Achieving Single-User Performance in a FEC-coded DS-CDMA system using the Low-Complexity Soft-Output M-Algorithm

Kitty K.Y. Wong and Peter J. McLane
Department of Electrical and Computer Engineering
Queen's University, Kingston, Ontario, Canada
Tel: +1 613 533 6274, Fax: +1 613 533 6615
Email: {wongk, mclane}@ee.queensu.ca

Abstract— **This paper presents a soft-output version of the M-algorithm (SOMA) to reduce the complexity of a turbo multiuser detector in a DS-CDMA system employing a forward error correction code. The SOMA reduces the complexity by pruning an equivalent tree that represents all possible transmit signal-vectors. It utilizes paths that are traversed but discarded in a pruned tree to reduce the total number of paths that need to be visited. We show a simple and effective method of using these discarded paths without having to store them. BER performance is presented for convolutional-coded CDMA system employing both random spreading sequences and Gold spreading sequences. Both the AWGN and Rayleigh fading channels are considered.**

I. Introduction

In a code-division multiple-access (CDMA) system, the presence of the multiple-access interference (MAI) limits the number of users that can simultaneously communicate over the channel. To overcome the detrimental effects of the MAI, it has been shown that with the use of multiuser detection techniques and a forward error correction (FEC) code, one can potentially improve the performance of a multiuser system to match that of a single-user system. The optimum receiver for such a system involves decoding a joint-trellis which combines multiuser detection and channel decoding. Its complexity is exponential in the product of the number of users and the constraint length of the code. This complexity prohibits the use of an optimal detector even for small systems. A popular alternative is to apply the turbo processing principle [1] which uses a soft-input soft-output (SISO) multiuser detector (MUD) and a SISO channel decoder in an iterative manner. The SISO-MUD computes and delivers the log-likelihood ratio (LLR) of each coded bit to the channel decoder of each user. Based on the code constraints, the SISO-decoders update this information and send them back to the SISO-MUD for refinement. This process is repeated until convergence is achieved. It has been shown that near-optimal performance can be achieved with the turbo processing technique.

In order to compute the LLRs, the SISO-MUD needs to calculate the *a posteriori* probability (APP) for each possible transmit-vector. For a K-user system, 2^K APPs need to be computed per transmission. Therefore, it is impractical to implement such a MUD when K is large.

In this paper, we propose a low-complexity algorithm, the soft-output M-algorithm (SOMA) [2], to reduce the complexity of the SISO-MUD by considering only a subset of all possible vectors. We first obtain an equivalent tree structure representing all possible vectors. Then, the tree is pruned based on the M-algorithm. Unlike existing algorithms such as LISS decoding [3] and the iterative tree search method [4], branches and paths that are traversed but pruned are also involved in the LLR computation. We will show a simple and effective method of using these discarded paths without having to store them. Unlike the LISS method, no metric adjustment is needed for non-fully extended paths. Simulation results show that by including these paths in the LLR computation, single-user performance can be achieved with a relatively small M in a few number of iterations.

In the paper, the transmitter model and the iterative receiver structure are shown in section II. The SOMA is presented in section III. Simulation results for the additive white Gaussian noise (AWGN) channel is presented in section IV. Simulation results for Rayleigh fading channel is presented in section V. Finally, concluding remarks are given in section VI.

II. System Model

The transmitter for a FEC-coded CDMA system with K users communicating over a common fading channel is schematically shown in Fig. 1. The binary input sequence $\mathbf{u}^k$ for each user k is first encoded with a convolutional encoder, followed by a random bit interleaver. The resulting sequence of interleaved bits $\mathbf{x}^k$ is spread by the user-specific spreading code and transmitted over a common fading channel. Let x_t^k be the transmitted bit for user k at time index t. Let $\mathbf{s}^k = [s_1^k, \cdots, s_N^k]$ be the spreading code employed by user k, where $s_n^k \in \{\pm 1/\sqrt{N}\}$ denotes the n-th chip of user k's spreading sequence and N is the spreading gain. Let a_t^k be the fading gain for user k at time index t. The fading amplitude is assumed to be constant for one symbol interval.

