ITERATIVE MULTIUSER DETECTION IN UMTS WIDEBAND CDMA

Francisco-Javier Escribano-Aparicio† and Manel Martínez-Ramón‡ and Antonio Artés-Rodríguez‡

†Departamento de teoría de la Señal y Comunicaciones, Universidad de Alcalá de Henares, Escuela Politécnica, 28871, Alcalá de Henares - Madrid. e-mail fjavier@tsc.uc3m.es; antonio@ieee.org
‡Departamento de Tecnologías de las Comunicaciones, Universidad Carlos III de Madrid, Avda. de la Universidad, 30, 28911 Leganés - Madrid. e-mail manel@tsc.uc3m.es

ABSTRACT

In this paper we propose a turbodecoding based Multiuser Detection (MUD) technique for an asynchronous Code Division Multiple Access (CDMA) environment where the users can employ different transmission rates. In particular, we apply this iterative method to the Universal Mobile Telecommunications System Wideband CDMA (UMTS W-CDMA) standard, and show through simulation that bit error rates (BER) very near the single user bound can be achieved.

1. INTRODUCTION.

The European standard for third generation mobile systems, known as Universal Mobile Telecommunications System (UMTS), has chosen a Wideband Code Division Multiple Access (W-CDMA) standard which allows intercell asynchronous operation, coherent detection (due to the addition of a pilot channel associated with each data channel), and a high data rate flexibility (arising form the employment of Orthogonal Variable Spreading Factor (OVSF) coding sequences [1]).

The limitation in the number of active users in a cell using code division stems not only from the finite number of available spreading sequences with low crosscorrelation, but also from the degree of interference given by the ensemble of transmitting mobile stations (MS). The larger the number of active stations, the lower the allowable transmitting power for each of them, thus the performance degrades until a limit is reached and subsequent transmissions are rejected. For this reason, it would be interesting to develop detection techniques providing high coding gain.

In particular, multiuser detection (MUD) based on maximum likelihood estimation for CDMA, as proposed in [2], has shown to be useful to improve CDMA performance when additionally combining the probabilistic information drawn from the channel coding [3]. A promising extension of this method involves the application of the turbodecoding principle [4]. It has been shown by simulation that this framework, both for synchronous [5] and asynchronous [6] Direct Sequence-CDMA (DS-CDMA), can lead to performances very close to the single user bound.

Here we apply the above mentioned MUD iterative technique in the uplink of a UMTS W-CDMA environment with asynchronous operation (Frequency Division Duplex (FDD) mode). In our approach we basically make use of the framework developed in [6], but with some major modifications to apply it to the access proposed in UMTS. In the next section we describe the system, while in the third one the theoretical basis for the MUD iterative technique is briefly reviewed and we explain with some detail the extension of the asynchronous MUD detection algorithm in [6] to the case where a high data rate flexibility is needed. The fourth section is devoted to the simulation results. Finally, some conclusions are drawn.

2. SYSTEM DESCRIPTION.

The spreading for the uplink dedicated physical channels in UMTS W-CDMA is shown in Figure 1 [7]. The Dedicated Physical Data Channel (DPDCH) and the Dedicated Physical Control Channel (DPCCH) are mapped to the I and Q branches. Both channels are first spread to the chip rate, 4096 Kcps, with two OVSF codes (c0, c1), whose spreading factors vary from 4 to 256. The allowable rates for each channel are 16 to 1024 Kbps. Both channels are complex spread with a MS specific scrambling code, which provides no gain,
but identifies the user. The scrambling code
\[ c_{sc} = c_I + j c_Q \]
is composed of two different codes \((c_I, c_Q)\) from the extended Very Large Kasami set of length 256 \([8]\). In multi-code transmission, the additional DPDCH's are transmitted on either the I or the Q branch.

![Diagram](image)

Figure 1: Spread spectrum in the uplink for UMTS W-CDMA.

Looking at Figure 1, it can be seen that DPDCH and DPCCH are two independent sequences spread with global orthogonal spreading codes
\[ c_D = c_D(c_t + j c_Q) \quad c_C = c_C(j c_I + c_Q) \]
respectively. One of the possibilities for service multiplexing in one connection is depicted in Figure 2. When parallel services are multiplexed like this, each channel in a MS can be thought of as a different user with its own channel encoder and interleaver.

