Efficient Soft-Input Soft-Output MIMO Detection Via Improved $M$-algorithm

Jun Won Choi*, Byonghyo Shim**, Jill K. Nelson***, and Andrew C. Singer*

*University of Illinois at Urbana-Champaign
** Korea University
***George Mason University

Abstract—In this paper, we propose a new soft-input soft-output (SISO) multi-input multi-output (MIMO) detection technique, called an improved SISO $M$-algorithm (ISS-MA). We modify the conventional $M$-algorithm to improve the performance-complexity trade-off of the SISO symbol detector. Towards this end, an improved path metric is proposed, which accounts for the information on undecided symbols at a particular path visited. The inclusion of this information is enabled through a bias term which is added to the conventional path metric in order to reflect the contributions of the undecided symbols. We derive the bias term using soft unconstrained linear estimates of undecided symbols. As a result, the ISS-MA that picks up the best $M$ candidates based on this modified path metric exhibits improved performance/complexity trade-off compared to the existing SISO detectors. According to extensive simulations performed over i.i.d. Rayleigh fading channels, the proposed SISO detector yields significantly lower complexity than other symbol detectors while maintaining strong performance especially in high dimensional systems.

I. INTRODUCTION

Owing to the capability to achieve low bit error rate (BER) performance, tree detection techniques have regained attention in solving a maximum likelihood (ML) or a posteriori probability (APP) detection problem arising in digital communication systems [1]. Recently, a variety of tree detection techniques have been proposed for multi-input multi-output (MIMO) detection in the rubric of sphere decoding algorithms (SDA) [2]–[4]. The SDA employs a systematic tree pruning strategy to reduce the computational complexity of candidate search relative to an exhaustive search. Nevertheless, the complexity of the SD algorithm is still exponential in problem size, which makes real-time implementation challenging for systems of large size [4]. Therefore, guaranteeing manageable complexity for such scenario is essential for the success of next generation wireless technology supporting data rates of hundreds of mega bit per second.

For coded symbol transmission, a common practical decoding scheme is the iterative detection and decoding (IDD) method. Motivated by the turbo principle, the IDD receiver exchanges soft information between symbol detector and channel decoder for performance improvement. In fact, through the iterative IDD process, near-capacity performance in the MIMO system has been achieved [6]. The symbol detector exchanges the soft information with the channel decoder by computing the a posteriori probability (APP) of the transmitted bits given the received signal and a priori probabilities. This detector is called an APP detector. Since direct computation of APP involves large computational complexity, tree detection algorithms have been employed for low-complexity APP detection. Modification of the SDA is first considered in [5], where a priori information is used to speed up the tree search. In [6], a list of multiple symbol candidates with high likelihood functions are found by a fixed-radius sphere search and then the APP is estimated by counting the symbol candidates in the list. In [7], a hard sphere decoder is employed to find a maximum a posteriori (MAP) symbol estimate, and a candidate list is generated by bit-flipping of the MAP estimate. An approach for computing the APP of all bits comprising the symbol vector through a single search is proposed in [8], and more sophisticated extension of this idea is presented in [9]. While the complexity of these APP detectors depends on noise and channel realizations, there have been other algorithms that offer fixed complexity. In [10], the $M$-algorithm is used to find a fixed size candidate list and in [11], the stack algorithm combined with soft augmentation of tail bits is employed for list generation. Other fixed complexity SISO detection methods include [12] and [13].

The $M$-algorithm [14], which identifies the best $M$ candidates for each symbol, has computational advantages due to its parallel structure. In spite of these merits, it suffers from poor performance-complexity trade-off due to its greedy nature, i.e., the algorithm check the paths only in forward direction but never goes back for reconsideration. Once a correct path is rejected in the candidate selection, it is never picked up in the subsequent selections, causing a considerable amount of wasteful search efforts. These erroneous decisions tend to occur more often in early stages of the algorithm since the path metric accumulated reflects only a few symbol spans and thus is not likely to reflect the likelihood of paths reliably.

In this paper, we introduce a new SISO detector, called improved SISO $M$-algorithm (ISS-MA), which employs the improved path metric in selecting symbol candidates. While the conventional path metric counts the contributions of the deterministic paths visited, the proposed detector, referred to as the improved SISO $M$-algorithm (ISS-MA), looks ahead on the unvisited paths and accounts for the contributions of them through a bias term. By adding the bias term on the conventional path metric, we can improve sorting process of the $M$-algorithm, making better symbol candidate selection.