The received signal is the sum of K transmitted signals embedded in the AWGN. Using a discrete-time model, the received signal for time index t can be written as [5]:

$$\mathbf{y}_t = \mathbf{S}\,\mathbf{A}_t\,\mathbf{x}_t + \mathbf{w}_t,$$

where $\mathbf{y}_t = [y_{t,1} \cdots y_{t,N}]^T$ is the matched-filer output of the received signal for transmission index t sampled at every chip interval, $\mathbf{S}$ is the $N \times K$ spreading matrix with elements $\{\mathbf{S}\}_{n,k} = s_n^k$, $\mathbf{A}_t = diag\{a_t^1, \cdots, a_t^K\}$ is the $K \times K$ attenuation matrix, $\mathbf{x}_t = [x_t^1 \cdots x_t^K]^T$ is the transmitted vector of all users at time t, $\mathbf{w}_t = [w_{t,1} \cdots w_{t,N}]^T$ is the Gaussian noise vector with variance σ^2 per element and superscript T denotes matrix transpose.

Fig. 2 depicts the iterative receiver structure. At each iteration, the SISO-MUD updates and delivers to each channel decoder the extrinsic information for each user-bit. The channel decoders, employing the soft-output Viterbi algorithm [6], refine the extrinsic information and feed them back to the SISO-MUD for further processing. This iterative process continues until convergence is achieved.

The extrinsic value of each coded bit is given by:

$$L_e(x_t^k) = L(x_t^k) - L_a(x_t^k),$$

where $L(x_t^k)$ and $L_a(x_t^k)$ denote the LLR and the log-*a priori* value of bit x_t^k, respectively, $x_t^k \in \{0, 1\}$.

Using the linear model above, the LLR of each coded bit can be expressed as:

$$L(x_t^k) = \ln \frac{\sum_{\mathbf{x} \in \mathbb{X}_1^k} \exp\left(\frac{-||\mathbf{y}_t - \mathbf{S}\mathbf{A}_t\mathbf{x}||^2}{2\sigma^2} + \ln \Pr\{\mathbf{x}_t = \mathbf{x}\}\right)}{\sum_{\mathbf{x} \in \mathbb{X}_0^k} \exp\left(\frac{-||\mathbf{y}_t - \mathbf{S}\mathbf{A}_t\mathbf{x}||^2}{2\sigma^2} + \ln \Pr\{\mathbf{x}_t = \mathbf{x}\}\right)},$$

where $\mathbb{X}_a^k$, $a \in \{0, 1\}$, contains the set of sequences $\mathbf{x} = \{x^1, \cdots, x^k, \cdots, x^K\}$ with $x^k = a$, $1 \le k \le K$.

Applying the max-log approximation [7], the LLR becomes:

$$L(x_t^k) \approx \max_{\mathbf{x} \in \mathbb{X}_1^k} \left(\frac{-||\mathbf{y}_t - \mathbf{S}\mathbf{A}_t\mathbf{x}||^2}{2\sigma^2} + \ln \Pr\{\mathbf{x}_t = \mathbf{x}\}\right) - \max_{\mathbf{x} \in \mathbb{X}_0^k} \left(\frac{-||\mathbf{y}_t - \mathbf{S}\mathbf{A}_t\mathbf{x}||^2}{2\sigma^2} + \ln \Pr\{\mathbf{x}_t = \mathbf{x}\}\right). \quad (1)$$

