Iterative multi-user detection and decoding for turbo-coded DS-CDMA systems

Sandrine Vaton, Thierry Chonavel and Samir Saoudi
ENST-Bretagne, BP 832, 29285 BREST Cedex, FRANCE
email: {sandrine.vaton, thierry.chonavel, samir.saoudi}@enst-bretagne.fr

ABSTRACT

In this paper we consider turbo-coded multi-user DS-CDMA communications. We first present a new scheme for Soft-Input Soft-Output multi-user detection. Our method provides the Maximum A Posteriori estimate for each user bit together with its fiability in the form of an extrinsic information. It is based on the stack algorithm over the tree diagram of the CDMA channel. The Soft-Input Soft-Output multi-user detector is then associated to a bank of turbo-decoders in a global turbo loop to perform iteratively multi-user detection and turbo-decoding. Simulation results show the high efficiency of the proposed method.

1 Introduction

The turbo principle was first introduced in [4] for the iterative decoding of parallel concatenated convolutional codes; it has been generalized to different communication problems such as the decoding of serially concatenated codes, joint source and channel coding, turbo interference reduction or turbo equalization.

Recently some authors have considered the application of the turbo principle to the separation and decoding of K coded users that communicate over a CDMA channel [8][2][10][12]. Basically, the solution consists in considering the CDMA channel as a code, and in applying the principles of the turbo decoding of serially concatenated convolutional codes [3].

The problem of Maximum Likelihood (ML) estimation of the information bits is broken into two "simple" ones: (i) Soft Input Soft Output (SISO) Multiple Access Interference suppression, and (ii) SISO decoding. The two modules communicate iteratively in a global loop by exchanging a "global" extrinsic information over information bits and coded bits, until no substantial amelioration is provided by an additional iteration. The SISO decoder itself is turbo since the code considered in this paper is the turbo code normalized in the UMTS [1]; the turbo code consists in the parallel association of two elementary 8-states convolutional codes. Therefore, in the global loop, one pass of the SISO decoder can be decomposed into several passes in the local turbo decoding loop. In the local loop a local extrinsic information over the information bits is exchanged between the two component SISO 8-states trellis decoders.

In Section 2, we present a suboptimal Maximum A Posteriori multiuser detector with soft output. Our subMAP detector is based on the stack algorithm [7] over the tree diagram of the CDMA channel. The stack algorithm presents two advantages (i) it provides the exact ML estimate and (ii) its numerical complexity is linear with the number of users when the Signal to Noise Ratio (SNR) is high. In Section 3 we integrate in the branch metrics of the tree diagram a Bayesian prior over the user bits; this Bayesian prior is in practice the global extrinsic information provided by the bank of turbo decoders. In Section 3 we also explain how one computes the extrinsic information provided by the SISO detector. In Section 4, we discuss the association of the multiuser detector with the bank of single user turbo decoders. Simulation results are presented in Section 5.

2 Multiuser detection based on the tree diagram

Recall that the model of a CDMA channel with additive white Gaussian noise is the following:

$$\mathbf{r} = \mathbf{S}\mathbf{b} + \mathbf{n}, \tag{1}$$

where $\mathbf{b} = [b_1, \ldots, b_K]^T$ is the vector of users bits, $\mathbf{S} = [\mathbf{s_1}, \ldots, \mathbf{s_K}]$ is the matrix of normalized signatures, and $\mathbf{n}$ is a zero mean Gaussian vector with independent components: $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$.