It is worth comparing the path metric derived here with
those of previous tree search methods. A Fano metric, proposed for sequential decoding, introduces a bias term proportional to tree depth to account for the contribution of additive noise [15]. Recently, a similar bias term has been proposed to reduce the complexity of the SDA in [16]. In [11], the probabilities of observed signals were used as a bias term to speed up the tree search. While the bias term in these algorithms are equally assessed for paths of the same length, the ISS-MA assigns different path metrics to each path, which allows for the application to breadth-first search.

II. PROBLEM DESCRIPTION

In this section, we formulate an APP detection problem for the IDD systems and briefly discuss the motivation for our approach.

A. APP Detection

The rate $R_c$ recursive systematic convolutional (RSC) encoder is used to convert the sequence of i.i.d. binary information bits $\{b_i\}$ to the coded sequence $\{c_i\}$. The bit sequence $\{c_i\}$ is permuted using a random interleaver and $Q$ interleaved bits are mapped to a symbol $x_k$ for $2^Q$-QAM transmission. We form an $N \times 1$ symbol vector, i.e., $x = [x_1, \cdots, x_N]^T$. We label the interleaved bits associated with $x_k$ as $c_{k,1}, \cdots, c_{k,Q}$. Due to the interleaving, we can assume that the interleaved bits and corresponding symbols are uncorrelated with each other. The vector of received signal can be expressed as

$$\mathbf{y} = \mathbf{Hx} + \mathbf{n}, \quad (1)$$

where $\mathbf{y}$ and $\mathbf{n}$ are the $L \times 1$ received signal and noise vectors, respectively, and $\mathbf{H}$ is a $L \times N$ complex channel matrix.

Given the observation $\mathbf{y}$ and a priori probabilistic knowledge on $x$, the APP detector computes the a posteriori log-likelihood ratio (LLR) of $\tau_{k,i}$, i.e.,

$$L_{\text{post}}(\tau_{k,i}) = \ln \frac{Pr(\tau_{k,i} = +1 | \mathbf{y})}{Pr(\tau_{k,i} = -1 | \mathbf{y})}. \quad (2)$$

Based on Gaussian noise model, i.e., $\mathbf{n} \sim CN(0, \sigma_n^2 \mathbf{I})$, (2) can be expressed as

$$L_{\text{post}}(\tau_{k,i}) = \ln \frac{\sum_{\mathbf{x}_k \in X_{k,i}^+} \exp \left( \lambda(\mathbf{x}) \right)}{\sum_{\mathbf{x}_k \in X_{k,i}^-} \exp \left( \lambda(\mathbf{x}) \right)} \quad (3)$$

where

$$\lambda(\mathbf{x}) = -\frac{1}{\sigma_n^2} \| \mathbf{y} - \mathbf{Hx} \|^2 + \sum_{i = 1}^{N} \sum_{j = 1}^{Q} \ln Pr(\tau_{i,j}),$$

$$Pr(\tau_{i,j}) = \frac{1}{2} \left( 1 + \tau_{i,j} \tanh \left( \frac{L_{\text{pri}}(\tau_{i,j})}{2} \right) \right), \quad (4)$$

where the set $X_{k,i}^{\pm 1}$ is the set of all symbol combinations satisfying $\tau_{k,i} = \pm 1$, and $L_{\text{pri}}(\tau_{k,i})$ is a priori LLR defined as

$$L_{\text{pri}}(\tau_{k,i}) = \ln Pr(\tau_{k,i} = +1) - \ln Pr(\tau_{k,i} = -1).$$

Using max-log approximation $\ln(e^a + e^b) \approx \max \{a, b\}$, the approximated APP is obtained as

$$L_{\text{post}}(\tau_{k,i}) \approx \max_{\mathbf{x}_k \in X_{k,i}^+} \lambda(\mathbf{x}) - \max_{\mathbf{x}_k \in X_{k,i}^-} \lambda(\mathbf{x}). \quad (5)$$

Once $L_{\text{post}}(\tau_{k,i})$ is found, the extrinsic LLR is computed using

$$L_{\text{ext}}(\tau_{k,i}) = L_{\text{post}}(\tau_{k,i}) - L_{\text{pri}}(\tau_{k,i}).$$

Then, the extrinsic LLRs of all bits contained in a processing block are delivered to the channel decoder. Since a tree detection algorithm searches for the symbol vectors of $x$ maximizing $\lambda(\mathbf{x})$, equivalently, minimizing $-\sigma_n^2 \lambda(\mathbf{x})$, the function $d_{\text{APP}}(x) = -\sigma_n^2 \lambda(\mathbf{x})$ is called a cost metric.