The vector norm in (1) can be written as:

$$||\mathbf{y}_t - \mathbf{S}\mathbf{A}_t\mathbf{x}||^2 = (\mathbf{A}_t\mathbf{x} - \hat{\mathbf{x}}_t)^T \mathbf{S}^T \mathbf{S} (\mathbf{A}_t\mathbf{x} - \hat{\mathbf{x}}_t), \quad (2)$$

where $\hat{\mathbf{x}}_t = (\mathbf{S}^T\mathbf{S})^{-1}(\mathbf{S}^T\mathbf{y}_t) = (\mathbf{R})^{-1}(\mathbf{S}^T\mathbf{y}_t)$ is the decorrelator output at time index t, $\mathbf{R} = (\mathbf{S}^T\mathbf{S})$ is the cross-correlation matrix, and $(\mathbf{S}^T\mathbf{y}_t)$ is the despreader output. If $\mathbf{R}$ is positive-definite, it has a Cholesky factorization $\mathbf{R} = \mathbf{L}^T\mathbf{L}$, where $\mathbf{L}$ is a lower triangular matrix with dimension $K \times K$. Then, $(\mathbf{A}_t\mathbf{x} - \hat{\mathbf{x}}_t)^T \mathbf{S}^T \mathbf{S} (\mathbf{A}_t\mathbf{x} - \hat{\mathbf{x}}_t)$

$$= (\mathbf{A}_t\mathbf{x} - \hat{\mathbf{x}}_t)^T \mathbf{L}^T \mathbf{L} (\mathbf{A}_t\mathbf{x} - \hat{\mathbf{x}}_t)$$
$$= \sum_{k=1}^{K} \left(l_{kk}(a_t^k x^k - \hat{x}_t^k) + \sum_{j=1}^{k-1} l_{kj}(a_t^j x^j - \hat{x}_t^j) \right)^2, \quad (3)$$

where l_{kj} are elements of $\mathbf{L}$. Using the recursive structure in (3) and substituting (2) and (3) into (1):

$$\begin{aligned} L(x_t^k) \approx & \max_{\mathbf{x} \in \mathbb{X}_1^k} \left\{ \sum_{k=1}^{K} \left(-\frac{1}{2\sigma^2} \Big(l_{kk}(a_t^k x^k - \hat{x}_t^k) + \sum_{j=1}^{k-1} l_{kj}(a_t^j x^j - \hat{x}_t^j) \Big)^2 + x_t^k \cdot L_a(x_t^k) \right) \right\} \\ & - \max_{\mathbf{x} \in \mathbb{X}_0^k} \left\{ \sum_{k=1}^{K} \left(-\frac{1}{2\sigma^2} \Big(l_{kk}(a_t^k x^k - \hat{x}_t^k) + \sum_{j=1}^{k-1} l_{kj}(a_t^j x^j - \hat{x}_t^j) \Big)^2 + x_t^k \cdot L_a(x_t^k) \right) \right\} \\ = & \max_{\mathbf{x} \in \mathbb{X}_1^k} \Gamma_t^K(\mathbf{x}) - \max_{\mathbf{x} \in \mathbb{X}_0^k} \Gamma_t^K(\mathbf{x}), \text{ with} \end{aligned}$$

$$\begin{aligned} \Gamma_t^K(\mathbf{x}) &= \Gamma_t^{K-1}(\mathbf{x}_{[1,K-1]}) + \gamma_t^K(\mathbf{x}), \\ \Gamma_t^k(\mathbf{x}_{[1,k]}) &= \Gamma_t^{k-1}(\mathbf{x}_{[1,k-1]}) + \gamma_t^k(\mathbf{x}_{[1,k]}), \; 1 \le k \le K-1, \\ \Gamma_t^0(\cdot) &= 0, \text{ and} \\ \gamma_t^k(\mathbf{x}_{[1,k]}) &= \frac{-1}{2\sigma^2} \Big(l_{kk}(a_t^k x^k - \hat{x}_t^k) + \sum_{j=1}^{k-1} l_{kj}(a_t^j x^j - \hat{x}_t^j) \Big)^2 \\ & \quad + x^k \cdot L_a(x_t^k), \end{aligned}$$