Clearly, the problem of ML detection on the additive white Gaussian noise channel is equivalent to minimizing $\parallel \mathbf{S}\mathbf{b} - \mathbf{r} \parallel^2$:

$$\hat{\mathbf{b}} = \text{Arg} \min_{\mathbf{b} \in \{\pm 1\}^{\mathbf{K}}} \parallel \mathbf{S}\mathbf{b} - \mathbf{r} \parallel^2 . \tag{2}$$

Consider the QR decomposition of $\mathbf{S}$: $\mathbf{S} = \mathbf{Q}\mathbf{R}$ where $\mathbf{R}$ is an upper triangular matrix and where $\mathbf{Q}$ is a unitary matrix, that is to say $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. It can be shown that minimizing $\parallel \mathbf{S}\mathbf{b} - \mathbf{r} \parallel^2$ amounts to minimizing $\parallel \mathbf{R}\mathbf{b} - \mathbf{z} \parallel^2$, where $\mathbf{z} = \mathbf{Q}^\mathbf{T}\mathbf{r}$ is an exhaustive statistics for the problem. The upper triangular structure of $\mathbf{R}$ makes it possible to solve this problem as the detection of a shortest path in a tree diagram of the form displayed on Figure 1. The branch metrics over the tree diagram are:

$$\text{branch } (b_K, b_{K-1}, \cdots, b_k) \leftarrow [(\mathbf{R}\mathbf{b})_k - z_k]^2 \tag{3}$$

where $(\mathbf{Rb})_k = \sum_{i=k,K} R(k,i) b_i$ is the component number k of vector $\mathbf{Rb}$.

Recently, we have proposed to use the stack algorithm to solve this problem [13]. The stack algorithm (Zigangirov, 1966 [14]; Jelinek, 1969 [7]) keeps track of a few paths and their corresponding metrics in a stack; the head of the stack is always the path with the shortest metric among all the paths in the stack. At each step of the algorithm, the path at the head of the stack is extended by one branch, thus yielding two successors; the two successors along with the other paths in the stack are reordered, so that the head of the stack is always the path with the shortest metric. The process of extending the path is then repeated until a complete path with K branchs is found.

The stack algorithm has the advantage to produce the exact ML solution, contrary to the other methods over the tree diagram such as the Fano algorithm or the feedback decoding algorithm which are suboptimal [9]. What is more, for high SNRs, the number of branch metrics computed per user is equal to 2, leading to a linear complexity with the number of users. At a high SNRs, the stack algorithm never turns back over the tree diagram; the process of selecting the best successor among the two successors of the head of the stack always selects the true path because this path is much more probable that the other paths in the stack.

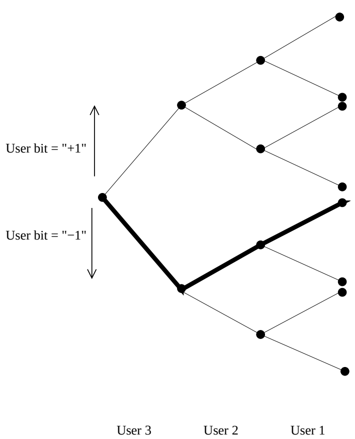

Figure 1: Tree diagram for $K = 3$ users. The bold path is $\mathbf{b} = (b_1, b_2, b_3)^T = (+1, +1, -1)^T$.

3 SubMAP detection over the tree diagram

In order to perform iteratively the multiuser detection and the decoding in a turbo manner, we must adapt the detection algorithm (i) to account for a Bayesian prior over the user bits (this information is supplied by the output of the turbo decoders at the previous iteration) (ii) and to supply a soft decision in form of an extrinsic information that will be used as the input of the bank of turbo-decoders.

3.1 Branch metrics update with Bayesian prior

Suppose that there is a Bayesian prior over b_k:

$$\log p(b_k = \pm 1; Z_k^{\text{in}}) = \text{sign}(b_k)\ Z_k^{\text{in}} + \text{constant}, \qquad (4)$$

Then the complete log-likelihood of $\log p(\mathbf{b} \mid \mathbf{z}; \mathbf{Z}^{\text{in}})$ is, up to an additive constant term:

$$\begin{aligned}\log p(\mathbf{b} \mid \mathbf{z}; \mathbf{Z}^{\text{in}}) \quad = \quad & \textstyle\sum_{k=K}^{1} \text{sign}(b_k)\ Z_k^{\text{in}} \\ & -(2\sigma^2)^{-1} \textstyle\sum_{k=K}^{1} ((\mathbf{Rb})_k - z_k)^2\end{aligned} \qquad (5)$$

This is the cumulated metrics of the path $\mathbf{b}$ where the branch metrics are defined as:

$$\text{branch}\ \ (b_K, b_{K-1}, \cdots, b_k) \leftarrow \text{sign}(b_k) Z_k^{\text{in}} - \frac{1}{2\sigma^2}[(\mathbf{Rb})_k - z_k]^2. \qquad (6)$$

where $\mathbf{Z}^{\text{in}} = (Z_1^{\text{in}}, \cdots, Z_K^{\text{in}})$ is the parameter of the distributions.

