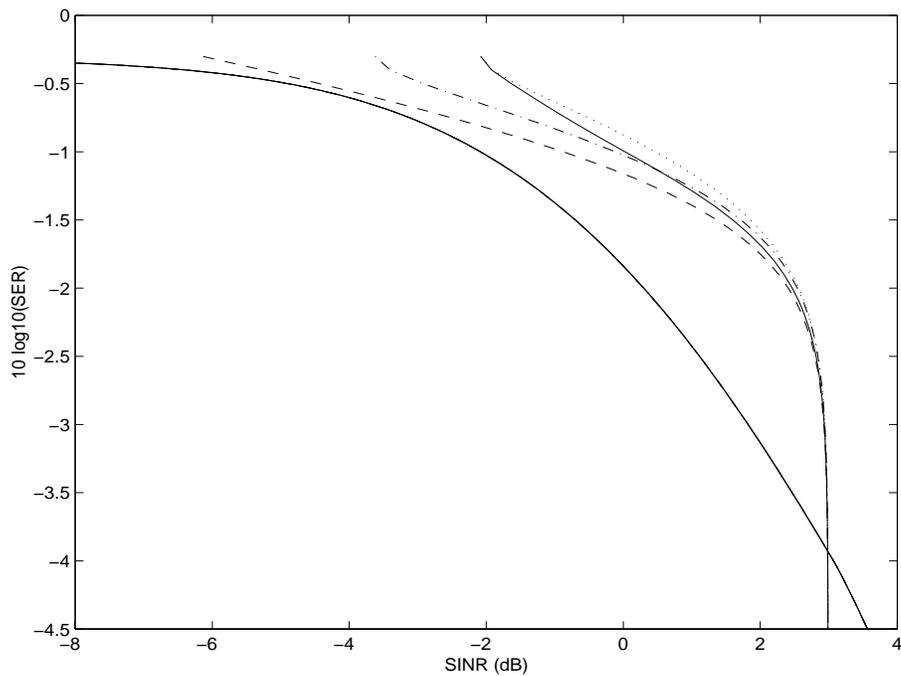


Figure 9: SER curve of the (5,7) convolutional code (lower solid line) and SINR curves for h-PIC (dash-dotted line), s-PIC-SUMF (dashed line), s-PIC-LMMSE (upper solid line) and s-PIC-LMMSE with ideally matched LMMSE filters (dotted line) for  $E_b/N_0 = 6$  dB and  $\alpha = 1.8$ . All the SINR curves are obtained by the GA.

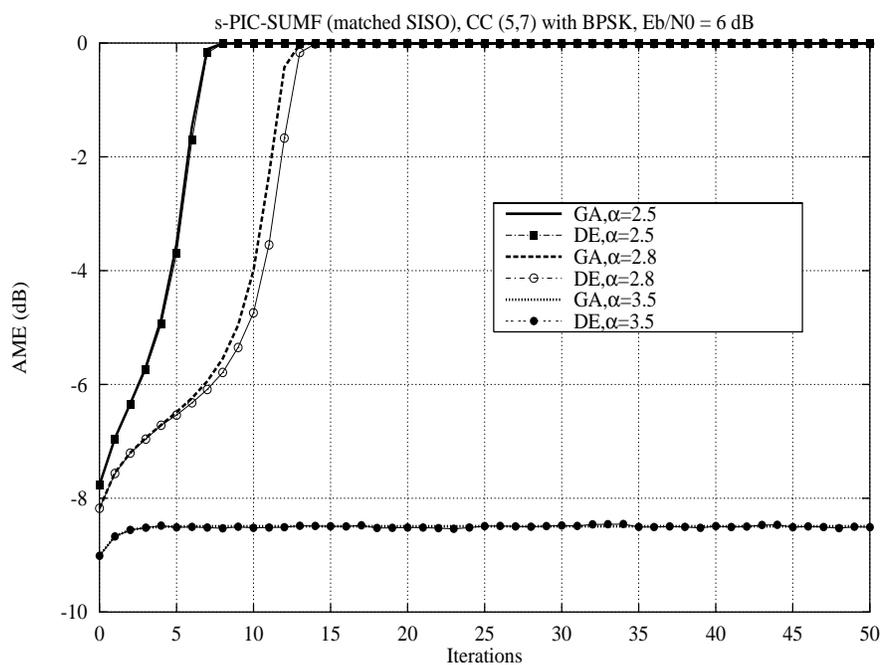


Figure 10: AME vs. number of iterations for s-PIC-SUMF (with matched SISO) with BPSK, (5,7) binary convolutional code and  $E_b/N_0 = 6$  dB, obtained by DE and by GA.