![Diagram](image)

Figure 2: Multiplexing of parallel services.

We will assume that we have \(K\) code divided independent transmissions in the uplink (summing up the amount of channels in each MS connection), and that the Base Station (BS) will manage the received data irrespectively of its origin. The medium access is asynchronous and, despite the channels within one connection are synchronous, these transmissions will be naturally matched in the proposed asynchronous detection algorithm.

There will be thus \(K\) transmitters consisting in a convolutional encoder, a block interleaver (a block being a 10 ms frame), a spreader, and a Root-Raised Cosine (RRC) filter \([7]\). We assume a non-dispersive Additive White Gaussian Noise (AWGN) channel. We assume as well perfect delay and phase estimation and perfect power control at the receiver, so that the only effects to compensate are the CDMA interference and the Gaussian noise. The output from the receiver RRC filter,

\[ s(t) = \sum_{j=0}^{K} b_j(t - t_j) c_j(t - t_j) + n(t) \quad (1) \]

feeds a bank of \(K\) matched filters whose task is to correlate \(s(t)\) with the corresponding delayed version of the spreading sequence. In Equation 1, \(b_j(t)\) is the sequence of coded bits from user \(j\), with period \(T_j\); \(c_j(t)\) is the spreading sequence (including the filtering), with period 256 chips (see \([7]\)), and \(t_j\) is the random delay of transmission \(j\), distributed uniformly over \((0, T_F)\). \(T_F = 10\) ms is the frame period. \(n(t)\) is white noise with power spectral density \(N_0/2\). The outputs from the matched filters (represented by the vector \(y\), see Figure 3) make up a set of sufficient statistics to perform MUD, as will be shown in the subsequent section.

3. MUD ITERATIVE ALGORITHM.

To perform turbodecoding \([4]\), it is necessary to have more than one encoder and a method to get the A Posteriori Probabilities (APP) for each transmitted bit \(b_{jt}(t)\) as a function of the A Priori Probabilities (APrP), the outputs from the channel and the information conveyed by the encoding process. This APP value will be employed as the APrP value in the next decoding step, whose APP outputs feed in turn the APrP inputs of the previous one, thus closing the iterative loop in a two encoder system. In addition to this, the information drawn from each decoder should be uncorrelated, otherwise no important improvement arises.

The classical turbodecoding scheme does not apply to our system, as we have only a convolutional encoder for each user, and, besides, we are interested on taking advantage of the convolutional encoding to help in
the MUD step. Fortunately, a method exists to get the APP values making use of the knowledge the receiver has about the spreading sequences. In [2] a Maximum Likelihood Sequence Estimation (MLSE) development to decode CDMA is introduced. It is shown there that the log-likelihood of the complete encoded sequence \( b = [b_1[1], \ldots, b_K[1], b_1[2], \ldots]^T = [b_1 \cdots b_M]^T \) is proportional to the quantity,

\[
\ln p(y | b) \propto \sum_i \lambda_i(s_i, s_{i+1}, b_i, y_i)
\]

where \( \lambda_i \) is a metric whose expression can be found in [2] and [6], \( s_i \) and \( s_{i+1} \) are a pair of fictitious states depending on the interfering bits, \( b_i \) is the bit connecting both states and \( y_i \), the matched filter output. Maximizing the likelihood means maximizing the sum of \( \lambda_i \) over all possible transmitted sets \( b \), for \( \lambda_i \) is proportional to the distribution

\[
\lambda_i \propto \ln p(y_i | s_i, s_{i+1})
\]

When the encoding can be seen as a Markov process, it is well known in turbodecoding [9] that, to get the APP value

\[
APP_i = P(b_i = \pm 1 | y) = \sum_{s_i, s_{i+1}, b_i = \pm 1} P(s_i, s_{i+1})
\]

we factorize

\[
P(s_i, s_{i+1}) = p(s_i, [y_i \cdots y_{i-1}]^T).
\]

\[
p(s_{i+1}, y_i | s_i) = p([y_{i+1} \cdots y_M]^T | s_{i+1})
\]