3.2 Extrinsic information computation

The a posteriori log-likelihood ratio of b_k is:

$$\text{LLR}(b_k \mid \mathbf{z}) = \log \frac{p(b_k = +1 \mid \mathbf{z}; \mathbf{Z}^{\text{in}})}{p(b_k = -1 \mid \mathbf{z}; \mathbf{Z}^{\text{in}})}. \qquad (7)$$

For computational reasons it is approximated as:

$$\begin{aligned}\text{LLR}(b_k \mid \mathbf{z}; \mathbf{Z}^{\text{in}}) \quad & \simeq \max\nolimits_{\mathbf{b}/b_k=+1} \log p(\mathbf{b}, \mathbf{z}; \mathbf{Z}^{\text{in}}) \\ & - \max\nolimits_{\mathbf{b}/b_k=-1} \log p(\mathbf{b}, \mathbf{z}; \mathbf{Z}^{\text{in}}).\end{aligned} \qquad (8)$$

This suboptimal approach considers only the best path such that $b_k = +1$, and the best path such that $b_k = -1$. This approximation is valid at high SNRs. It is conventional to use suboptimal components in turbo loops since the iterations have the potentiality to correct these suboptimalities [11]. Note that in Eq. (8) the joint log-likelihood $\log p(\mathbf{b}, \mathbf{z})$ is the cumulated metrics of the path $\mathbf{b}$ in the tree diagram with the branch metrics defined by (6).

Suppose first of all that $k = K$, that is to say that the user considered corresponds to the root of the tree displayed on Figure 1. The half upper part of the tree corresponds to $b_K = +1$ whereas the half lower part of the tree corresponds to $b_K = -1$.

The most probable path such that $b_K = +1$ is then obtained by application of the stack algorithm to the tree reduced to its half upper part. This reduction can be performed, for example, by setting the metrics of the branch $b_K = -1$ to minus infinity. Similarly, the most probable path such that $b_K = +1$ is obtained with the stack algorithm over the tree reduced to its half lower part.

Suppose now that $k < K$ so that the user k is not at the root of the tree. We propose the following strategy: first, we reorder the users in vector $\mathbf{b}$ so that bit b_k be at the root; after reordering,

$$\mathbf{b} = (b_1, \cdots, b_{k-1}, b_{k+1}, \cdots, b_K, b_k)^T. \qquad (9)$$

The same permutation should be applied on the vectors $\mathbf{z}$ and $\mathbf{Z}^{\text{in}}$ as well as on the columns of the matrix $\mathbf{R}$.

The permutation of the columns of $\mathbf{R}$ produces one subdiagonal element on columns $k, k+1, \cdots, K-1$. With subdiagonal elements the tree representation of the problem is lost.

The subdiagonal elements are therefore eliminated by means of Givens rotations [6]. The first rotation combines the line $K-1$ and the line K of the matrix $\mathbf{R}$ to eliminate the subdiagonal element $R(K, K-1)$ of column $K-1$. The second rotation combines the lines $K-2$ and $K-1$ of the matrix $\mathbf{R}$ to eliminate the subdiagonal element $R(K-1, K-2)$ on column $K-2$, and so on. The same Givens rotations are applied to the vector $\mathbf{z}$.

