Near Single User Performance using Iterative Multi-User Detection for CDMA with Turbo-Code Decoders

Mark C. Reed, Paul D. Alexander, John A. Asenstorfer
Mobile Communications Research Centre
University of South Australia
SPRI Building, Warrendi Road
The Levels SA 5095, Australia
FAX: +018 8302 3873

Christian B. Schlegel
Department of Electrical Engineering
3262 Merrill Engineering Building
University of Utah
Salt Lake City UT 84112, U.S.A
FAX: +1 801 581 5281

ABSTRACT

This paper discusses a code-division multiple access (CDMA) iterative multi-user receiver with forward error control (FEC) decoding. The maximum a-posteriori probability (MAP) criteria is used to derive the receiver. The decoding is done using a turbo-code decoder with modifications which are discussed. Iterations of the system are used to attain large performance improvements over conventional systems.

1. INTRODUCTION

In mobile communications, direct-sequence code-division multiple-access (DS-CDMA) systems, are now becoming available on the market. Unless orthogonal spreading codes are used the performance of these systems has been limited by their inherent multiple access interference. In [1] we demonstrated with random spreading codes, multiuser performance close to single user performance. This motivates us to use more powerful codes to see if we still get close to single user performance. In this paper we investigate an iterative DS/CDMA multi-user receiver, primarily intended for the uplink that uses turbo codes for FEC.

Recently a new coding method, named Turbo Coding, was introduced [2]. This technique achieves results close to the Shannon-limit. The technique combines the concepts of iterative decoding, soft-in/soft-out decoding, and non-uniform random interleaving. In [1] some of the ideas of turbo codes were adapted to the multi-user environment. Namely, the concepts of iterative decoding and soft input/soft output detection were used. In [1] we used a likelihood estimator based on the Reduced Complexity Recursive Detector [3]. The estimator outputs likelihoods which are then used by a metric generator to provide soft inputs for the single user decoders.

In this paper we extend the results of [1]. We adopt a turbo-code encoder/decoder set instead of convolutional codes as the turbo-code decoders are known to give better performance results. We discuss the implementation of this decoder and differences from the conventional turbo-code decoding implementation. We also perform a complexity analysis and show methods of reducing the complexity to \( O(K) \) per bit. We finally conclude with simulations of the full complexity turbo coded system before showing results for a reduced complexity system. Both results outperform previously published results for turbo coded DS/CDMA.

Throughout this paper scalars are lower case, vectors are underlined lower case, and matrices are underlined upper case. The symbols \( (\cdot)^t \) and \( |\cdot| \) are the transposition and determinant operators respectively. Variables have subscripts that refer to the time increment and superscripts that refer to the user.

2. SYSTEM DESCRIPTION

In this paper we model the uplink of a DS/CDMA communication system, which is based on a coded, discrete-time system. The channel adds white Gaussian noise with zero mean and variance \( \sigma^2 = N_0/2 \), where \( N_0 \) is the single sided noise power spectral density. The channel model is chip and symbol synchronous. There are \( K \) users each transmitting \( L \) coded symbols \( d_k^t \) where \( k \in \{1, \ldots, K\} \) and \( t \in \{0, \ldots, L-1\} \) identifies the symbol interval. The spreading code employed by user \( k \) at symbol interval \( t \) consists of \( N \) chips and is denoted \( s_k^t \in \{ \frac{1}{\sqrt{N}}, \frac{-1}{\sqrt{N}} \}^N \). The matched filter output \( y_k^t \) at time \( t \) can therefore be expressed as

\[
y_k^t = H_k \hat{d}_k^t + \tilde{n}_k^t,
\]

where \( \hat{d}_k^t = (d_k^{t_1}, \ldots, d_k^{t_L})^t \) is the coded data vector, \( H_k \) is the \( K \times K \) cross correlation matrix of the spreading sequences, and \( \tilde{n}_k^t \) is the noise vector. For a synchronous system the total received sequence is \( \hat{y} = (\hat{y}_1^0, \ldots, \hat{y}_L^{L-1}) \) and \( H_k = A_k^T A_k \) where \( A_k = (s_k^1, \ldots, s_k^K) \) is the bank of sequence matched filters.