The first and the last distributions can be obtained with a recursive forward and backward algorithm (the BCJR algorithm, see [9]) which only needs to know \( p(s_{k+1}, y_k | s_k) \) for all \( k \). Moreover,

\[
p(s_{i+1}, y_i | s_i) = P(s_{i+1} | s_i)p(y_i | s_i, s_{i+1})
\]

The first term is the APP of the bit \( b_i \) (to be updated with the APP values from other decoders) and the logarithm of the second one is precisely proportional to \( \lambda_i \). So the framework introduced in [2] provides us with an artifact to compute all the quantities needed for turbodecoding. Besides, the independence between the spreading sequences, the data and the channel encoding, joined to the presence of the interleavers, guarantees the necessary uncorrelation between the decoding steps. The algorithm based on this principles is proposed and simulated in [6], where all the details about it can be found (they are not reproduced here for brevity's sake).

Therefore the reception technique is as depicted in Figure 3. The set of \( K \) matched filter outputs and the crosscorrelations of the spreading sequences provide, for the \( q \)-th iteration, the soft output \( \Psi^{(q)} \). This Log Likelihood Ratio (LLR) is obtained from the metrics defined in [2] and the application of the BCJR algorithm or any one of its suboptimum simplifications [10]. This information is fed to the corresponding deinterleavers and channel decoders (each working at its own data rate). After a similar soft-input soft-output processing (also based on the BCJR algorithm, now for convolutional decoders), we obtain the LLR of the channel encoder output \( \Lambda^{(q)} \), which, after interleaving, update the Multiuser Decoder metrics as new a priori information. \( \Theta^{(q)} \) is the LLR of the channel encoder input, which will be employed in the decision stage after some iterations.

Nevertheless, the application of this algorithm, as described in [6], to our system is not straightforward, since,

- the algorithm were only intended for users with same channel rates
- the delays considered were shorter than the bit period

The first drawback is overcome as follows: each channel in UMTS W-CDMA has to comply with rates multiple of the basic channel rate, 16 Kbps (spreading factor 256 [7]), so that we can assume that all users have an equivalent period of symbol

\[
T_{eq} = \min_{j \in \{1, \ldots, K\}} \{ T_j \}
\]

where \( T_j \) is the period of symbol of user \( j \). Each symbol transmitted by user \( j \) is thus theoretically split in \( T_j / T_{eq} \) equivalent symbols, each one spread by a fragment of the spreading sequence. Now we obtain \( T_j / T_{eq} \) different values of \( \Psi^{(q)}_j \) for each bit in the first decoder. This values are proportional to the matched filter outputs [6] (no longer working at the rate \( 1/T_j \), but at \( 1/T_{eq} \)) plus a correction term taking into account the CDMA interference. Accordingly, the value sent to the deinterleaver and to the channel decoder is the accumulation of these \( T_j / T_{eq} \) values (thus rebuilding the appropriate matched filter output plus the correction terms). In the feedback step, the LLR's from the channel decoders \( \Lambda^{(q)}_j \) are assigned to the corresponding equivalent bits weighted by \( T_{eq}/T_j \), because symmety between the decoding steps is to be maintained (\( T_j / T_{eq} \) LLR values were added before channel decoding, now they are split in \( T_j / T_{eq} \) equal values so that, when added again, the original LLR results).

On the other hand, the presence of delays higher than the equivalent bit period leads to the following
problem (see Figure 4): the first decoding step (box labelled Multiuser Detector in Figure 3) is to be performed over a sliding window containing at least one entire frame from each user, otherwise no deinterleaving and channel decoding could be performed for all of them. But even with a window of appropriate length (i.e., $V_1, V_2,...$), we can see that, after channel decoding in the $q$-th iteration, only the data LLR’s of the bits contained in the $T_F = 10$ ms frame are updated for each user and employed in the $q + 1$-th iteration. In this way, the MUD decoding step cannot exploit the information conveyed by the encoded data at the head of the subsequent frames (not entirely contained in the window), and the interference suffered by the bits in the tails of the actually decoded frames is not properly corrected.