Givens rotations are unitary transforms. Therefore, the quantity $||\mathbf{z} - \mathbf{R}\mathbf{b}|\,|^2$ is not modified by the set of permutations and Givens rotations. The complete log-likelihood $\log p(\mathbf{b}, \mathbf{z}; \mathbf{Z}^{\text{in}})$ is the cumulated metrics of the path $\mathbf{b}$ in the reordered tree diagram. And for b_k the computation of the a posteriori log-likelihood ratio (7) can be done easily from the reordered tree diagram.

The global extrinsic information produced by the SISO detector is obtained as:

$$Z_k^{\text{out}} = 0.5\,\text{LLR}(b_k \mid \mathbf{z}; \mathbf{Z}^{\text{in}}) - Z_k^{\text{in}}. \tag{10}$$

The extrinsic information represents the distribution of bit b_k given the soft channel values $\mathbf{z}$, and given the Bayesian priors over bits $b_i, i \neq k$ with an equiprobable prior over bit b_k:

$$Z_k^{\text{out}} = 0.5\,\log \frac{p(b_k = +1 \mid \mathbf{z}; \mathbf{Z}_{-k}^{\text{in}})}{p(b_k = -1 \mid \mathbf{z}; \mathbf{Z}_{-k}^{\text{in}})}. \tag{11}$$

where $\mathbf{Z}_{-k}^{\text{in}} = (Z_1^{\text{in}}, \cdots, Z_{k-1}^{\text{in}}, 0, Z_{k+1}^{\text{in}}, \cdots, Z_K^{\text{in}})$. As always, the Bayesian prior Z_k^{in} over b_k is not taken into account in the output extrinsic information Z_k^{out} of the same bit. This makes it possible to avoid too strong correlations between the successive iterations of the global turbo process. Too much correlation between the iterations could indeed lead to a fixed point that could be a local extremum of the likelihood.

4 Turbo decoding and the global extrinsic information passing

In a very similar way to the turbo decoding of serially concatenated codes, the extrinsic informations Z_k^{out} produced by the detector are the only inputs of the bank of turbo decoders. These quantities are demultiplexed at the output of the detector, so that each turbo decoder has as inputs the extrinsic informations relative to the information bits and parities of the concerned user (see Fig. 3).

The turbo decoding is performed as if the extrinsic informations Z_k^{out} were soft channel values over an additive white Gaussian noise channel. Indeed if $z = b + n$ where $b = \pm 1$ and where n is a Gaussian random variable with mean 0 and variance σ^2, then $\log[p(b_k = +1 \mid z)/p(b_k = -1 \mid z)] = 2/\sigma^2 z$. z is therefore homogeneous to half the log-likelihood ratio. The turbo decoding loop is displayed on Fig. 2. Each component decoder generates a local extrinsic information relatively to the information bits only. This extrinsic information is exchanged iteratively between the two component SISO decoders inside the local loop. At the last iteration of turbo decoding, the turbo decoder provides a global extrinsic information relatively to both the information bits and the

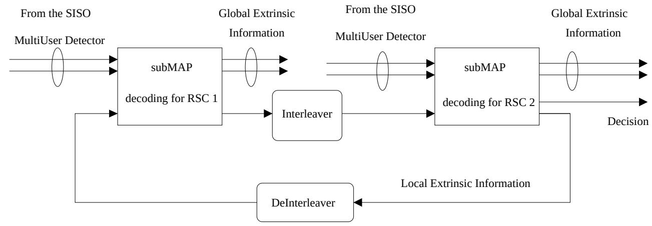

Figure 2: Local Extrinsic Information passing.

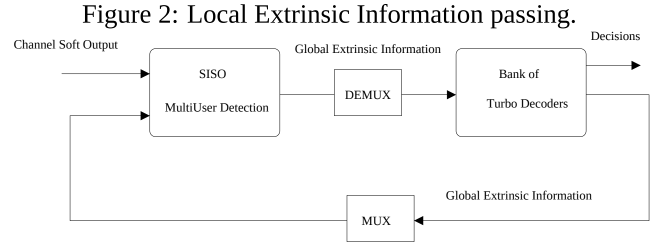

Figure 3: Global Extrinsic Information passing.

parities. This information is globally extrinsic because, if it concerns bit b_k, it does not take into account the input of the turbo decoder relatively to b_k but only the inputs relatively to bits $b_i, i \neq k$. The global extrinsic information produced by the turbo decoder is then one of the inputs of the detector, together with the channel values, at the next iteration of the global loop.

