A Low-Complexity Iterative MIMO Detection Scheme Using the 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) for complexity reduction in an iterative MIMO receiver. Unlike existing algorithms, SOMA utilizes paths that are traversed but discarded in a pruned tree to reduce complexity. It is shown that discarded paths in a heavily pruned tree carry a significant amount of soft-information which should not be ignored. We show a simple and effective method of using these paths without having to store them. BER performance is presented for both convolutional- and turbo-coded MIMO systems that use bit-interleaved coded-modulation. Flat fading with known channel state information is assumed. Improvement is observed in using soft-information from discarded paths for a convolutional outer code. For a powerful turbo outer code, assigning a constant to unknown likelihoods suffices.**

I. Introduction

Multiple-input Multiple-output (MIMO) transmission techniques have received a great deal of attention due to their ability to provide higher spectral efficiency than conventional single-input single-output systems. To detect symbol vectors transmitted over a multiple-antenna system, the optimal maximum-likelihood (ML) approach is to examine all possible transmit vectors to find the most probable one. If a forward error-correction (FEC) code is used, one also needs to compute the log-likelihood ratio (LLR) for each coded bit by examining the *a posteriori* probability of each of these vectors. The number of hypotheses to examine is Q^{N_T}, where Q is the constellation size and N_T is the number of transmit antennae. When a high-order modulation scheme such as 64-QAM is used, ML decoding becomes prohibitively complex, even for systems with a small number of transmit antennae.

A solution to the complexity issue is to examine only a subset of all hypotheses. This can be achieved by first obtaining an equivalent tree structure for the problem [1] and then applying a sequential decoding technique [2] on the tree to obtain the desired subset. Sphere decoding [3], LISS decoding [4] and iterative tree search (ITS) [5] are some examples of this technique. These algorithms are all common in the sense that they first obtain a list of paths by pruning the equivalent tree based on their own pruning criterion. Then, LLRs are computed based solely on the obtained list. Paths that are traversed but pruned do not play a role in the LLR computation. In sphere decoding, the list contains a set of some best X paths whose metric values are smaller than some chosen radius r. This list is fixed from one iteration to the next. However, metric values of paths on the list are updated at each iteration by incorporating new *a priori* information. In LISS decoding, the stack algorithm is used for tree pruning. When the end of the tree is reached, only paths in the stack are used in the LLR computation. For this method, adjustment is needed for the metric value of each non-fully extended path in the stack. In the ITS, the M-algorithm is used for tree pruning. When the end of the tree is reached, only the M-best fully-extended paths are used in the LLR computation. Our algorithm, the soft-output M-algorithm (SOMA) [6], also prunes the code tree using the M-algorithm. However, branches and paths that are traversed but pruned are also involved in the LLR computation. We will show a simple and effective method for using these discarded paths without having to store them. Unlike the LISS method, no metric adjustment is needed. Simulation results show by including these paths, a much smaller M can be used to achieve the same BER performance obtained by the ITS for the case with a convolutional outer code, thereby reducing the overall complexity.

In the paper, the system model and the derivation of the LLR are shown in section II. The SOMA is described in section III. Simulation results for convolutional-coded MIMO systems are given in section IV. The complexity analysis of the SOMA is presented in section V. Simulation results for turbo-coded MIMO systems are presented in section VI. Finally, concluding remarks are given in section VII.

II. System Model

We consider the transmission of FEC-coded symbols over a general MIMO channel with N_T transmit and N_R receive antennae as depicted in Fig. 1 and 2.

A binary input sequence $\mathbf{u}$ is encoded with either a convolutional or turbo encoder, followed by a random bit-interleaver [8]. The resulting sequence of interleaved bits, $\mathbf{x}$, is mapped

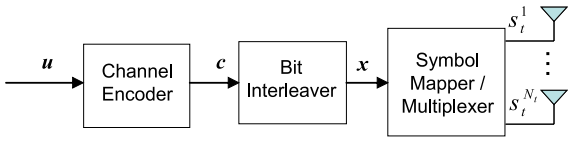

Fig. 1. Transmitter Description of a FEC-coded MIMO system.

into Q-QAM symbols based on the Gray rule. The sequence of QAM symbols, $\mathbf{s}$, is multiplexed onto each transmit antenna in a round-robin fashion, i.e., no space-time coding is considered.