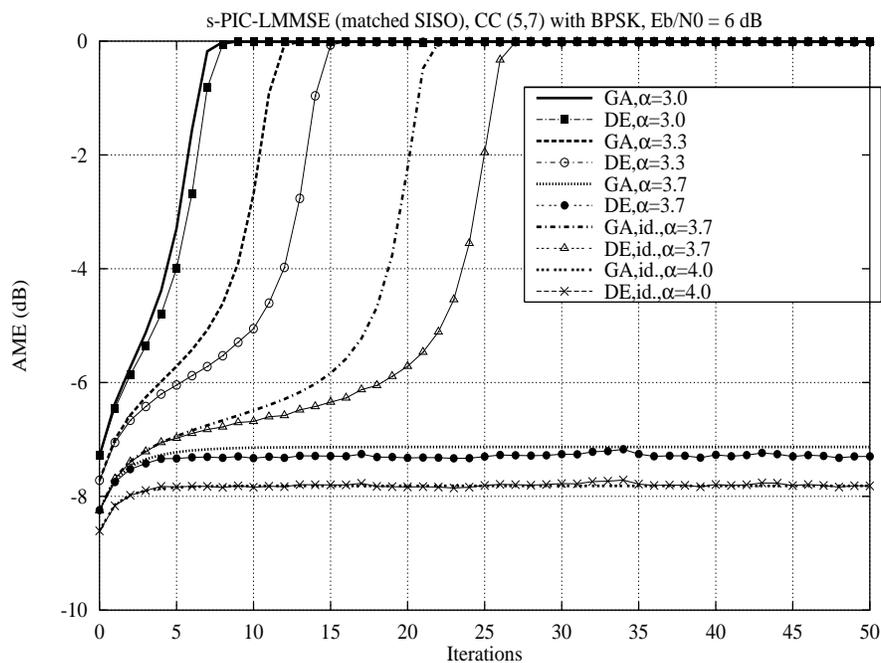


Figure 11: AME vs. number of iterations for s-PIC-LMMSE (with matched SIS0) and s-PIC-LMMSE with ideally matched LMMSE filters (denoted by “id.”) with BPSK, (5,7) binary convolutional code and  $E_b/N_0 = 6$  dB, obtained by DE and by GA.

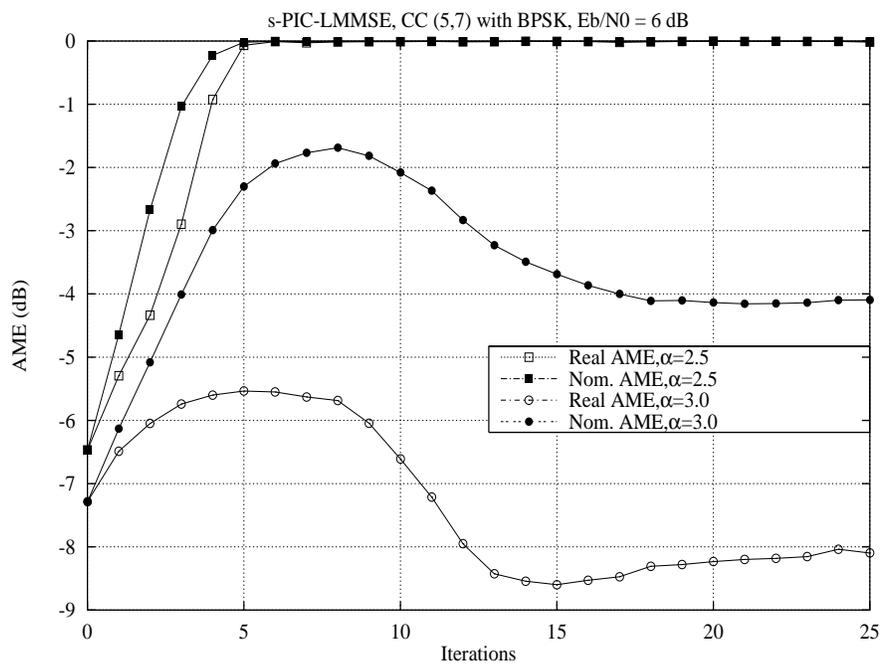


Figure 12: True and nominal AME vs. number of iterations for s-PIC-MMSE (mismatched SISO), BPSK, (5,7) binary convolutional code and  $E_b/N_0 = 6$  dB.