The noise samples \( \tilde{n}_k^t \) have the correlation \( \text{E}[\tilde{n}_k^t \tilde{n}_k^t^*] = H_k \sigma^2 \). The coding method we use is limited to trellis codes and FEC is provided by turbo codes.

3. ITERATIVE DETECTOR WITH SINGLE USER FEC BLOCK

3.1 Uncoded MAP Criteria

To reduce complexity the multiuser receiver takes the MAP criteria across all users but assumes there is no coding. This reduces the number of states (or memory) in the received sequence dramatically. The multi-user
MAP criteria is now
\[
\hat{d}_t = \arg\max_{\hat{d}} \Pr\{\hat{d}|y_t\} = \arg\max_{\hat{d}} \frac{p(\hat{d}, y_t)}{p(y_t)} = \arg\max_{\hat{d}} p(\hat{d}|y_t)
\]

3.2 Implementation with a Single User FEC Block

In this section we will summarise the result from [1] before discussing the extensions to this work. Figure 1 shows a block diagram of the proposed iterative multiuser receiver with single user FEC blocks.

From the channel model (1), we know that the conditional probability of \(y_t\) is the multivariate Gaussian distribution \(p(y_t|d_t)\) [4]. We call this the likelihood calculation, as shown in Figure 1. Since we require single user information at each of the decoders we must convert \(p(y_t|d_t)\) into an appropriate form.

The single user FEC decoders for a turbo code decoder normally take as input \(p(y_t|d_t)\). We instead specify that the decoder input metric be equal to \(p(y_t|d_t)\), this is justified as the vector \(y_t\) contributes signal plus multiple access interference (MAI) to the bit of interest \(d_t\).

The metric for the single user FEC decoders is generated by the metric generator, as shown in Figure 1, which manipulates the conditional probability to get the joint probability
\[
p(y_t, d_t^{(k)} = d) = \sum_{d_t^{(k)} = d} p(y_t|d_t^{(k)}) \Pr\{d_t^{(k)}\}
\]
The specified conditional probability for SU decoding is \(p(y_t|d_t^{(k)})\). The metric generator then manipulates (2) into this form by
\[
p(y_t|d_t^{(k)} = d) = \frac{p(d_t^{(k)} = d, y_t)}{\Pr\{d_t^{(k)} = d\}} = \sum_{d_t^{(i)} = d} p(y_t|d_t^{(i)}) \prod_{i \neq k} \Pr\{d_t^{(i)}\},
\]
with the assumption that the coded symbols \(d_t^{(i)}\) are independent among the different users, \(d \in \{-1, +1\} \).

The FEC block can generate \(\Pr\{b_t^{(k)}|y\}\) (6) and \(\Pr\{d_t^{(k)}|y\}\) (3) as discussed in Section 5. Note the subscripts \(r\) and \(t\) are for the uncoded data and coded data time indices respectively.

4. ITERATING THE MULTIUSER RECEIVER

Multiuser systems describe sharing information between users. If this is done correctly a joint detection process results which attains improved performance. The metric generator (3) is described in a form that takes a-priori information (\(\Pr\{d_t^{(i)}\}\)), we now investigate the operation of the proposed system with iterations, and making a final decision.

In Turbo Code decoding [2] the output probability of the first MAP decoder \(\Pr\{b_t = b[y]\}\) used as a-priori information for the second MAP decoder. In a similar fashion we assign the turbo code decoder output probabilities from iteration \(i\) to the a-priori input probabilities for iteration \(i + 1\) in (3) where
\[
\Pr\{d_t^{(i)} = d\} = \Pr\{d_t^{(i)} = d|y\}
\]
On the first iteration \(\Pr\{d_t^{(k)} = d\} = 1/2\), i.e., all symbol sequences are equi-probable. This allows now to iterate our system a number of times in a similar fashion to that of a turbo code decoder.