The received complex symbol vector at each time index t can be described using a linear model as:

$$\mathbf{y}_t = \mathbf{H}\,\mathbf{s}_t + \mathbf{n}_t,$$

where $\mathbf{y}_t = [y_t^1 \cdots y_t^{N_R}]^T$ is the received vector at time t, $\mathbf{H}$ is the channel propagation matrix of dimension $N_R \times N_T$, $\mathbf{s}_t = [s_t^1 \cdots s_t^{N_T}]^T$ is the transmit vector at time t, and $\mathbf{n}_t$ is a vector of independent zero-mean complex Gaussian noise entries with variance N_0. Each element in $\mathbf{H}$ is modeled as an independent complex Gaussian random variable with variance 1/2 per dimension. As such a flat fading model is assumed. Fig. 2 depicts the iterative receiver structure based on the turbo-processing principle [9]. At each iteration, the MIMO detector updates and delivers to the channel decoder the extrinsic information for each coded bit. The channel decoder, employing the BCJR algorithm [10], refines the extrinsic information provided by the MIMO detector and feeds them back to the MIMO detector for further processing. This iterative process continues until convergence is achieved.

The extrinsic value of each coded bit is given by:

$$L_e(x_{t,k}^i) = L(x_{t,k}^i) - L_a(x_{t,k}^i),$$

where $L(x_{t,k}^i)$ and $L_a(x_{t,k}^i)$ denote the LLR and the log-*a priori* value of bit $x_{t,k}^i$, respectively. Bit $x_{t,k}^i$, $x_{t,k}^i \in \{0,1\}$, denotes the k-th bit of a K-bit vector whose corresponding 2^K-QAM symbol s_t^i is transmitted on antenna i at time t, with $s_t^i = map\big((x_{t,1}^i, \cdots, x_{t,K}^i)\big)$, $map(\,\cdot\,)$ denotes the function that Gray-maps a K-bit vector into a 2^K-QAM symbol. The LLR of each coded bit can be expressed as:

$$L(x_{t,k}^i) = \ln \frac{\sum_{\mathbf{x}\in\mathbb{X}_1^{i,k}} \exp\left(\frac{-||\mathbf{y}_t - \mathbf{H}\mathbf{s}||^2}{N_0} + \ln \Pr\{\mathbf{x}_t = \mathbf{x}\}\right)}{\sum_{\mathbf{x}\in\mathbb{X}_0^{i,k}} \exp\left(\frac{-||\mathbf{y}_t - \mathbf{H}\mathbf{s}||^2}{N_0} + \ln \Pr\{\mathbf{x}_t = \mathbf{x}\}\right)},$$

where $\mathbb{X}_a^{i,k}$, $a \in \{0,1\}$, contains the set of sequences $\mathbf{x} = \Big\{(x_1^1, \cdots, x_K^1), \cdots, (x_1^{N_T}, \cdots, x_K^{N_T})\Big\}$ with $x_k^i = a$ and $\mathbf{s} = [s^1 \cdots s^{N_T}]^T$, with $s^i = map\big((x_1^i, \cdots, x_K^i)\big)$.

Applying the max-log approximation [11], the LLR becomes:

$$\begin{aligned} L(x_{t,k}^i) \approx & \max_{\mathbf{x}\in\mathbb{X}_1^{i,k}} \left(\frac{-||\mathbf{y}_t - \mathbf{H}\mathbf{s}||^2}{N_0} + \ln \Pr\{\mathbf{x}_t = \mathbf{x}\}\right) - \\ & \max_{\mathbf{x}\in\mathbb{X}_0^{i,k}} \left(\frac{-||\mathbf{y}_t - \mathbf{H}\mathbf{s}||^2}{N_0} + \ln \Pr\{\mathbf{x}_t = \mathbf{x}\}\right). \quad (1) \end{aligned}$$

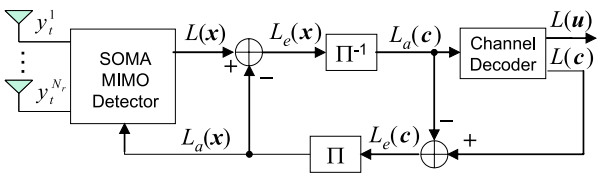

Fig. 2. Iterative receiver for a FEC-coded MIMO system.