where $\mathbf{x}_{[1,k]}$ denotes the first k elements of vector $\mathbf{x}$. Thus, (1) reduces to a decoding problem of a tree with depth K, 2 branches per tree node, and branch metrics $\gamma_t^i(\cdot)$. Each path in the tree represents a possible transmit-vector $\mathbf{x}$ with cumulative metric $\Gamma_t^i(\mathbf{x}_{[1,k]})$ at depth k and total path metric $\Gamma_t^K(\mathbf{x})$. Note that this tree-model derivation can be used for systems with complex modulation schemes. In such a case, the tree has Q branches per node where Q is the constellation size. Note that the matrix $\mathbf{L}$ has to be computed only once if the set of spreading sequences remains unchanged. However, the decorrelator output has to be computed for every time index.

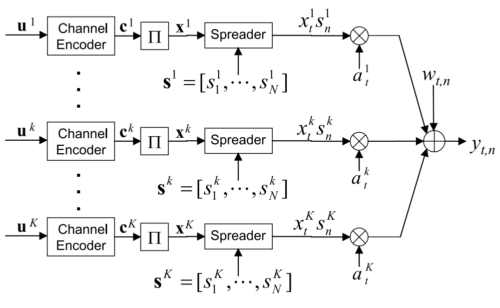

Fig. 1. Transmitter Description of a FEC-coded CDMA system.

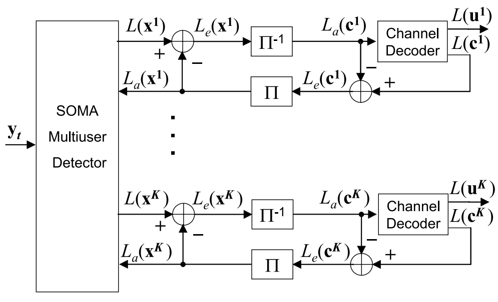

Fig. 2. Iterative Receiver Description of a FEC-coded CDMA system.

The above derivation assumes the cross-correlation matrix $\mathbf{R}$ is positive definite. If not, $\mathbf{R}$ can be replaced by the minimum-mean squared-error matrix $\mathbf{H} = (\mathbf{R} + \sigma^2\mathbf{I})$, where $\mathbf{I}$ is a $K \times K$ identity matrix. Since, $\mathbf{H}$ is positive definite, both its inverse and Cholesky decomposition exist [5]. Thus, $\mathbf{H} = \mathbf{L}^T\mathbf{L}$ and $\hat{\mathbf{x}}_t = (\mathbf{H})^{-1}\mathbf{S}^T\mathbf{y}_t$ are used instead.

Let $\tilde{\mathbf{x}}$ be the ML path that maximizes $\Gamma_t^K(\mathbf{x})$. Then,

$$L(x_t^k) = (2 \cdot \tilde{x}^k - 1) \cdot \left(\Gamma_t^K(\tilde{\mathbf{x}}) - \max_{\mathbf{x}: x^k \neq \tilde{x}^k} \Gamma_t^K(\mathbf{x}) \right).$$

Therefore, the LLR for bit x_t^k is given by the metric difference between the ML path and its best competitor with an opposite bit decision for depth (user) k. The SOMA searches selectively from the tree for these two paths and the algorithm is explained in the follow section.

III. The Soft-output M-algorithm

The SOMA reduces the complexity of tree-decoding as follows: at each tree depth, the algorithm extends only the M-best paths with the highest metric value to the next depth. It sorts the new set of paths according to their cumulative metric values, retains only the best M of them and discards the rest. Once a path is discarded, branches stemmed from the discarded nodes are removed. When the end of the tree is reached, the approximate ML (AML) path is the one with the highest metric value. The M-best paths that reach the end of the tree and all discarded paths are used for LLR computation.

