

ITERATIVE MULTIUSER DETECTION AND DECODING

Suman Das Elza Erkip Joseph R. Cavallaro Behnaam Aazhang
suman@rice.edu elza@rice.edu cavallar@rice.edu aaz@rice.edu

Department of Electrical and Computer Engineering

Rice University, MS 366

6100 South Main Street

Houston, TX 77005, USA

Phone: (713)-527-4719 Fax: (713)-524-5237

Abstract— An iterative detection and decoding algorithm is explored for a convolutionally coded DS-CDMA system. The optimum decoding strategy is to consider the trellises of all the users simultaneously, but this has exponential complexity in the number of users. We use an iterative +multistage detection scheme combined with the MAP algorithm to reduce the complexity of the receiver without significant performance degradation. The simulation results show that our algorithm consistently outperforms other existing low complexity algorithms. We also provide a framework where the similarities and differences between various algorithms become transparent. Based on this framework, we make a quantitative comparison of the various schemes under investigation.

I. INTRODUCTION

Direct-sequence code-division multiple-access (DS-CDMA) has emerged as an attractive alternative for wireless communication networks. In this system, a multitude of users transmit their information bits at the same time using the same frequency band. However each user is assigned a different signature waveform to modulate the information bits. Due to the inherent asynchronous nature of the system, the signature waveforms cannot be designed to be mutually orthogonal for all possible delays and hence CDMA systems become inherently interference limited. Multiuser detection schemes provide one possible way to mitigate this problem. The optimal detector proposed by Verdú [1] has been shown to effectively combat the *near-far* problem, but the exponential complexity of this algorithm has led to the development of several suboptimum linear multiuser detectors [2], [3], [4].

Apart from the multiple access interference (MAI), the second source of error is the thermal noise. The raw information bits of the users are coded using various channel encoders to provide added protection against detection errors due to the noise, thereby improving the performance of the system. Convolution coding schemes in conjunction with the DS-CDMA system have been shown ([5], [6]) to provide sufficient protection against detection errors. The Viterbi

algorithm is the traditional method to decode information bits from convolutionally encoded data bits. For an asynchronous multiuser system the optimum decoding method is to combine the trellises of all the users. The complexity of this trellis is $O(2^{\kappa K})$ where K is the number of users and κ is the code constraint length, and this exponential complexity in the number of users limits the feasibility of the algorithm.

In this paper we discuss strategies to reduce the complexity of the joint decoder and detector. We also describe a general framework and compare other related schemes in this framework. The receiver we describe is essentially derived from the *maximum a-posteriori* (MAP) criteria for all the users. It incorporates the iterative parallel interference cancellation scheme and single user MAP decoders to reduce complexity and yet perform close to the optimal algorithm.

In the next section we present the mathematical model of our system. Section 3 contains most of our results. We briefly describe the multistage detector and MAP algorithm and show how to integrate these two ideas. We describe a few other related results. In section 4 we provide a framework to compare some of the existing schemes with our scheme. Finally we provide concluding remarks.

II. SYSTEM MODEL

We consider an asynchronous system with K users. Let the k^{th} user have a normalized signature waveform given by $s_k (< s_k, s_k > = 1)$ which extends over the symbol period $[0, T]$ and consists of N_c chips. The received power corresponding to the user k is given by A_k . Let the raw information bits of user k be given by the vector \mathbf{b}_k and the corresponding coded bits be given by \mathbf{d}_k . Thus we can write the cumulative received signal over the period of N coded bits as

$$r(t) = \sum_{k=1}^K \sum_{i=1}^N A_k d_k(i) s_k(t - iT - \tau_k) + z(t),$$

where $z(t)$ is the additive white Gaussian noise. The discretized chip matched filter output of this signal is an $N_c(N + 1) \times 1$ vector

$$\mathbf{r} = \mathbf{S}\mathbf{A}\mathbf{d} + \mathbf{z}, \quad (1)$$

Research supported by Nokia Inc., by the Texas Advanced Technology Program under grants 1995-#003604-049 and 1997-#003604-044, and by NSF under grants NCR-9506681 and CDA-9617383.