To avoid this, the first decoding step is performed over the $V_1$ window, the resulting LLR’s are grouped as stated above, depending on the user’s rate, then they are channel decoded and the new LLR’s are fed back. Now the MUD decoding step is performed over the window labelled $V_2$. The LLR’s involved are fed back and now it is again $V_1$’s turn. This makes up a complete iteration. After some iterations, $V_2$ and $V_3$ take the place of $V_1$ and $V_2$ respectively, and the algorithm is repeated as explained. When the last iteration has been reached, so that no LLR contained in $V_1$ (i.e. the heads of the frames in $V_2$) is going to be updated any longer, a final decoding step is performed over $V_1$ and a decision is made with the LLR’s $\Omega^{(N)}$. The processing windows are shifted again a length $T_F$ and this procedure is repeated with the next pair of decoding windows. In the following section, the parameters employed in our simulations are described and the results depicted.

4. EXPERIMENTS.

We have simulated as stated a UMTS W-CDMA mobile environment with four MS’s, each one transmitting over one DPDCH and one DPCCH. This means $K = 8$. All DPCCH’s employ a channel rate of $R_j = 32$ Kbps, with a spreading factor $N_j = 128$. Two MS’s transmit at $R_j = 64$ Kbps over the DPDCH, and the other two at $R_j = 16$ Kbps (spreading factors $N_j = 64$ and 256 respectively). Each MS accesses the medium with a randomly generated delay between $(0, T_F)$. The spreading sequences were also randomly assigned. The channel encoder is the rate $R_c = 1/2$ convolutional encoder described in the standard. Trellis termination is made and no padding bits are added. The rate 1/3 convolutional encoder for the DPCCH’s [7] is not employed in order to make comparisons easier between the different channel rates.

The RRC filters and the AWGN channel guarantee that no intersymbol interference is present, so that the signals can be sampled at chip rate. The LLR’s in the two decoding steps are obtained just as in [6], where the log-MAP algorithm [10] is employed. The results are represented in terms of $BER = f(E_b/N_0)$. The signal to noise ratio for user $j$ as a function of the chip energy $E_c$ is

$$
\frac{E_{bl}(dB)}{N_0} = \frac{E_c}{N_0} (dB) + 10 \log N_j - 10 \log R_c
$$

Figure 5: BER as a function of $E_b/N_0$ (dB) for the user at 64 Kbps. $\bigcirc$: single user bound. $\boxplus$: matched filter + FEC. $\triangle$: Multiuser Turbo decoding; in order of descendig Pe: 0 iterations (no forward-backward windowing), 1 iteration, 2 iterations.
Simulations were made for $E_b/N_0 = -21.1, -20.1, -19.1$ and $-18.1$ dB. The worst case user is the channel with $R_f = 64$ Kbps (minimum $N_f$). Users with $R_f = 32$ and 16 Kbps have $E_b/N_0$ 3 and 6 dB over the 64 Kbps ratio respectively. For this reason, no results are given for 16 or 32 Kbps, as $BER < 10^{-5}$ leads to a prohibitive simulation time. Nevertheless, the BER's shown in Figure 5 are meaningful enough. It can be verified that turbodecoding clearly outperforms the conventional matched filter + FEC approach, even when no forward and backward windowing is applied (0 iterations). With one and two iterations further improvements can be achieved (though the differences are only important with signal to noise ratios over 1 dB) and the single user bound is approached.

5. CONCLUSIONS.

We have introduced here some major modifications to an already described turbodecoding method for CDMA, and have proposed solutions to the problems related with them, namely the different data rates and the random delays in asynchronous transmission. Although particularized to the UMTS W-CDMA standard, this application has provided a generalization of the turbodecoding algorithm which has proved to yield decoding gain. The only problem is the complexity involved, as the resources needed grow exponentially with the number of CDMA users. Nonetheless, turbodecoding can be employed to perform MUD over small groups of users, specially those ones with worse spreading sequence properties, or high channel rates. As a future work, it would be desirable to test this method with more realistic channels (i.e. dispersive channels) in order to verify it is an interesting alternative in Multiuser Detection.

6. ACKNOWLEDGEMENTS.

We are highly indebted to Luis Mendo Tomás, whose valuable comments on CDMA transmissions have played a key role in this article. We are also grateful to Carmen Peláez, who kindly helped us to correct the article.

7. REFERENCES