Consideration of discarded paths is crucial in computing LLRs. When performing the M-algorithm on a large tree, the M-best fully-extended paths are similar to each other in the sense that they all have the same bit decisions as the AML path for a number of bit locations. Therefore, the LLR for these bits cannot be evaluated. One solution is to assign a constant value to the LLR of these bits [4]. However, when a large number of bits are involved, which is the case when a small value of M is used on a large tree, this method fails as most of the LLRs will result in the same constant assigned. Therefore, it is not sufficient to use only the M-best paths in the LLR computation. Discarded paths should also be considered [2].

To use the discarded paths in the LLR computation efficiently, the following two approximations are made:

1) The metric difference between the AML path $\tilde{\mathbf{s}}$ and a discarded path $\bar{\mathbf{s}}_{[1,j]}$ is approximated by subtracting the metric of the shorter path from the cumulative metric of the AML path at the depth where the shorter path is discarded:

$$\Gamma_t^{N_T}(\tilde{\mathbf{s}}) - \Gamma_t^{N_T}(\bar{\mathbf{s}}) \approx \Gamma_t^j(\tilde{\mathbf{s}}_{[1,j]}) - \Gamma_t^j(\bar{\mathbf{s}}_{[1,j]}) \quad (4)$$

This simple approximation gives a good indication of how far the discarded path is away from the AML one. More importantly, it avoids the need of metric adjustment for the length difference between the two paths.

2) To avoid the need of storing all discarded paths, (4) is further approximated by the metric difference between the first-ranked path at the depth where the shorter path is discarded and the metric of the shorter path.

Let $\Delta_a(k)$, $a \in \{0, 1\}$, be two arrays that keep track of the smallest metric difference between the AML path and a competitor with bit decision a at depth k, $1 \leq k \leq K$. With the above approximations, $\Delta_a(k)$ can be easily obtained as follows: for each tree depth k, $1 \leq k \leq K$,

1. Extend each of the M retained nodes to the next level.
2. Rank all extended paths. Let path r represents the r-th ranked path with cumulative metric Γ_r.
3. Initialize $\Delta_1(k) = \Delta_0(k) = \infty$.
4. For each path r with $r > M$:
   - Let $(x_r)^j$ be the bit decision for path r at depth j, $1 \leq j \leq k$.
   - For each j, $1 \leq j \leq k$, update $\Delta_a(j)$ as:

$$\Delta_a(j) = \min\left\{\Delta_a(j),\ \Gamma_1 - \Gamma_r\right\} \quad \text{for } (x_r)^j = a.$$

When the end of the tree is reached, update $\Delta_a(k)$ with the M-best fully-extended paths. Then,

$$L(x_t^k) \approx (2 \cdot \tilde{x}^k - 1) \cdot \Delta_{1-\tilde{x}^k}(k).$$

IV. Simulation Results for AWGN channels

This section presents the simulation results for using SOMA in FEC-coded CDMA systems in AWGN channels. The simulation parameters are adapted from [3]. Blocks of 512 information bits are encoded with a rate $1/2$, 16-state, convolutional code with generator polynomials [46,66] in octal. Random bit interleavers of sizes 1024 are used. For spreading, both random codes and Gold codes are considered. For the case of random spreading, a new set of spreading codes is generated for each block and remains constant for the entire block. For the Gold case, the same set of codes is used throughout the entire transmission.

A. Random Spreading codes

Fig. 3 shows the performance of the SOMA with $M = 32$ in a 32-user system employing random spreading codes with spreading gain of 32. The result from LISS decoding is also plotted for this case. With $M = 32$, the SOMA achieves the single-user performance bound in 6 iterations at a minimum signal-to-noise ratio (SNR) of 4 dB. LISS decoding requires 18 iterations with two stacks of sizes 1024 for the main stack and 512 for the auxiliary stack to achieve the same performance. Note that the SOMA is less complex than LISS decoding for this case. The auxiliary stack for LISS decoding that computes $\ln \Pr\{\mathbf{y}_t\}$ is essentially a M-algorithm with M equals to the stack size [3]. Thus, the implementation of the auxiliary stack alone is more complex than SOMA with $M = 32$. Fig. 4 illustrates the performance of the SOMA after 6 iterations in a 64-user system with spreading gain of 64. Due to the large amount of MAI caused by the large number of simultaneous users, $M = 256$ is needed to achieve the single-user performance at a minimum SNR of 4 dB.