B. TREE REPRESENTATION OF SEARCH SPACE

Tree search algorithm is operated on the tree structured search space. The search space spanned by $N$ symbols $x_1, \cdots, x_N$ can be represented by a tree structure as follows. First, depending on the value of $x_N$, $2^Q$ branches are extended from the root. For each path generated, $2^Q$ child branches are extended according to the value of $x_{N-1}$. These branch extensions repeat until all branches corresponding to $x_{N-2}, \cdots, x_1$ are generated. These steps produce a tree of the depth $N$ and each path reaching the bottom level represents a realization of $x$. In the sequel, we will denote a path associated by a set of symbols $x_i, \cdots, x_j, (i < j)$ using a column vector $x_i^j = [x_i, \cdots, x_j]^T$.

For the systematic search of symbol candidates with smallest cost metric, a path metric is assigned to each path. Towards this end, we perform QR decomposition of $\mathbf{H}$ as

$$\mathbf{H} = [\mathbf{Q}_1 \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}, \quad (6)$$

where $\mathbf{R}$ has an upper-triangular structure and $[\mathbf{Q}_1, \mathbf{Q}_2]$ is an $L \times N$ unitary matrix. We can express $d_{\text{APP}}(x)$ as

$$d_{\text{APP}}(x) = \| \mathbf{y} - \mathbf{Rx} \|^2 - \sigma_n^2 \sum_{k=1}^{N} \sum_{i=1}^{Q} \ln Pr(\tau_{k,i}) \quad (7)$$

$$= \sum_{i=1}^{N} d(x_i^N) + C, \quad (8)$$

where $d(x_i^N) = |y_i - \sum_{j=1}^{N} r_{i,j} x_j^N|^2 - \sigma_n^2 \sum_{i=1}^{Q} \ln Pr(\tau_{k,i})$, $y_i = \mathbf{Q}_2^H \mathbf{y}$ and $C = \| \mathbf{Q}_2^H \mathbf{y} \|^2$. The path metric corresponding to $x_i^N$ is chosen as

$$p^{(C)}(x_i^N) = \sum_{i=k}^{N} d(x_i^N). \quad (9)$$

Note that the path metric is the partial sum of the cost metric in (8). In the sequel, we refer to $p^{(C)}(x_i^N)$ as conventional path metric. Note that the terms $d(x_i^N)(1 \leq i \leq k-1)$ depending on unvisited paths in the cost metric are dropped in this path metric.

C. MOTIVATION OF THIS PAPER

Before discussing a new path metric used in this paper, consider the following path metric;

**Definition 2.1**: A genie-aided path metric is defined as

$$p^{(G)}(x_i^N) = p^{(C)}(x_i^N) + \max_{x_i^N, x_i^{N-1}} \left( \sum_{i=1}^{k-1} d(x_i^N) \right), \quad (10)$$

bias term.
The sum $\sum_{i=1}^{k-1} d(x_i^N)$ that has been dropped from the conventional path metric, is minimized over combinations of undecided symbols $x_i^{k-1}$ and added to the conventional path metric as a bias term. In this setup, the following theorem holds:

**Lemma 2.2:** If the genie-aided path metric defined in (10) is employed for the $M$-algorithm, a closest path can be found with minimal number of node visits, i.e., with $M = 1$.

This lemma can be easily proved by the fact that the genie-aided path metric provides the smallest cost metric among all tail paths.

**Theorem 2.3:** Given the transmitted symbol vector $\vec{x}_k^N$, i.e., $x_k^N = \vec{x}_k^N$, the minimizer $x_i^{k-1}$ of the bias term in (10) is expressed as

$$x_i^{k-1} = \arg \max_{x_i^{k-1}} \ln Pr \left( x_i^{k-1} \mid y' \right),$$

This implies that the minimizer $x_i^{k-1}$ corresponds to the MAP estimate of $x_i^{k-1}$. Due to space consideration, we omit the proof of Theorem 2.3.

**Theorem 2.3** states that the bias term in (10) can be obtained by evaluating the sum $\sum_{i=1}^{k-1} d(x_i^N)$ at the MAP estimate of $x_i^{k-1}$. Although it sounds promising, computation of the MAP sequence estimate requires tremendous complexity, particularly for longer sequence. By employing a linear minimum mean square error (MMSE) estimate of $x_i^N$ instead of the MAP estimate, we can improve search efficiency at a small cost.

### III. IMPROVED SISO M-ALGORITHM (ISS-MA)

In this section, we first derive a new path metric used for the ISS-MA and then, study the application of the new path metric to the $M$-algorithm.