5 Simulation results

For computational issues we consider the case of $K = 9$ users with $N_c = 20$ chips. The signatures are Gold sequences of length 63, truncated at $N_c = 20$.

5.1 Tree decoding algorithm

We first simulate the performances of the (hard output) ML detector based on the stack algorithm over the tree diagram. Figure 4 plots the BER obtained with the stack algorithm, the feedback decoding algorithm [13], which implements approximate ML detection by means of a sliding window of length L (here, $L = 3, L = 2, L = 1$), the decorrelator detector, as well as the single user bound (BPSK). At high SNRs the stack algorithm achieves the single user bound, unlike other techniques. For example, a gain of 3.57 dB is obtained at a BER of 10^{-5} if one replaces the standard decorrelator by the stack algorithm. The average number of branch metrics computed to go through the tree is displayed on Figure 5 for different SNRs. At high SNRs the number of branch metrics computed is equal to 2 per user.

5.2 Iterative SISO detection/turbo-decoding

We consider now the iterative association of the subMAP multiuser detector to a bank of turbo decoders as displayed on Fig. 3. Within each global loop, 5 iterations of turbo decoding are performed. Moreover, we consider up to 5 global iterations. The normalized UMTS turbo code with 640 information bits is used; this turbo code has a free distance $d_f = 26$ [5]. In the decoding the SNR is not known and

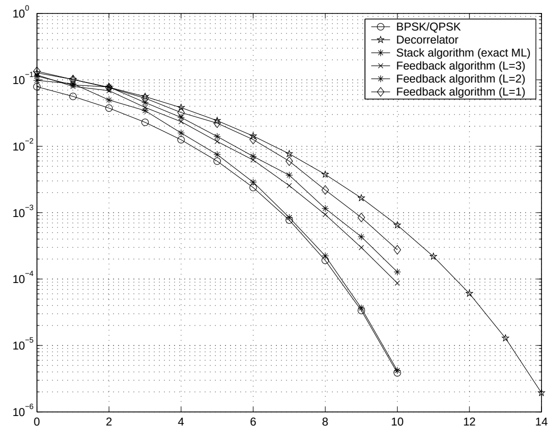

Figure 4: Stack algorithm, feedback decoding algorithm ($L = 3$,$L = 2$ and $L = 1$), decorrelator. $K = 9$ users, Gold sequences of length 63 truncated at $N_c = 20$.

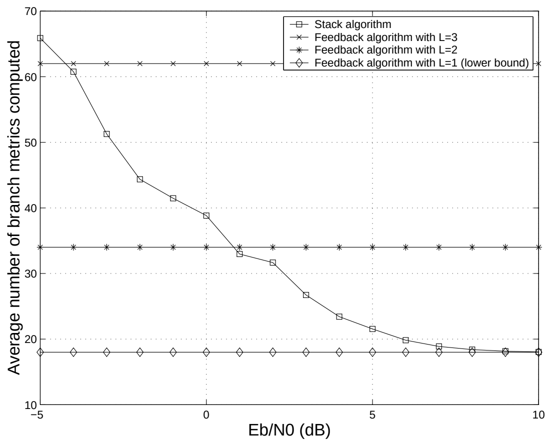

Figure 5: Average number of branch metrics computed per block of K users. $K = 9$ users, Gold sequences of length 63 truncated at $N_c = 20$.