The vector norm in (1) can be expressed as [3]:

$$\begin{aligned} &||\mathbf{y}_t - \mathbf{H}\mathbf{s}||^2 = \\ &\quad (\mathbf{s} - \hat{\mathbf{s}}_t)^\dagger \mathbf{H}^\dagger \mathbf{H}(\mathbf{s} - \hat{\mathbf{s}}_t) - \mathbf{y}_t(\mathbf{I} - \mathbf{H}(\mathbf{H}^\dagger\mathbf{H})^{-1}\mathbf{H}^\dagger)\mathbf{y}_t, \end{aligned} \quad (2)$$

where $\hat{\mathbf{s}}_t = (\mathbf{H}^\dagger\mathbf{H})^{-1}\mathbf{H}^\dagger\mathbf{y}_t$ and $\dagger$ denotes complex conjugate transpose. For $N_R \geq N_T$, the matrix $\mathbf{H}^\dagger\mathbf{H}$ is positive-definite. It has a Cholesky factorization $\mathbf{H}^\dagger\mathbf{H} = \mathbf{L}^\dagger\mathbf{L}$ where $\mathbf{L}$ is a lower triangular matrix with dimension $N_T \times N_T$. Then,

$$\begin{aligned} &(\mathbf{s} - \hat{\mathbf{s}}_t)^\dagger \mathbf{H}^\dagger \mathbf{H}(\mathbf{s} - \hat{\mathbf{s}}_t) \\ &\qquad = \sum_{i=1}^{N_T} \left| l_{ii}(s^i - \hat{s}_t^i) + \sum_{j=1}^{i-1} l_{ij}(s^j - \hat{s}_t^j) \right|^2, \end{aligned} \quad (3)$$

where l_{ij}, $1 \leq i, j \leq N_T$, are elements of $\mathbf{L}$. Using the recursive structure in (3) and substituting (2) and (3) into (1):

$$\begin{aligned} L(x_{t,k}^i) \approx & \max_{\mathbf{x}\in\mathbb{X}_1^{i,k}} \left\{ \sum_{i=1}^{N_T} \left(\frac{-1}{N_0} \left| l_{ii}(s^i - \hat{s}_t^i) + \sum_{j=1}^{i-1} l_{ij}(s^j - \hat{s}_t^j) \right|^2 \right.\right. \\ & \left.\left. + \sum_{k=1}^{K} x_k^i \cdot L_a(x_{t,k}^i) \right) \right\} - \\ & \max_{\mathbf{x}\in\mathbb{X}_0^{i,k}} \left\{ \sum_{i=1}^{N_T} \left(\frac{-1}{N_0} \left| l_{ii}(s^i - \hat{s}_t^i) + \sum_{j=1}^{i-1} l_{ij}(s^j - \hat{s}_t^j) \right|^2 \right.\right. \\ & \left.\left. + \sum_{k=1}^{K} x_k^i \cdot L_a(x_{t,k}^i) \right) \right\} \\ = & \max_{\mathbf{x}\in\mathbb{X}_1^{i,k}} \Gamma_t^{N_T}(\mathbf{s}) - \max_{\mathbf{x}\in\mathbb{X}_0^{i,k}} \Gamma_t^{N_T}(\mathbf{s}), \text{ with} \end{aligned}$$

$$\begin{aligned} \Gamma_t^{N_T}(\mathbf{s}) &= \Gamma_t^{N_T-1}(\mathbf{s}_{[1,N_T-1]}) + \gamma_t^{N_T}(\mathbf{s}), \\ \Gamma_t^{i}(\mathbf{s}_{[1,i]}) &= \Gamma_t^{i-1}(\mathbf{s}_{[1,i-1]}) + \gamma_t^{i}(\mathbf{s}_{[1,i]}),\ 1 \leq i < N_T, \\ \Gamma_t^{0}(\cdot) &= 0, \text{ and} \\ \gamma_t^i(\mathbf{s}_{[1,i]}) &= -\frac{1}{N_0} \left| l_{ii}(s^i - \hat{s}_t^i) + \sum_{j=1}^{i-1} l_{ij}(s^j - \hat{s}_t^j) \right|^2 \\ &\quad + \sum_{k=1}^{K} x_k^i \cdot L_a(x_{t,k}^i), \end{aligned}$$

where $\mathbf{s}_{[1,i]}$ denotes the first i elements of vector $\mathbf{s}$. Thus, (1) reduces to a decoding problem of a tree with depth N_T, Q branches per tree node, and branch metric $\gamma_t^i(\cdot)$. Each path in the tree represents a possible transmit vector $\mathbf{s}$ with cumulative metric $\Gamma_t^i(\mathbf{s}_{[1,i]})$ at depth i and total path metric $\Gamma_t^{N_T}(\mathbf{s})$.