#### A. Derivation of New Path Metric

Consider a path $x_k^N$. Note that $x_k^N$ are determined by the paths visited thus far and $x_i^{k-1}$ represents the undecided paths. While the conventional path metric $p(C)(x_k^N)$ considers contributions from the deterministic path $x_k^N$, only, new path metric takes the undecided path $x_i^{k-1}$ into account. Now, we define the new path metric

**Definition 3.1:** The path metric of ISS-MA, denoted as $p^{(N)}(x_k^N)$ is defined as

$$p^{(N)}(x_k^N) \triangleq p^{(C)}(x_k^N) + \sum_{i=1}^{k-1} d(x_i^N) \bigg|_{x_i^{k-1} = x_i^{k-1}} b(x_k^N)$$

where $\vec{x}_i^{k-1}$ is the linear MMSE estimate of $x_i^{k-1}$ derived assuming that $x_k^N = \vec{x}_k^N$.

A new bias term $b(x_k^N)$ induced from the MMSE estimation is added to the conventional path metric. The transformed vector $y' = Q_1^H y$ can be expressed as $y' = Rx + n'$ and $n' = Q_1 n$.

We can partition the vectors $y'$, $x$, and $n'$ into two $(k-1) \times 1$ and $(N-k+1) \times 1$ vectors such that

$$y' = \begin{bmatrix} y_1^{k-1} \\ y_k^N \end{bmatrix} = \begin{bmatrix} R_{11,k} & R_{12,k} \\ 0 & R_{22,k} \end{bmatrix} \begin{bmatrix} x_1^{k-1} \\ x_k^N \end{bmatrix} + \begin{bmatrix} n_1^{k-1} \\ n_k^N \end{bmatrix},$$

where $R$ is partitioned into four sub-matrices accordingly. Using (13), we can express the conventional path metric as

$$p^{(C)}(x_k^N) = \| y_1^{k-1} - R_{11,k} x_1^{k-1} + R_{12,k} x_k^N \|^2 - \sigma_n^2 \sum_{i=1}^N \sum_{q=1}^Q \ln Pr(\tau_{k,i}).$$

The path metric of ISS-MA, denoted as $p^{(M)}(x_k^N)$ includes all possible symbol values. From (14) to (16) where $x_i^{k-1} = E[x_i^{k-1}]$ and $\mathbf{F}_k = \text{Cov}(x_i^{k-1}, y_1^{k-1}, y_k^N)$, we can derive $x_i^{k-1}$ and $\mathbf{F}_k$ [17].

**Theorem 2.3:**

$$x_i^{k-1} = \begin{bmatrix} \sum_{\theta \in \Theta} \prod_{j=1}^Q \left( 1 + \tau_{i,j} \tanh \left( \frac{L_{\text{pri}}(\tau_{i,j})}{2} \right) \right) \\ \sum_{\theta \in \Theta} \prod_{j=1}^Q \left( 1 + \tau_{i-1,j} \tanh \left( \frac{L_{\text{pri}}(\tau_{i-1,j})}{2} \right) \right) \end{bmatrix}$$

$$\mathbf{F}_k = \mathbf{A}_k(\mathbf{R}_{11,k})^H \left[ (\mathbf{R}_{11,k}) \mathbf{A}_k(\mathbf{R}_{11,k})^H + \sigma_n^2 \mathbf{I} \right]^{-1},$$

where $\mathbf{A}_k = \text{diag}(\lambda_1, \ldots, \lambda_{k-1})$ and

$$\lambda_i = \sum_{\theta \in \Theta} |\theta - \mathbf{x}_i^N|^2 \prod_{j=1}^Q \left( 1 + \tau_{i,j} \tanh \left( \frac{L_{\text{pri}}(\mathbf{E}_{i,j})}{2} \right) \right).$$

The set $\Theta$ includes all possible symbol values. From (14) to (16), the bias term $b(x_k^N)$ is written

$$b(x_k^N) = \| \mathbf{Z}_k (y_1^{k-1} - R_{11,k} x_1^{k-1} - R_{12,k} x_k^N) \|^2,$$

where

$$\mathbf{Z}_k = \mathbf{I} - R_{11,k} \mathbf{F}_k$$

$$= \sigma_n^2 [R_{11,k} \mathbf{A}_k(\mathbf{R}_{11,k})^H + \sigma_n^2 \mathbf{I}]^{-1}.$$ Further, denoting $\mathbf{q}_k = \mathbf{Z}_k (y_1^{k-1} - R_{11,k} x_1^{k-1})$ and $\mathbf{V}_k = \mathbf{Z}_k \mathbf{R}_{12,k}$, we simply express the new path metric as

$$p^{(N)}(x_k^N) = p^{(C)}(x_k^N) + \| \mathbf{q}_k - \mathbf{V}_k x_k^N \|^2.$$
TABLE I
SUMMARY OF ISS-MA