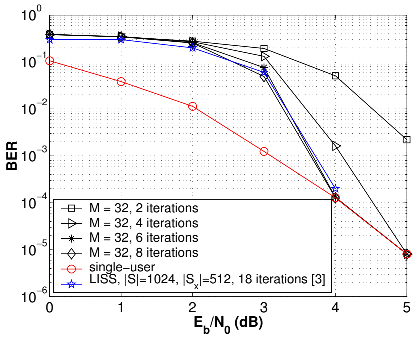

Fig. 3. Performance of the SOMA with $M = 32$ in a 32-user system using random spreading codes with spreading gain of 32 in an AWGN channel.

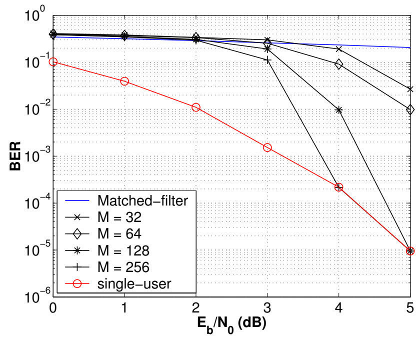

Fig. 4. Performance of the SOMA 64-user FEC-coded CDMA system using random spreading codes with spreading gain of 64 in an AWGN channel.

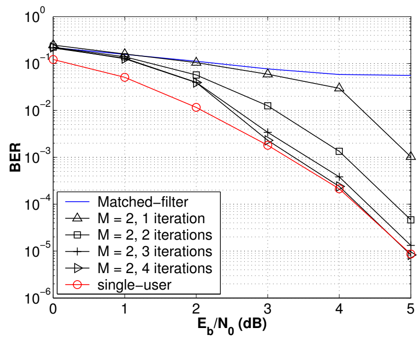

Fig. 5. Performance of the SOMA in a 9-user FEC-coded CDMA system using Gold spreading codes with spreading gain of 7 in an AWGN channel.

B. Gold Codes

The set of Gold codes with degree r has $(2^r + 1)$ distinct sequences with length $(2^r - 1)$. Thus, the spreading gain of CDMA systems employing these codes are restricted to be $(2^r - 1)$ and can support at most $(2^r + 1)$ users. We will show the simulation results for cases with small spreading gain and large spreading gain. Fig. 5 shows the performance of the SOMA in a 9-user system employing Gold sequences with spreading gain of 7. Due to the good correlation properties of Gold codes, each user experiences a reduced amount of MAI as compared to the random spreading case. This is verified by the performance improvement of the matched-filter receiver over the case with random spreading. The single-user performance bound is achieved by using $M = 2$ in 4 iterations at a minimum SNR of 3 dB. Fig. 6 shows the performance of the SOMA with $M = 2$ in a 65-user system employing Gold sequences with spreading gain of 63. Due to the large spreading gain, the single-user performance bound is achieved by using $M = 2$ in only 2 iterations even for the low-SNR region. LISS decoding (not plotted) requires a stack of size 128 in 5 iterations to achieve the single-user performance for the high-SNR region. Note that SOMA for this case is also less complex than LISS decoding. The total number of branches traversed by SOMA is approximately $(2 \cdot M \cdot K) \cdot I = 520$, where $(2 \cdot M)$ is the number of branches extended per tree level, K is the tree depth and I is the number of iterations. The number of branches traversed by LISS decoding with stack size $|S|$ can be computed as follows: when the root node is extended, two branches are traversed and two path entries are inserted into the stack. To fill up the remaining stack, at least $\left(2 \cdot (|S| - 2)\right)$ branches has to be traversed since one additional entry is created when the top path in the stack is extended. Thus, the number of branches traversed by LISS decoding is at least $\left(2 + 2 \cdot (|S| - 2)\right) \cdot I = 1270$.