Let $\tilde{\mathbf{s}} = map(\tilde{\mathbf{x}})$ be the ML path that maximizes $\Gamma_t^{N_T}(\mathbf{s})$. Then the LLR becomes:

$$L(x_{t,k}^i) = (2 \cdot \tilde{x}_k^i - 1) \cdot \left(\Gamma_t^{N_T}(\tilde{\mathbf{s}}) - \max_{\mathbf{x}: x_k^i \neq \tilde{x}_k^i} \Gamma_t^{N_T}(map(\mathbf{x})) \right).$$

Therefore, the LLR for bit $x_{t,k}^i$ is given by the metric difference between the ML path and its best competitor with an opposite bit decision for bit location k at depth (antenna) i.

III. The Soft-output M-algorithm

The SOMA prunes the code tree based on the M-algorithm. At each tree depth, only the M-best paths with the highest metric values are extended to the next depth. It sorts the new set of paths according to their cumulative metric values, retains the best M of them, and discards the rest. When the end of the tree is reached, the approximate ML (AML) path is the one with the highest metric value. The M-best fully-extended paths and all of the discarded paths are used in the 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 bit locations cannot be evaluated. One solution is to assign a constant value to the LLR of these bits [5]. However, when a large number of bits are involved, which is the case when a small 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 [6].

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:

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

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 $\Delta_a(i,k)$, $a \in \{0,1\}$, be two matrices whose (i,k) element keeps track of the smallest metric difference between the AML path and a competitor with decision a at depth i for bit location k, $1 \le i \le N_T$, $1 \le k \le K$. With the above approximations, $\Delta_a(i,k)$ can be easily obtained as follows: for each tree depth i, let all of the extended paths be ranked such that path r represents the r-th ranked path with cumulative metric Γ_r. For each path r with $r > M$:

- Let $(x_r)_k^j$ be the bit decision for path r at depth j and bit position k, $1 \le j \le i$, $1 \le k \le K$.
- For each j and k, update $\Delta_a(j,k)$ as:

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

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

$$L(x_{t,k}^i) \approx (2 \cdot \tilde{x}_k^i - 1) \cdot \Delta_{1-\tilde{x}_k^i}(i,k).$$

IV. Results for Convolutional-Coded Systems

Simulation results for convolutional-coded MIMO systems with $N_T = N_R = 4$ and $N_T = N_R = 8$, transmitting 4-QAM, 16-QAM, and 64-QAM symbols, are presented in this section. The parameters used in the simulations are based on [7]. Blocks of $130K$ information bits are encoded with a rate-$1/N_T$, 4-state convolutional code with generator polynomials $[5, 7, \cdots, 7]$ in octal. Bits from the i-th polynomial are sent to the i-th antenna. N_T random interleavers of length $130K$ are used prior to Gray-mapping to QAM symbols. The number of bits transmitted per channel use is $K = \log_2(Q)$, regardless of the number of antennae used. The channel gain coefficients are constant for each block and vary from one block to the next. Perfect channel knowledge is assumed at the receiver. The average signal-to-noise ratios (SNRs) used for simulations is defined as:

$$\text{SNR} = \frac{E_s}{N_0} \sum_{j=1}^{N_R} \sum_{i=1}^{N_T} E\{|s_t^i|^2\} \cdot E\{|h_{j,i}|^2\} = \frac{E_s N_T N_R}{N_0}.$$

The BER performance of the SOMA for a 4×4 channel are shown in Fig. 3. Eight iterations are performed. For the 4-QAM case, the performance of full-complexity decoding with no pruning ($M = 256$) can be achieved by the SOMA with $M = 4$. For the 16-QAM case, increasing M beyond 16 does not provide any performance gain. For the 64-QAM case, no significant improvement is observed by increasing M beyond 64. Fig. 4 shows the results of a 8×8 channel. For the 4-QAM case, no improvement is observed by increasing M beyond 8. For the 16-QAM and the 64-QAM cases, no gain is observed by increasing M beyond 16 and 32, respectively.