After the number of iterations required has been completed a final decision is made at the output of the turbo code decoders to determine \(d_t^{(k)}\). This is done by making a hard decision on the most probable outputs according to the rule
\[
\hat{b}_t^{(k)} = \text{sgn}\left(\frac{\Pr\{b_t^{(k)} = 1|y\}}{\Pr\{b_t^{(k)} = 0|y\}} - 1\right) \forall k
\]

5. MAP DECODER

As the MAP decoder forms a central component in this paper we will briefly discuss it’s operation as documented by [5, 6].

The MAP decoder’s purpose is to estimate the a-posteriori channel symbol \(\Pr\{d_t|y\}\) or the a-posteriori information bit probability, \(\Pr\{b_t|y\}\), for a given received sequence \(y\). The MAP decoder for convolutional codes achieves this by traversing the decoding trellis in a backward and forward direction; if \(S_{t-1}\) and \(S_t\) are two states of the code trellis at time \(r = t\) and \(r\) the MAP decoder calculates
\[
\Pr\{S_t = m'; S_t = m|y\} = \Pr\{S_{t-1} = m'; d_t = d|y\}
\]
or \(\Pr\{S_{t-1} = m'; b_t = b[y]\}\) or \(\Pr\{S_{t-1} = m'; d_t = d|y\}\) which can be used to determine
\[
\Pr\{d_t|y\} = \sum_{S_{t-1}} \Pr\{S_{t-1} = m'; d_t = d|y\}
\]
and
\[
\Pr\{b_t|y\} = \sum_{S_{t-1}} \Pr\{S_{t-1} = m'; b_t = b[y]\}
\]
The MAP decoder calculates (4) by evaluating the joint probabilities [5, 6] (applying Bayes’ theorem). The joint probability, can be shown [5, 6] to be equal to
\[
\Pr\{S_{t-1} = m', S_t = m, y\} = \beta_r(m) \gamma_r(m') \alpha_{r-1}(m')
\]
where \(\beta_r(m) = \Pr\{y_{t-1}'|S_{t-1} = m\}\) and \(\gamma_r(m') = \Pr\{S_r = m' | y_{t-1}'\}\). The transition metric, \(\gamma_r(m', m)\), is equal to [5, 6]
\[
\gamma_r(m', m) = \sum_{t=b_r} \Pr\{S_t = m, y_t|S_{t-1} = m'\}
\]
where
\[
\Pr\{S_{t-1} = m'; S_t = m; y_t|S_{t-1} = m'\} = \sum_{b=b_r} p(y_t|b) \Pr\{b|m, m'\} \Pr\{S_t = m|S_{t-1} = m'\}
\]
where \( \Pr\{S_r = m|S_{r-1} = m'\} \) is the state transition probability. The distribution \( p(y_t|b) \) is the memoryless channel symbol distribution given the transmitted information bit. When the transition from state \( m \) to \( m' \) exists for a known value of \( b \), and there is only one transition then \( \Pr\{b|m, m'\} = 1 \), and (7) can be simplified to
\[
\gamma_r(m', m) = p(y_t|b) \Pr\{S_r = m|S_{r-1} = m'\} \tag{8}
\]

6. NEW FEC DECODER

In previous work [1] a MAP decoder was used as the FEC block. In this paper we introduce a new technique using a turbo-code decoder [2]. The use of turbo codes for error correction gives us a very powerful coding stage. As with turbo codes we expect and get higher performance than using a single MAP decoder. The turbo code decoders need to be modified slightly to produce suitable output probabilities for the a-priori input to the multiuser metric generator. The single user turbo code encoders were implemented in the same way as those used in [2], except that there is \( K \) of them, one for each user.

For each user there is a turbo code decoder which consists of two MAP decoders, the first MAP decoder receives a block of likelihood values \( p(y_t|d^{(k)}_t) \). This data is used directly in the branch metric calculation of the first MAP decoder so (8) is modified to
\[
\gamma_r(m', m) = \Pr\{S_r = m|S_{r-1} = m'\} \prod_{d^{(k)}_t} p(y_t|d^{(k)}_t)
\]

where the product is over all the \( d^{(k)}_t \) values that produce the transition of the MAP decoder from state \( m \) to state \( m' \). Like the serial concatenated convolutional code (SCCC) solution [7] the turbo code decoder takes no a-priori information \( \Pr\{S_r = m|S_{r-1} = m'\} = 1/2 \), although the MAP decoders within the turbo decoder pass a-priori information between each other.