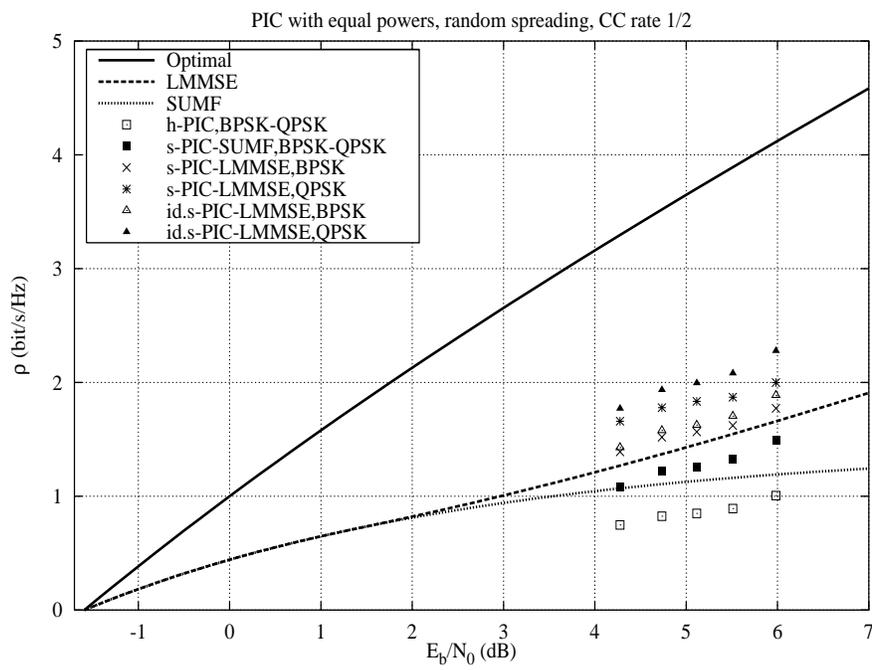


Figure 13: Spectral efficiency of iterative PIC decoders with equal power users and convolutional codes of rate 1/2, BPSK and QPSK modulation.

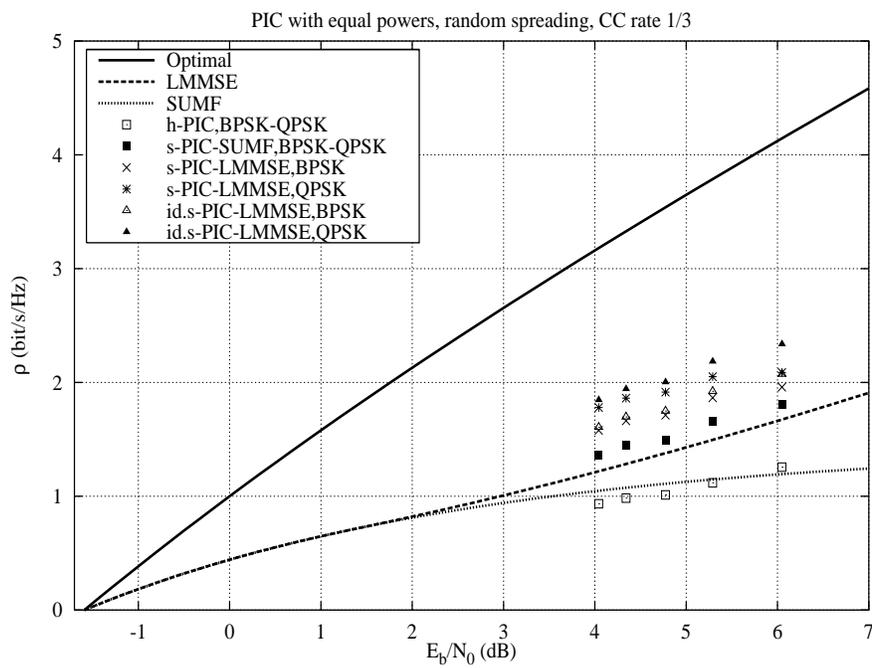


Figure 14: Spectral efficiency of iterative PIC decoders with equal power users and convolutional codes of rate 1/3, BPSK and QPSK modulation.

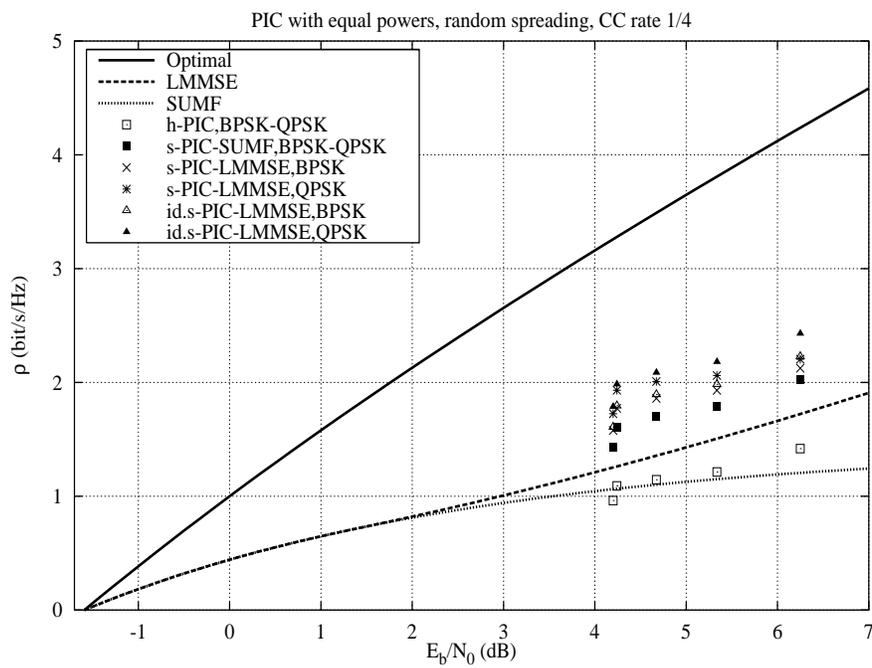


Figure 15: Spectral efficiency of iterative PIC decoders with equal power users and convolutional codes of rate 1/4, BPSK and QPSK modulation.