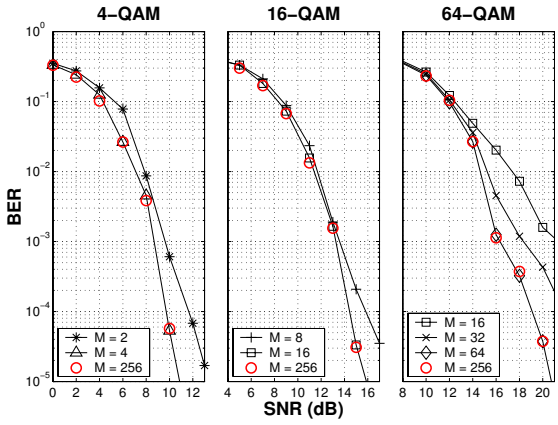

Fig. 3. Bit-error performance of the SOMA in a 4×4 MIMO system transmitting 4-QAM, 16-QAM, and 64-QAM symbols. Rate-1/4 Convolutional Code is used. The transmission rate is K bits/channel use for 2^K-QAM.

V. Complexity Reduction of the SOMA

In general, the SOMA involves the following operations:

1. Branch metric calculation: the number of branches extended per tree depth is (MQ), giving approximately $(N_T\, M\, Q)$ branch metric calculations in total.
2. Sorting: sorting of (MQ) elements is needed per depth.

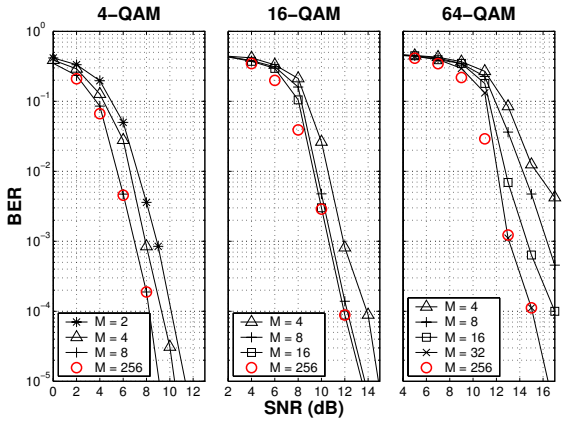

Fig. 4. Bit-error performance of the SOMA in a 8×8 MIMO system transmitting 4-QAM, 16-QAM, and 64-QAM symbols. Rate-1/8 Convolutional Code is used. The transmission rate is K bits/channel use for 2^K-QAM.

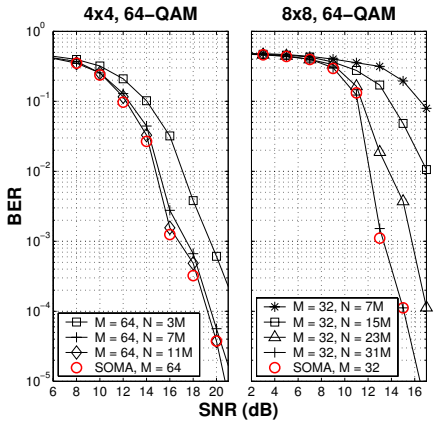

Fig. 5. Bit-error performance of the SOMA in a 4×4 and a 8×8 systems transmitting 64-QAM symbols when only a subset of discarded paths are used for the LLR calculation. N = number of discarded path used per tree depth.

3. Update of $\Delta_a(i,k)$: for each discarded path at depth i, entries in $\Delta_a(i,k)$ are compared with new values for (iK) times. Therefore, the total number of comparisons is $\sum_{i=1}^{N_T}(QM-M)\,i\,K \approx \frac{1}{2}M(Q-1)\,K\,N_T{}^2$.

A. reducing the number of comparisons

Since it is unlikely that a low-ranked discarded path can survive the minimization operation in the $\Delta_a(i,k)$-update step, a simple method to reduce the number of comparison operations is to consider only the best N discarded paths per tree depth with $N < (M(Q-1))$. Fig. 5 shows the performance of the SOMA using a reduced number of discarded paths. The figure on the left is for a 4×4 system transmitting 64-QAM symbols with $M = 64$. No loss is observed by using only 17% of the total. The figure on the right shows the performance of a 8×8 system transmitting 64-QAM symbols with M = 32. The loss to the no-reduction case is quite significant. One has to use at least 50% of all discarded paths to achieve the no-reduction performance. This result validates that discarded paths of a pruned tree can carry a significant amount of soft-information which should not be ignored. Note that with $N = 0$, the SOMA is equivalent to the ITS.