where \mathbf{S} is an $N_c(N+1) \times NK$ matrix of the signature waveforms, \mathbf{A} is the $NK \times NK$ diagonal matrix of user amplitudes and \mathbf{d} is the coded bits of all users over period $[0, N]$. Further details about this model of the uplink can be found in [5]. If the received signal is passed through a bank of code matched filters, then the output of the filter bank can be expressed as an $NK \times 1$ vector

$$\mathbf{y} = \mathbf{R}_N \mathbf{d} + \eta, \quad (2)$$

where $\mathbf{R}_N = \mathbf{S}'\mathbf{S}$ is the asynchronous code correlation matrix and η is the additive noise vector. The output of the bank of code-matched filter provides a sufficient statistic for the estimation of the vector \mathbf{d} .

III. JOINT DETECTION AND DECODING

The receiver has to retrieve the information bits of all users from the received signal sequence. We will use a joint detection and decoding strategy to achieve this goal.

A. MAP decoding

Since MAP forms the underlying decoding strategy for our algorithm, let us give a brief exposition of the MAP algorithm for both uncoded and coded data bits. The MAP decoding was first described by Bahl et al in [7] in the context of minimizing bit error probability for a given received signal sequence. However our MAP decoder is slightly different in spirit. Like the Viterbi algorithm, we will use the MAP rule to minimize the probability of error when we estimate the whole codeword rather than a single bit.

In MAP decoding, the receiver's estimate is the most probable information bit sequence, $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_K]'$, of all users given the received signal \mathbf{r} . In log probability form this can be expressed as,

$$\begin{aligned} \hat{\mathbf{b}} &= \arg \max_{\mathbf{b}} \log p(\mathbf{b}|\mathbf{r}) \\ &= \arg \max_{\mathbf{b}} [\log p(\mathbf{b}, \mathbf{r}) - \log p(\mathbf{r})] \\ &= \arg \max_{\mathbf{b}} \log p(\mathbf{b}, \mathbf{r}). \end{aligned}$$

The second equality above follows from the Bayes' rule and the third equality is true because the maximization process is not dependent on $p(\mathbf{r})$. This is an unconstrained maximization problem in the sense that all information bit sequences are feasible. However if we consider the estimation of the coded bit sequences then we can rewrite the same optimization problem in terms of coded bits as

$$\hat{\mathbf{d}} = \arg \max_{\mathbf{d} \in \mathcal{C}} \log p(\mathbf{d}|\mathbf{r}), \quad (3)$$

where \mathcal{C} consists of the codebook of all users. The above expression can be rewritten as

$$\hat{\mathbf{d}} = \arg \max_{\mathbf{d}} \log(p(\mathbf{d}|\mathbf{r})\mathcal{I}(\mathbf{d} \in \mathcal{C}))$$

$$= \arg \max_{\mathbf{d}} \log(p(\mathbf{d}, \mathbf{r})\mathcal{I}(\mathbf{d} \in \mathcal{C})),$$

where \mathcal{I} denotes the indicator function. Hence in this optimization we have to restrict our search space to valid codewords \mathbf{d} of length $NK \times 1$.

B. Multistage detection

A multiuser MAP decoding scheme requires exponential complexity with respect to the number of users [8], as we have to jointly search through the codebooks of all the users. One way to reduce the complexity of the multiuser MAP decoder is to use only the outputs of the i^{th} matched filter as a statistic to estimate the coded bits for the i^{th} user and perform single user decoding. However this statistic is corrupted by the interference from other users. The soft output of a multistage detector is a better statistic for MAP decoding as it mitigates the MAI.