V. Results for Rayleigh Fading Channels

This section presents the simulation results for using SOMA in FEC-coded CDMA systems in Rayleigh fading channels. The fading gain is assumed to be constant for each symbol interval and varies independently from one symbol to the next. Perfect channel knowledge is assumed at the receiver.

A. Random Spreading codes

Fig. 7 illustrates the results of SOMA after 4 iterations for a 32-user system employing random spreading codes with spreading gain of 32. Single-user performance is obtained with either $M = 16$ at a minimum SNR of 8 dB or $M = 32$ at a minimum SNR of 6 dB. To further improve the performance of the SOMA, one can utilize the known channel information to rearrange the structure of the tree such that users with larger fading amplitudes are detected first. Note that the penalty is to recompute the matrix $\mathbf{L}$ for every new transmission. Fig. 8 shows the new result of the SOMA after 4 iterations. Single-user performance is obtained with either $M = 8$ at a minimum SNR of 8 dB or $M = 16$ at a minimum SNR of 6 dB. LISS decoding (not plotted) requires two stacks of sizes 256 for the main stack and 128 for the auxiliary stack in 5 iterations to achieve the bound at 6 dB. Fig. 9 shows the result of the 64-user case with spreading gain of 64 using the SOMA. $M=16$ is sufficient.

B. Gold Codes

Simulation results for Gold codes in Rayleigh fading channel are shown in Fig. 10 and 11. For both the 9-user case with spreading gain of 7 and the 65-user case with spreading gain of 63, single-user performance is achieved with $M=2$.

VI. Conclusions

A low-complexity soft-output tree decoding algorithm is proposed for DS-CDMA systems employing FEC. The al-

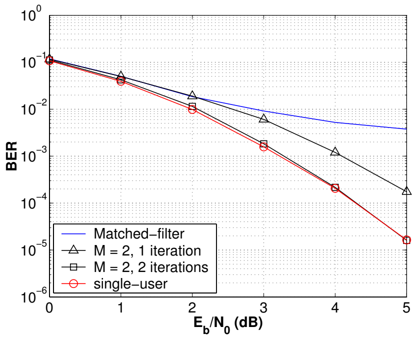

Fig. 6. Performance of the SOMA in a 65-user FEC-coded CDMA system using Gold spreading codes with spreading gain of 63 in an AWGN channel.

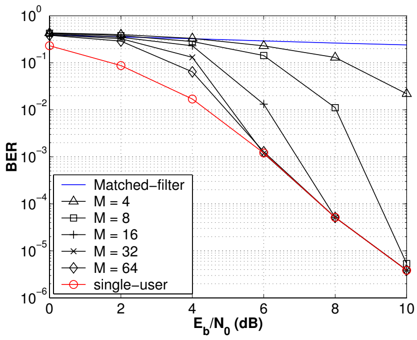

Fig. 7. Performance of the SOMA in a 32-user system using random spreading codes with spreading gain of 32 in a Rayleigh Fading channel. 4 iterations are performed.

gorithm utilizes discarded paths in a pruned tree to reduce the total number of traversed branches. The complexity of the SOMA increases linearly with M, the number of retained paths per depth. The algorithm is evaluated in both the AWGN and Rayleigh fading channels using both random spreading codes and Gold spreading codes. Simulation results show near-optimal performance can obtained using the SOMA with a relatively small M in a few iterations.

Acknowledgement

This work is supported by NSERC of Canada and the Bell Mobility/Samsung Grant on Smart Antennas at Queen's University.