Iterative decoding is performed slightly differently to that of [2]. That is, the factorisation of the log likelihood output of decoder 1 needs to be modified as we are provided with a likelihood measure of \( p(y_t|d^{(k)}_t) \) not a bit \( b \in \{+1, -1\} \) with additive white Gaussian noise (AWGN). In a way similar to the factorisation of the output of the MAP decoder shown in [6] we express the log likelihood ratio as
\[
\Lambda^{(k)}_r(b_r) = L_{a2,r}^{(k)} + L_{b1,r}^{(k)} + L_{e1,r}^{(k)} \tag{9}
\]

where
\[
L_{b1,r}^{(k)}(b_r) = \ln \left\{ \frac{\Pr\{b_r = 1|y_r^{(k)}\}}{\Pr\{b_r = 0|y_r^{(k)}\}} \right\}
\]
is the log likelihood ratio output of MAP decoder 1 at time \( r \). \( L_{a2,r} \) is the a-priori information from MAP decoder 2.

\[
L_{e1,r}^{(k)} = \ln \left\{ \frac{\Pr\{b_r^{(k)}|b = 1\}}{\Pr\{b_r^{(k)}|b = 0\}} \right\}
\]
is the log likelihood ratio of the systematic (information) bits and \( L_{e1}^{(k)} \) is called the extrinsic information, which after being computed from (9) and interleaved is the a-priori information for the second MAP decoder for user \( k \).

In [2] after the desired number of iterations a hard decision on the information bits is performed to generate the final result. We however want to produce a soft a-posterior probability output which can be used as a-priori information for the metric generator. The turbo code decoder is modified to produce uncoded and coded bit probabilities as a-priori information to the metric calculation (\( \Pr\{d^{(k)}_t|y_r^{(k)}\} \)). The output probabilities are calculated using (5). The coded bit probabilities are combined from the output of the first and second MAP decoders because the two codes are punctured and because each MAP decoder uses a different coded sequence (due to interleaving). The uncoded bit probabilities are taken from the output of the second MAP decoder for user \( k \). The turbo code decoder outputs a block containing \( B = L/R \) values \( \Pr\{d^{(k)}_t|y_r^{(k)}\} \) in the same sequence as

\[ b_r^{(K)} \]
7. COMPLEXITY ANALYSIS

In this section we study the complexity of the system described. We determine the number of floating point operations required per information symbol transmitted for the likelihood calculation, metric generator and turbo decoder. We assign the variable $S$ to be the number of states in the decoder, the number of users to equal $K$, the number of paths out of each state to equal $P$, and the rate of the FEC code to equal $R$.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Complexity (FLOP per Information Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likelihood</td>
<td>$(2^K((K^3/6 + 4K^2 + 6K + 4))/RK)$</td>
</tr>
<tr>
<td>Metric (1 It.)</td>
<td>$(2^K(K - 1))/RK$</td>
</tr>
<tr>
<td>Decode (1 It.)</td>
<td>$2(PS^{-1}/2 + 10) - 2S + 16$</td>
</tr>
</tbody>
</table>

Table 1: Complexity of Iterative Multiuser Receiver Components

Table 1 shows the complexity in terms of these variables. Fig. 2 shows the complexity of the three components of the system for a varying number of users. For more than six users (the case of interest for practical implementation) the $2^K$ term in the likelihood generation and metric generation dominates complexity. Figure 2 shows the complexity of the likelihood calculation as “like”, the complexity of the metric generation (3) as “metric”, and the complexity of the turbo code decoder as “tc”, against users. The curve “total” is the sum of these three complexities. The number of turbo code iterations was set to four and the number of multiuser iterations was set to three, the same as used in the final performance tests.