For our algorithm we divide the users into two disjoint categories \mathcal{C} - the set of users of concern and \mathcal{I} - the set of interfering users. Note that $\mathcal{C} \cup \mathcal{I} = \{1, \dots, K\}$. The quantities \mathbf{d}_C and \mathbf{d}_I represent the coded bits of users in set \mathcal{C} and \mathcal{I} respectively. Similarly we separate the columns of \mathbf{S} into \mathbf{S}_C - the signature waveforms of the users of concern and \mathbf{S}_I - the signature waveforms of the interfering users. The dimension of \mathbf{S}_I is given by $N_c(N+1) \times NK_I$, where K_I is the number of interfering users. We can now write the received signal as the sum of the contribution from the users of concern and the interfering users

$$\mathbf{r} = \mathbf{S}_C \mathbf{A}_C \mathbf{d}_C + \mathbf{S}_I \mathbf{A}_I \mathbf{d}_I + z.$$

If we get an error free estimate of the bits transmitted by the interfering users, we can remove the contribution of the users in the set \mathcal{I} to get a interference free signal. In the multistage detector the statistic used to estimate \mathbf{d}_C is given by

$$\hat{\mathbf{y}}_C = \mathbf{S}'_C (\mathbf{r} - \mathbf{S}_I \mathbf{A}_I \hat{\mathbf{d}}_I) \quad (4)$$

where $\hat{\mathbf{d}}_I$ is the estimate of interfering users bits from the previous step of the multistage detector. We subtract the estimated contribution of the interfering users and project this signal into the signal space of the user of concern to get an improved statistic for detecting \mathbf{d}_C .

C. Iterative multiuser detection and decoding

As explained in the previous section we can use a parallel interference cancellation scheme to obtain an interference free statistics to estimate the transmitted bits of a particular user. The MAP criteria in (3) can be rewritten as

$$\hat{\mathbf{d}} = \arg \max_{\mathbf{d} \in \mathcal{C}} [\log p(\mathbf{r}|\mathbf{d}) + \log p(\mathbf{d})]$$

Ideally this optimization should be obtained over the entire joint code space of all users. However the complexity of calculating $p(\mathbf{r}|\mathbf{d})$ grows exponentially with the number of users. Let us now assume that \mathcal{C} , the set of users of concern consists of one user, c . Instead of using \mathbf{r} to jointly detect all the bits \mathbf{d} of all the users we can use the interference free statistic $\hat{\mathbf{y}}_c$ to decode the bits \mathbf{d}_c for user c . By doing this we approximate the ideal MAP decoder by K independent MAP decoders:

$$\begin{aligned}\hat{\mathbf{d}}_c &= \arg \max_{\mathbf{d}_c \in \mathcal{C}_c} \log p(\mathbf{d}_c|\hat{\mathbf{y}}_c) \quad (\forall c \in \{1, K\}) \\ &= \arg \max_{\mathbf{d}_c \in \mathcal{C}_c} [\log p(\hat{\mathbf{y}}_c|\mathbf{d}_c) + \log p(\mathbf{d}_c)]\end{aligned}\quad (5)$$

We will approximate $p(\hat{\mathbf{y}}_c|\mathbf{d}_c)$ with the conditional distribution which results when we cancel the interference from other users.

We start by finding the initial estimates for the coded bits of all users by considering the matched filter outputs in (2) and ignoring the interference from all the other users. Hence for user c , we use Viterbi algorithm on the corresponding matched filter output \mathbf{y}_c to get $\hat{\mathbf{d}}_c^1$, the initial estimate for \mathbf{d}_c . In the calculation of $p(\mathbf{d}_c|\hat{\mathbf{y}}_c)$ we assume that all the codewords are equally probable and the prior $p(\mathbf{d}_c)$ is uniformly distributed over the code space \mathcal{C}_c . In the next iteration step, we use these initial estimates $\hat{\mathbf{d}}_1^1, \dots, \hat{\mathbf{d}}_K^1$ to subtract off the interference and use $\hat{\mathbf{y}}_c$ to update our estimate of \mathbf{d}_c . In this iteration step, the prior $p(\mathbf{d}_c^2)$ is assumed to be equal to the posterior probability, $p(\mathbf{d}_c^1|\hat{\mathbf{y}}_c^1)$, calculated in the previous step. We repeat this procedure iteratively by replacing $\hat{\mathbf{d}}_1^{i-1}, \dots, \hat{\mathbf{d}}_K^{i-1}$ with $\hat{\mathbf{d}}_1^i, \dots, \hat{\mathbf{d}}_K^i$ at iteration step i while updating the prior probability $p(\mathbf{d}_c)$ with the posterior probability $p(\mathbf{d}_c|\hat{\mathbf{y}}_c)$ of the previous iteration step. As the number of iterations increase, we expect the estimates for the coded bits to become more reliable, which in turn makes $\hat{\mathbf{y}}_c$ a better statistic for the new estimate of \mathbf{d}_c . The iteration process can be terminated when there is no difference between the estimates of the bits in consecutive iteration steps. It should be noted that if we can successfully estimate all the users bits then the single user decoding scheme is identical to the joint multiuser decoding scheme.