B. reducing the number of branch metric calculations

The number of branch metric calculations and the amount of sorting effort can be reduced by transmitting QAM-signals with a multi-level bit mapping property [5]. A 2^K-QAM constellation with the such property can be partitioned into 4 equal subsets where each subset can be uniquely identified by the first 2 bits of its signal points. The remaining (K-2) bits of signal points residing in the same subset forms another $2^{(K-2)}$-QAM constellation with multi-level bit mapping property. Therefore, instead of detecting each QAM symbol as a whole, it is possible to detect the symbol 2 bits at a time. The first bit-pair is decided by selecting the subset whose center of gravity best matches the received signal. The subsequent pair is then detected by partitioning the selected subset into 4 smaller subsets and again select the one whose center of gravity best matches the received signal. Since a 2^K-QAM signal contains $\frac{K}{2}$ bit-pairs, an equivalent tree with tree depth $(\frac{K}{2}N_T)$ and 4 branches per node can be constructed. Therefore, the total number of branches traversed is reduced from $(N_t\,M\,Q)$ to $(2K\,N_t\,M)$. The number of discarded path per tree depth is reduced from $M(Q-1)$ to $3M$. Branch metric derivation for signals with the above property can be found in [5].

Figures 6 and 7 show the simulation results of the SOMA in systems transmitting QAM signals with multi-level bit mapping property. The results are also compared with that of ITS [5] which assigns a constant for the missing LLRs. Note that all of the discarded paths are used in the simulation by the SOMA since the number of them is much smaller than the case with no multi-level bit mapping property. For the 4×4 channel with 16-QAM, the SOMA with $M=16$ achieves the same result as the ITS with $M=32$. For the 64-QAM case, BER for the ITS with $M=128$ is achieved by the SOMA with $M=64$. For the 8×8 channel, the gain by considering discarded paths is more significant. For 16-QAM, the ITS needs to retain at least 128 paths to reach the performance of the SOMA with $M=16$. For the 64-QAM case, the ITS requires to retain 1024 paths to reach the performance of the SOMA with $M=32$. This indicate the M-best fully-extended paths for a deep tree are too similar to each other and they bear only little soft-information. Thus, a large value of M is needed if discarded paths are not used in the LLR calculation.

VI. Results for Turbo-Coded Systems

This section presents the simulation results for a turbo-coded system. The simulation parameters are adapted from [5]. Frames of 9216 information bits are encoded by a rate-1/2 punctured turbo code with recursive and output generators 7 and 5 in octal, respectively. The channel gain coefficients vary from one symbol interval to the next. Fig. 8 shows the results for a 4×4 channel. Four iterations are performed over the MIMO detection loop and eight iterations within the turbo detector. For the 4-QAM case, full-complexity performance is

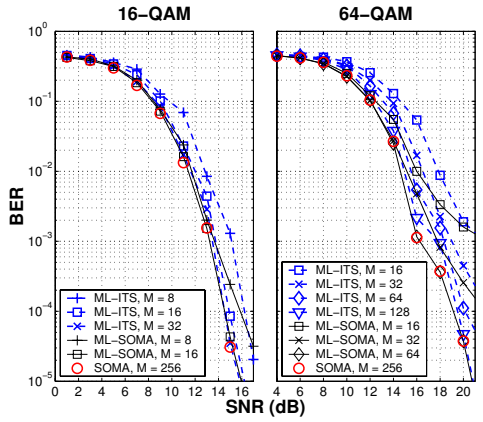

Fig. 6. Performance comparison of the SOMA and the ITS in a 4×4 MIMO system transmitting QAM symbols with multi-level bit mapping property.

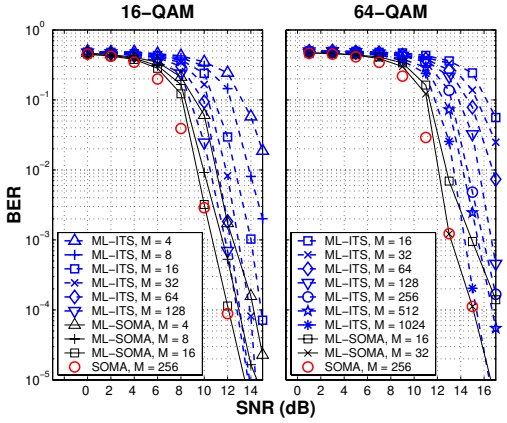

Fig. 7. Performance comparison of the SOMA and the ITS in a 8×8 MIMO system transmitting QAM symbols with multi-level bit mapping property.