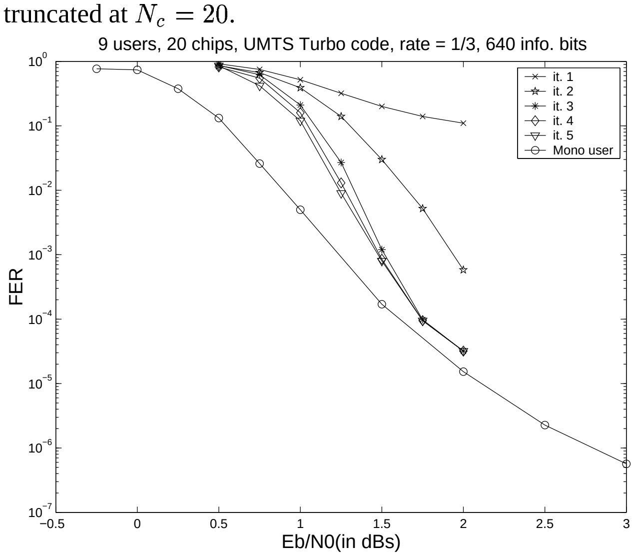

Figure 6: $K = 9$ users, $N_c = 20$ chips, UMTS Turbo Code (rate = 1/3), 640 information bits per user, 5 iterations of turbo decoding per global iteration, 5 global iterations.

we set $\sigma^2 = 1$ in the branch metrics. The SISO component decoders are max log MAP decoders [11]. The Frame Error Rate after each global iteration is displayed on Fig. 6. Convergence is obtained after only 3 iterations; this can easily be explained by the quasi optimality of the subMAP detector, and of the turbo decoder (with 5 iterations). A residual gap of about 0.1 dB is observed between the single user bound and the multi user performance.

6 Conclusion

In this paper we have proposed a new SISO multiuser detection structure of the subMAP type, and we have associated it to a turbo-decoder in an iterative global detection/decoding loop. We have checked by simulations the high efficiency of the proposed structure.

References

[1] Spreading and modulation (FDD). Technical report, Third Generation Partnership Project (3GPP), TS 25.213 v2.3.0, 1999-9.

[2] P. Alexander, M. Reed, J. Asenstorfer, and C. Schlegel. Iterative multiuser interference reduction : Turbo CDMA. *IEEE Trans. on Comm.*, 47:1008–1014, july 1999.

[3] S. Benedetto and G. Montorsi. Iterative decoding of serially concatenated convolutional codes. *Electronic Letters*, 32:1186–1188, june 1996.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding : Turbo Codes. In *Int. Conf. on Comm.*, pages 1064–1070, 1993.

[5] C. Berrou and S. Vaton. Computing the minimum distance of linear codes by the error impulse method. In *Submitted to ITC 2002*.

[6] G. H. Golub and C. F. V. Loan. *Matrix Computation - Second Edition*. John Hopkins Series in Mathematical Sciences, John Hopkins University Press, 1990.

[7] F. Jelinek. Fast sequential decoding algorithm using a stack. *IBM Jour. Res. Dev.*, 13:675–685, November 1969.

[8] M. Moher. An iterative multiuser decoder for near-capacity communications. *IEEE Trans. on Comm.*, 46:870–880, july 1998.

[9] J. Proakis. *Digital Communications.* McGraw-Hill Higher Education, 2001.

[10] Z. Qin, K. Teh, and E. Gunawan. Iterative multiuser detection for asynchronous CDMA with concatenated convolutional coding. *IEEE Journ. on Selected Areas in Comm.*, 19(9):1784–1791, 2001.

[11] P. Robertson, E. Villebrun, and P. Hoeher. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. In *Int. Conf. on Comm.*, pages 1009–1013, Seattle, june 1995.

[12] A. Tarable, G. Montorsi, and S. Benedetto. A linear front end for iterative soft interference cancellation and decoding in coded CDMA. In *IEEE Int. Conf. on Comm.*, volume 1, pages 1–5, 2001.

[13] S. Vaton, T. Chonavel, and S. Saoudi. Approximate and exact ML detectors for CDMA and MIMO systems: a tree detection approach. *Submitted to IEEE Comm. Letters*.

[14] K. Zigangirov. Some sequential decoding procedures. *Probl. Peredach. Inform.*, 2:13–25, 1966.