C.1 Algorithm

Summarizing, the iterative detection and decoding algorithm is as follows:

1. Obtain an initial estimate of \mathbf{d}^0 , possibly from the output of matched filters.
2. In the i^{th} iteration,
 - For all $c \in \{1, K\}$ compute

$$\hat{\mathbf{y}}_c^i = \mathbf{S}'_c(\mathbf{r} - \mathbf{S}_I \mathbf{A}_I \hat{\mathbf{d}}_I^{i-1}) \quad I = \{\{1, K\} \setminus c\},$$

where the superscript i refers to the computation in the i^{th} step.

- From $\hat{\mathbf{y}}_c^i$ calculate $\hat{\mathbf{d}}_c^i$ and $p^i(\mathbf{d}_c|\hat{\mathbf{y}}_c^i)$ using equation (5). We use $p^{i-1}(\mathbf{d}_c|\hat{\mathbf{y}}_c^{i-1})$ as the prior $p^i(\mathbf{d}_c)$ in the calculation.

3. Stop when there is no further change in the estimate of \mathbf{d} between two successive iterations.

C.2 Strategy to save storage

The success of our iterative MAP algorithm hinges on how effectively the statistic $\hat{\mathbf{y}}_c$ captures the information about \mathbf{d}_c . This depends on how effective our multiuser detector is in mitigating the MAI. Equally important is how closely we can approximate the underlying distribution of \mathbf{d}_c for a given set of observations $\hat{\mathbf{y}}_c$. Our strategy is to iteratively update the prior to reach the underlying posterior distribution $p(\mathbf{d}_c|\mathbf{r})$. To achieve this we approximate the prior $p(\mathbf{d}_c)$ at the i^{th} step by the conditional distribution $p(\mathbf{d}_c|\hat{\mathbf{y}}_c)$ obtained at the $(i-1)^{th}$ step. However this necessitates that we store $p(\mathbf{d}_c|\hat{\mathbf{y}}_c)$ for all the possible sequences in the code space \mathcal{C}_c . This will require memory of size exponential in the length of the bit sequence N . To reduce the required storage we use a strategy similar to list decoding [9], and store the probabilities for the top M most likely codewords. Our assumption is $\sum_{\mathbf{d}_c \in \mathcal{M}} p(\mathbf{d}_c|\hat{\mathbf{y}}_c) \approx 1$, where \mathcal{M} denotes the set of M most likely codewords. Hence in the i^{th} iteration we use the probabilities for the codewords within this M -set as the prior in the next iteration and for all others we use a uniform probability smaller than the probability of the M^{th} most likely code.

Another standard approximation as explained in the M-Algorithm [10], is to consider a truncated trellis to avoid large decoding delay. We assume a path length of 5κ , where κ is the memory of the convolutional code. We use this to make a decision only on the first branch of the trellis and then we slide our trellis window to use the next level. The list decoding scheme does not guarantee that the top M paths stored upto level l will lead to the top M paths upto any subsequent stage. Thus it needs elaborate sorting at each stage. We will show in the following lemma, that for such a scheme, in order to evaluate the top M paths at any stage we need to store at most the top $M2^\kappa$ paths and we do not need an elaborate sorting scheme at each stage.

Lemma 1: *To evaluate the top M paths at any stage we need to store at most $M2^\kappa$ paths.*

Proof: In order to calculate the top M most probable paths, we argue that if we have M most probable paths to each state at a particular level, we will be able calculate the top M most probable paths in the next level. We will prove this lemma using induction on the level of the trellis. The lemma is true for level 2, since all the paths originate from the same node at level 1.