![Complexity in FLOP per symbol against number of users](image)

Figure 2: Complexity in FLOP per symbol against number of users.

7.1 Complexity Reduction Techniques

From the complexity analysis it is clear that our solution is exponentially complex with the number of users, like the optimal decoder [8]. In this section we propose techniques to reduce this complexity. The likelihood calculation of the multi-variate Gaussian distribution contains a number of complex linear algebraic tasks. Due to independence from the hypothesised vector $\hat{d}$, we can simplify this to

$$p(y_i|\hat{d}) = C_h \exp \left\{ -\frac{1}{2\sigma^2} (2y_i^T d + d^T H_i d_i) \right\} \quad (10)$$

where $C_h$ is a constant and does not have to be computed. This is a significant reduction as we have removed the need to compute the inverse of the cross correlation matrix $(H_i)$. Even with this scheme the dominant $2^K$ term still exists in the likelihood calculation and the metric generation.

Another technique, suggested independently by Hoeher et al. [9] and Nasiri-Kenari et al. [3], is to calculate likelihoods $p(y_i|\hat{d})$ based on a one bit difference from the previous likelihood. This means that the likelihood calculation only has to be done once. From there a step-wise difference calculation is required to determine all the possible likelihood values. The technique of [3] reduces the complexity but still requires the computation of $2^K$ likelihoods, this method can however be used with a $M$ algorithm requiring only order $K$ complexity. Using the two reduction techniques the complexity of the likelihood calculator and the metric generator are now

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Complexity (FLOP per Information Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likelihood Calc.(1 It.)</td>
<td>$(2^K + 3K + 5)/R$</td>
</tr>
<tr>
<td>Metric Gen.(1 It.)</td>
<td>$(K^2 - K)/(RK)$</td>
</tr>
</tbody>
</table>

Table 2: Reduced Complexity of Iterative Multiuser Receiver Components

The complexity of the likelihood calculation is shown in Fig. 2 as “malg-like” and the complexity of the metric calculation is “malg_metric”, when $M = K$. The total sum complexity is now “malg_total”, which is order $K$, meeting the reduced complexity requirement.

If the likelihood difference calculation is used to calculate only a list of $M$ outputs we no longer have all the possible likelihoods available. To try to maximise performance under these conditions the likelihood calculation is run with a different starting vector on the second iteration generating a second likelihood list. This starting vector is selected based on making a hard decision on the likelihood output from the single user decoders. The likelihood lists from previous iterations are combined to increase the likelihood data available for metric generation.

8. SIMULATION RESULTS

The simulation result in Fig. 3 shows the average
performance over users of the system for 1, 2, and 3 system iterations, each with four turbo decoder iterations. "tc_single_user" is single user performance using turbo codes for FEC, with four iterations. The simulation was for a synchronous chip and symbol channel with random spreading codes and values \( N = 7 \), \( K = 5 \). All of the possible \( 2^K \) likelihood values \( P(y|\theta) \) were calculated. A turbo-code encoder of rate \( R = 1/2 \), consisting of two parallel recursive systematic encoders, \( G(37,21) \), separated by random interleaving was used as the error control code. The block size of the interleave was set to \( B = L/R = 200 \) uncoded bits.

We see that most of the improvement occurs in the second iteration. It is also apparent that after 2 or 3 iterations in this highly loaded random code case near single user performance is obtained.

9. CONCLUSIONS

In this paper we have discussed an iterative multiuser receiver with single user turbo code decoders. This receiver is derived from the MAP criteria and takes a-priori information from the FEC decoding block to improve its probabilistic estimates. For the first time we incorporate a turbo-code decoder as the FEC block and discuss necessary modifications required to achieve the reported results. The complexity of this system is large so we discuss methods of reducing this with order \( K \) techniques finally used and simulated.

Simulation performance results show that the iterative multi-user receiver design combined with turbo code decoding approaches turbo code single user performance, even with a large number of users relative to the spreading factor. With the complexity reduced to \( O(K) \) per bit results within 2 dB of the single user result are achieved.

REFERENCES