achievable by using the SOMA with $M = 16$. For the 16-QAM case, no improvement is observed by increasing M beyond 32. For the 64-QAM case, the loss of using $M = 16$ to $M = 128$ is only 1 dB. Similar results to that in Fig. 8 are found for the 8×8 MIMO system (not shown). It should be noted that there is no significant performance gain in using the SOMA over the ITS in a turbo-coded system. For a powerful FEC outer code such as the turbo, the turbo decoder is capable of correcting errors without the aid of reliable extrinsic values provided by the MIMO detector. Therefore, a small value of M can be use to decode a large tree without incurring a significant performance loss. Consideration of discarded paths is not necessary in this case. For a weaker FEC code such as the convolutional code, the decoder relies on a good estimate of the extrinsic information. Thus, the consideration of the discarded paths is important.

VII. Conclusion

We proposed a low-complexity soft-output tree decoding algorithm for general MIMO systems. The algorithm utilizes discarded paths in a pruned tree to reduce the total number of traversed branches. Simulation results show that discarded

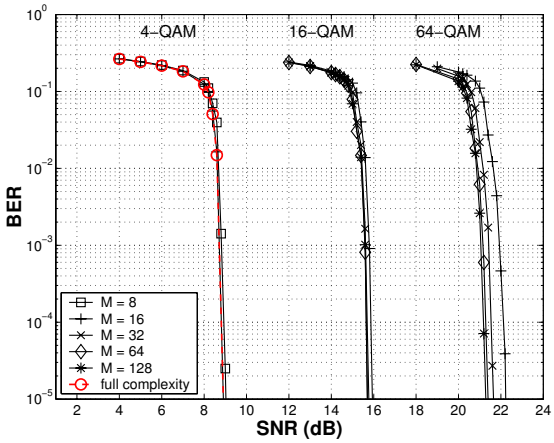

Fig. 8. Performance of the SOMA in a 4×4 MIMO system transmitting 4-QAM, 16-QAM, and 64-QAM symbols. Rate-1/2 punctured Turbo Code is used. The transmission rate is $2K$ bits/channel use for 2^K-QAM.

paths in a heavily pruned tree contain a large amount of information that should not be disregarded. In a convolutional-coded system, the SOMA offers significant improvement in terms of complexity reduction over the ITS. However, there is no advantage in using the SOMA in a turbo-coded system.

Acknowledgement

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

References

[1] M. Damen, H. Gamal, and G. Caire, "On Maximum-Likelihood Detection and the Search for the Closest Lattice Point," in *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2389-2402, Oct. 2003.

[2] J. Anderson, "Limited Search Trellis Decoding of Convolutional Codes," *IEEE Trans. Inform. Theory*, vol. 35, no. 5, pp. 944-955, Sep. 1989.

[3] B.M. Hochwald, and S. ten Brink, "Achieving near-capacity on a multiple antenna channel," in *IEEE Trans. Commun.*, vol. 53, pp. 389-399, Mar. 2003.

[4] S. Bäro, J. Hagenauer, and M. Witzke, "Iterative Detection of MIMO Transmission Using a List-Sequential (LISS) Detector," in *Proc. ICC'03*, vol. 4, pp. 2653-2657, May 2003.

[5] 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.

[6] K. Wong, and P. McLane, "Bi-directional soft-output M-algorithm for iterative decoding," in *Proc. ICC'04*, vol. 2, pp. 792-797, Jun. 2004.

[7] A. Tonello, "Space-Time Bit-Interleaved Coded Modulation with an Iterative Decoding Strategy," in *Proc. VTC-Fall'00*, vol. 1, pp. 473-478, Sep. 2000.

[8] G. Caire, G. Taricco and E. Biglieri, "Bit-Interleaved Coded Modulation," in *IEEE Trans. Inform. Theory*, vol. 44, No. 3, pp. 927-945, May 1998.

[9] J. Hagenauer, "The Turbo Principle: Tutorial Introduction and state of the art," in *Proc. Int. Sym. on Turbo Codes*, pp. 1-11, Sep. 1997.

[10] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, pp. 284-287, Mar. 1974.

[11] 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.