Let us assume the lemma is true for level $(l - 1)$. We have to prove that if we have the top M paths at each node in the $(l - 1)^{th}$ level we will be able to get the top M paths at the l^{th} level.

Let us assume the contrary. Suppose there exists a path P^l ending in node j_l at the l^{th} level which is one of the top M most probable paths but P^{l-1} does not correspond to one of the top M paths stored at each node in level $l - 1$. Here the superscript l denotes that we observe the path up to level l . Note that we are calculating the probabilities conditioned on the received signal \mathbf{y}^l . Now since P^l is a path from level 1 to level l it must pass through a node in level $l - 1$ of the trellis. Let the node at that level be j_{l-1} . That means there is a branch $P(l)$, from node j_{l-1} to node j_l . Since P^{l-1} is not one of the top M paths ending in node j_{l-1} , $p(P^{l-1}|\mathbf{y}^{l-1})$ is less than the corresponding probabilities for all the M paths ending at the same node. Using Bayes' rule we can write

$$\log p(P^l|\mathbf{y}^l) \propto \log p(\mathbf{y}^l|P^l) + \log(P^l)$$

where we have ignored $\log p(\mathbf{y}^l)$ because it contributes equally to all path and we will be comparing the probabilities of each path. Moreover

$$\log p(\mathbf{y}^l|P^l) = \log p(\mathbf{y}^{l-1}|P^{l-1}) + \log p(\mathbf{y}(l)|P(l))$$

as the channel is memoryless. Here $\mathbf{y}(l)$ denotes the output at stage l . Combining, we have

$$\begin{aligned} \log p(P^l|\mathbf{y}^l) &\propto \log p(\mathbf{y}^{l-1}|P^{l-1}) + \log p(\mathbf{y}(l)|P(l)) \\ &\quad + \log(P^l) \\ &\propto \log p(P^{l-1}|\mathbf{y}^{l-1}) + \log p(\mathbf{y}(l)|P(l)) \\ &\quad + \log P(l) \\ &< \log p(P_m^{l-1}|\mathbf{y}^{l-1}) + \log p(\mathbf{y}(l)|P(l)) \\ &\quad + \log P(l), \end{aligned}$$

for $m \in M$, where P_m^{l-1} denotes one of the M most probable paths ending at j_{l-1} at stage $l - 1$. Thus using one of the paths from the top M paths ending at node j_{l-1} and appending the branch $P(l)$, we get M paths which are more likely than P^l . This is a contradiction to our assumption and this proves our lemma. \square

This algorithm allows us to determine the M paths efficiently without needing to resort to elaborate sorting.

D. Simulation

Joint detection and decoding have been investigated by several researchers. Fawer et al [11] considered a scheme similar to the scheme we suggested. They use a multistage detector in conjunction with trellis coded modulation. After the removal of the interference from other users, they search for the most

likely coded bit sequence. However in their scheme they don't try to update their priors. Some recent work has been presented by Schlegel et al [12], where they try to minimize the symbol error using the iterative MAP algorithm. Since our aim is to reduce the sequence error we don't present the comparative study with this scheme.

For our simulations we have used gold code sequences of length 7. Our system has 4 users with the amplitude of all other users being twice that of user 1. We have used a convolutional code of rate 2/3 and depth 2. The delays of the users were assumed to be distributed uniformly over the bit period.