References

[1] J. Hagenauer, "The Turbo Principle: Tutorial Introduction and state of the art," in *Proc. Int. Sym. on Turbo Codes*, pp. 1-11, Sep. 1997.
[2] K. Wong, and P. McLane, "Bi-directional soft-output M-algorithm for iterative decoding," in *Proc. ICC'04*, vol. 2, pp. 792-797, June 2004.
[3] C. Kuhn, and J. Hagenauer, "Iterative List-Sequential (LISS) Detector for Fading Multiple-Access Channels," in *Proc. GLOBECOM'04*, vol. 1, pp. 330-335, Nov. 2004.
[4] Y. de Jong, and T. Willink, "Iterative Tree Search Detection for MIMO Wireless Systems," in *Proc. VTC'02*, vol. 2, pp. 1041-1045, Sep. 2002.
[5] S. Verdú, "*Multiuser Detection*", Cambridge University Press, 1998.
[6] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its applications," in *Proc. GLOBECOM'89*, vol. 3, pp. 1680-1686, Nov 1989.
[7] P. Robertson, E. Villebrun, and P. Hoeher, "A Comparison of Optimal and Suboptimal MAP Decoding Algorithms Operating in the Log Domain," in *Proc. ICC'95*, vol. 2, pp. 1009-1013, Jun. 1995.

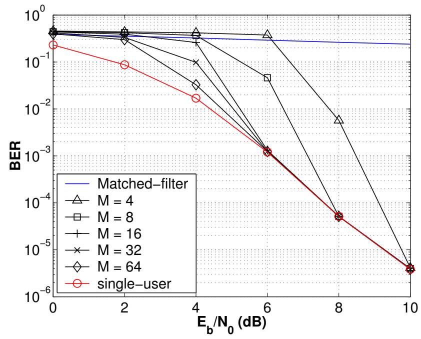

Fig. 8. Performance of the SOMA that utilizes the channel information in a 32-user system using random spreading code with spreading gain of 32 in a Rayleigh Fading channel. 4 iterations are performed.

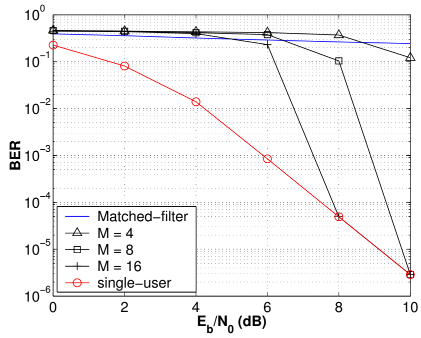

Fig. 9. Performance of the SOMA that utilizes the channel information in a 64-user system using random spreading code with spreading gain of 64 in a Rayleigh Fading channel. 4 iterations are performed.

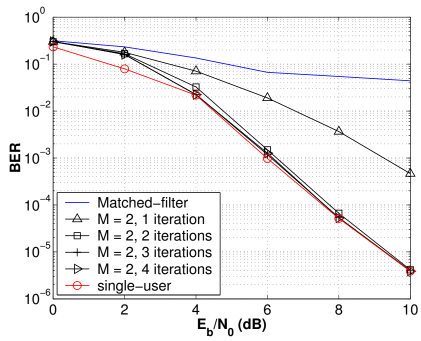

Fig. 10. Performance of the SOMA in a 9-user system using Gold spreading codes with spreading gain of 7 in a Rayleigh Fading channel.

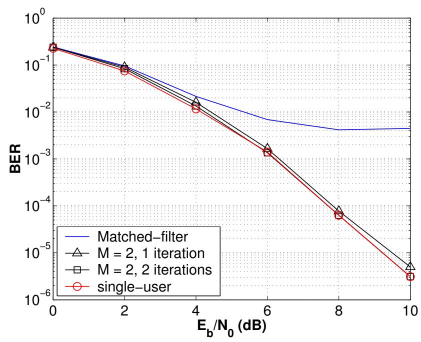

Fig. 11. Performance of the SOMA in a 65-user system using Gold spreading codes with spreading gain of 63 in a Rayleigh Fading channel.