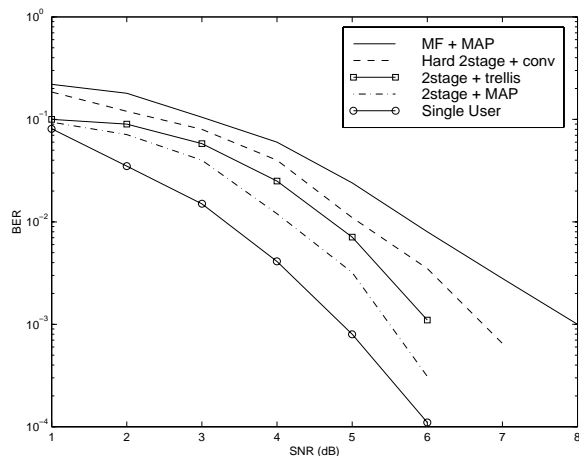


Fig. 1. Comparative study of various joint detection and decoding algorithms

The term “2stage + MAP” refers to our algorithm and “2stage + Trellis” refers to the algorithms proposed in [11]. In “MF + MAP” we use the output of the matched filter as the statistic for estimation of the coded bits of the corresponding users. We do not do any any form of interference cancellation. In “Hard 2stage + conv” we consider an isolated multistage detection scheme which outputs an unconstrained hard decision of coded bits. These hard decisions are fed into a Viterbi decoder to retrieve the information bits. Our simulation results (Figure 1) show that our algorithm consistently outperform all the other schemes. It provides almost 0.5dB gain over the best algorithm for a bit error rate of 10^{-3} . For comparison we have also included the performance of the single user system using the same convolutional code.

D.1 Storage and computational cost

One of the main drawbacks of the multiuser joint decoding is the computational complexity. However a single user decoding scheme as noted in “MF + MAP” suffers from performance degradation due to near far effect. The multistage detection combined with MAP decoding tries to achieve a balance between the two extremes. However with each iteration

while the performance of the system should improve, this would come at added computational cost. We performed simulation to study the sensitivity of the performance of our algorithm with the number of iterations.

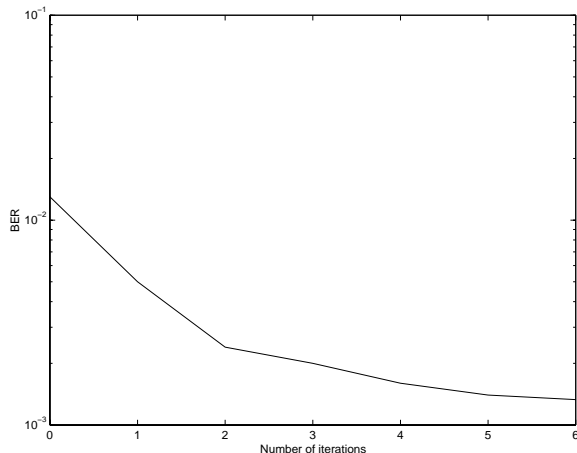


Fig. 2. Convergence study of the joint detection and decoding algorithm

We studied our simulation with the same system parameters as before. For our study we fixed our background SNR at 6dB. Figure 2 shows that our algorithm achieves its near optimal performance after only 2-3 iterations. Thus with a relatively low computational complexity we hope to achieve performance close to the multiuser joint decoding scheme.

We also studied the sensitivity of our algorithm with the number of paths being stored. This is directly related to the required storage of the system. Ideally we would require to store all possible paths

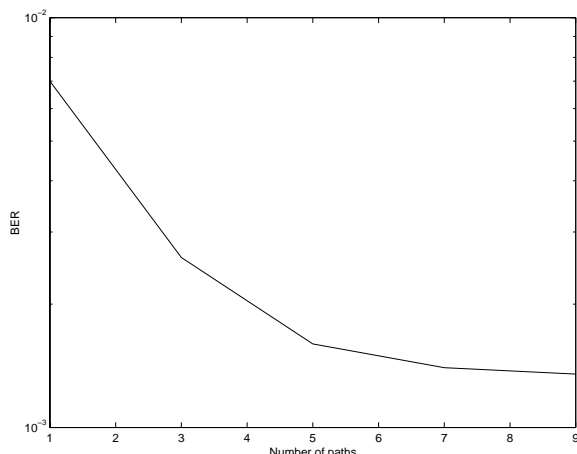


Fig. 3. Sensitivity study of the joint detection and decoding algorithm with required storage

that we enumerate in the Viterbi decoding scheme across iterations. However this would require a lot of storage. Our simulation results show that with

statistics for relatively few (5-6) number of paths being stored, we can achieve almost ideal performance. However it should be mentioned that both the above sensitivity results depend on other system parameters like the number of users in the system, the spreading gain N_c and the depth of the code κ .

IV. FRAMEWORK TO COMPARE VARIOUS SCHEMES

In this section, we provide a framework which helps us in identifying the difference between the proposed scheme and other existing schemes illustrated in figure 1. Recall that in our algorithm we first subtract off the interference from other users and then find the maximum a-priori codeword \mathbf{d}_c for user c . Let us consider iteration step i , and focus on user c , which is the user of interest. Let $q^i(\mathbf{d}_c)$ denote the conditional probability of \mathbf{d}_c after the interference cancellation in iteration i , where \mathbf{d}_c is restricted to the code space \mathcal{C} . We can write,

$$q^i(\mathbf{d}_c) = p^i(\mathbf{d}_c|\mathbf{r})\mathcal{I}(\mathbf{d}_c \in \mathcal{C}_c)\mathcal{I}(\mathbf{d}_I = \hat{\mathbf{d}}_I^{i-1}), \quad (6)$$

where $p^i(\mathbf{d}_c|\mathbf{r})$ is the updated unconstrained probability estimator for \mathbf{d}_c , and $\hat{\mathbf{d}}_I^{i-1}$ are the estimates of the coded bits of the interfering users in iteration step $i - 1$. The distribution $q^i(\mathbf{d}_c)$ is the updated distribution on the coded bits \mathbf{d}_c at iteration step i . The term $\mathcal{I}(\mathbf{d}_c \in \mathcal{C}_c)$ ensures that we restrict our attention to the codewords of \mathcal{C}_c and $\mathcal{I}(\mathbf{d}_I = \hat{\mathbf{d}}_I^{i-1})$ enables us to use $\hat{\mathbf{y}}_c^i$ as in equation (4) instead of \mathbf{r} in our calculations. Note that \mathbf{d}_c and \mathbf{d}_I are the coded bits corresponding to different users and thus they are independent. Hence equation (6) is equivalent to

$$q^i(\mathbf{d}_c) = p^i(\mathbf{d}_c|\hat{\mathbf{y}}_c^i)\mathcal{I}(\mathbf{d}_c \in \mathcal{C}_c).$$

Note that,

$$\begin{aligned} p^i(\mathbf{d}_c|\hat{\mathbf{y}}_c^i) &= \frac{p^i(\hat{\mathbf{y}}_c^i|\mathbf{d}_c)p^i(\mathbf{d}_c)}{p^i(\hat{\mathbf{y}}_c^i)} \\ &= C^i p^i(\hat{\mathbf{y}}_c^i|\mathbf{d}_c)p^i(\mathbf{d}_c), \end{aligned}$$

where C^i is the normalization constant which will be ignored in the calculations as it does not affect the order of the probabilities. The conditional distribution $p^i(\hat{\mathbf{y}}_c^i|\mathbf{d}_c)$ will be calculated from the channel characteristics assuming we have canceled out the interference completely. The marginal distribution $p^i(\mathbf{d}_c)$ is taken as

$$p^i(\mathbf{d}_c) = q^{i-1}(\mathbf{d}_c).$$

In the first iteration step, we consider uniform $p^1(\mathbf{d}_c)$. In the next iterations, $p^i(\mathbf{d}_c)$ is updated to be $q^{i-1}(\mathbf{d}_c)$, the posterior distribution of iteration $i - 1$. The distribution $q^{i-1}(\mathbf{d}_c)$ reflects our updated belief about the coded bit distribution at the end of iteration $i - 1$. Therefore we are using the results of the

previous iteration step in two ways: first to subtract off the interfering users, second to update our prior distribution about the user of concern.

The estimate $\hat{\mathbf{d}}_c^i$ is then chosen according to

$$\hat{\mathbf{d}}_c^i = \arg \max_{\mathbf{d}_c} q^i(\mathbf{d}_c). \quad (7)$$

In the scheme explained, estimates of coded bits of all users in iteration step i are obtained in a parallel fashion. Instead, we can also follow a serial approach where we start from user 1 and decode users sequentially. In that case for iteration step i , we can immediately make use of the current estimates $\hat{\mathbf{d}}_c^i$ for users $c+1, \dots, K$ and have possibly fewer number of iterations.

Now let us concentrate on (6) to point out some of the differences between the schemes in Figure 1. In the “2stage + trellis” scheme, $\hat{\mathbf{d}}_c$ is found as in (7). However the prior $p^i(\mathbf{d}_c)$ is always assumed to be uniform. The simulation results suggest that updating the prior is actually helpful. Multistage detection with hard decisions and Viterbi decoding scheme (hard 2stage + conv) ignores the constraint $\mathcal{I}(\mathbf{d}_c \in \mathcal{C}_c)$ in equation (6) and uses a uniform prior $p^i(\mathbf{d}_c)$ to obtain a hard decision for the coded bits. Then the information is extracted using Viterbi decoding. The “MF + MAP” decoding ignores the constraint $\mathcal{I}(\mathbf{d}_I = \hat{\mathbf{d}}_I^{i-1})$, assumes uniform prior and uses y_c , the matched filter output, to estimate \mathbf{d}_c . The interference from other users is ignored. There is essentially a single iteration step. This in fact constitutes the first step in our iterative algorithm and thus “MF + MAP” gives inferior results to our scheme. We should also note here that in Schlegel’s proposed iterative scheme [12], the prior distribution is updated in a similar manner, but the aim is to minimize the probability of symbol error as opposed to the probability of sequence error. Also the previous decisions on \mathbf{d}_I are not fed back, which we believe results in a performance degradation.

V. CONCLUSION

In this paper we have considered a joint detection and decoding scheme to extract the information bits in a DS-CDMA system. The optimum algorithm is exponential in complexity in the number of users. An iterative multistage detection combined with the single user MAP algorithm is described. The multiuser detection scheme provides a MAI free statistic which is used for the single user decoding. This technique helps us reduce the complexity of the receiver without sacrificing performance significantly. We have shown comparative studies with some of the existing schemes. Our algorithm has always outperformed the other algorithms. We have also developed a framework to evaluate the various existing algorithms. In

this framework we can provide a quantitative difference in the performance of the various schemes under investigation and identify their drawbacks.

REFERENCES

- [1] S. Verdú, “Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels,” *IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 85–96, 1986.
- [2] A. Duel-Hallen, “Decorrelating Decision Feedback Multiuser Detector for Synchronous Code-Division Multiple-Access Channel,” *IEEE Transactions on Communications*, vol. 41, no. 2, pp. 285–290, Feb. 1993.
- [3] R. Lupas and S. Verdú, “Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channels,” *IEEE Transactions on Information Theory*, vol. 41, no. 4, pp. 123–136, 1989.
- [4] M. K. Varanasi and B. Aazhang, “Multistage Detection in Asynchronous Code Division Multiple Access Communications,” *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 509–519, Apr. 1990.
- [5] J. G. Proakis, *Digital Communications*, McGraw-Hill, 1995.
- [6] A. J. Viterbi, *CDMA Principles of Spread Spectrum Communication*, Addison-Wesley Publishing Company, 1995.
- [7] L.R. Bahl, J.Cocke, F. Jelinek, and J.Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, 1974.
- [8] T.R. Giallorenzi and S.G. Wilson, “Multiuser ML Sequence Estimator for Convolutional Coded Asynchronous System,” *IEEE Transaction on Communication*, vol. 44, no. 8, pp. 997–1008, Aug. 1996.
- [9] K. R. Narayanan and G. L. Stuber, “List Decoding of Turbo Codes,” *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 754–762, June 1998.
- [10] J.B. Anderson and S. Mohan, “Sequential coding algorithms: A survey and cost analysis,” *IEEE Transactions on Communications*, vol. COM-32, pp. 169–176, Feb. 1984.
- [11] U. Fawer and B. Aazhang, “Multiuser Receivers for Code-Division Multiple-Access Systems with Trellis-Based Modulation,” *IEEE Journal on Selected Area in Communications*, vol. 14, no. 8, pp. 1602–1609, 1996.
- [12] C.B. Schlegel, M.C.Reed, P.D. Alexander, and J.A. Asentorfer, “Reduced Complexiy Iterative Multiuser Detection for DS/CDMA with FEC,” in *Proceedings of ICUPC*, 1997.