Output: \( \{ L_{\text{post}}(\vec{\tau}_{k,i}) \}_{k=1:N, i=1:Q} \)
Input: \( y, H, \{ L_{\text{init}}(\vec{\tau}_{k,i}) \}_{k=1:N, i=1:Q} \) and \( J \)

STEP 1: (Preprocessing) Perform V-BLAST symbol ordering and QR decomposition of \( H \). Obtain \( Z_1 \) to \( Z_N \) for all \( k \) levels.
STEP 2: (Initialization) Initialize \( i = N + 1 \) and start the tree search from the root node.
STEP 3: (Loop) Extend \( 2^Q \) branches for each of \( M \) paths that have survived at the \((i+1)\)th level. This generates \( 2^Q M \) paths at the \( i \)th level.
STEP 4: If \( i > 1 \), choose the best \( M \) paths with the smallest \( p^{(N)}(\vec{x}^N) \) and go to STEP 3 with \( i = i - 1 \). Otherwise, store all \( 2^Q M \) survival candidates into the list \( L \) and go to STEP 5.
STEP 5: (List extension & APP calculation) For each value of \( k \) and \( i \), compute \( \{ L_{\text{post}}(\vec{\tau}_{k,i}) \} \) based on \( L \). If the value of \( \vec{\tau}_{k,i} \) for all elements of \( L \) is either +1 or -1, the value of \( \vec{\tau}_{k,1} \) of the best \( J \) candidates is flipped and these counter-hypothesis candidates are added to \( L \) to generate the extended \( L^+_{k} \). The APP is calculated over the extended list.

It is worth mentioning that when the candidate list \( L \) contains symbol vectors whose particular bit entry is only either +1 or -1, generated LLR value would be infinite, causing a bias in APP calculation. If this case occurs, we flip the bits of the best \( J \) candidates in \( L \) and add these candidates under counter hypothesis to \( L \). Hence, as we increase the value of \( J \), the bias due to insufficient size of the candidate list would be reduced. The ISS-MA is summarized in Table II.

IV. SIMULATIONS

In this section, we evaluate the performance of the ISS-MA based on computer simulations.

A. Simulation Setup

A total of \( 10^6 \) information bits are generated. An i.i.d. Rayleigh fading channels are assumed for each element of \( H \). For channel coding, the rate of 1/2 recursive systematic convolution (RSC) code with generator matrix \((5, 7)\) is used. A random interleaver is used, and the max-log BCJR decoder is employed for channel decoding. The size of the frame for turbo iteration is set to 800 symbols, equivalently, a total of \( 800 \times QM \) bits. The complexity of the detectors is measured by counting the average number of complex multiplications per channel use. The complexity for QR decomposition and detection ordering is not included for our calculation and hence, the complexity is mostly due to tree search and APP calculation. Note that \( J \) is set to 4 for the whole simulations since larger \( J \) did not improve the performance for most of cases. Along with the ISS-MA, we consider the following algorithms;

1) Iterative tree search (ITS) \((M)\) [10]; The SISO-\(M \) algorithm based on the conventional path metric.
2) LISS algorithm \((|S|, |S_z|)\) [11] - sequential list stack algorithm. It is characterized by the size of stack \(|S|\) and that of auxiliary stack \(|S_z|\) for bias term.
3) Hard sphere decoding (HSD) [7] - Hard sphere search is performed to find a MAP estimate. Then, the candidate list is generated by flipping each bit of the MAP estimate. Only one bit flip is considered.

B. Simulation Results

Fig. 1 shows the BER performance of the ISS-MA after several iterations. The performance of the ITS is also included for comparison. The \( 8 \times 8 \) MIMO system with 16-QAM transmission is considered. We set \( M = 4 \) for both algorithms. As shown in the figure, the ISS-MA outperforms the ITS for each iteration. In addition, the performance of ISS-MA converges faster than that of the ITS. Note that at the BER of 0.01, the performance gap is 1.5 dB after first iteration. Even after performance converges, 1 dB performance gap is achieved.

In Fig. 2, the complexity and performance of the different SISO detectors are compared. The \( 6 \times 6 \) 16-QAM system is considered. The BER performance is measured after 6th iterations. While the average complexity of the HSD increases as SNR decreases, the remaining detectors retain fixed complexity. The performance gap among all SISO detectors considered is less than 0.5 dB at the BER of \( 10^{-3} \) while the gap becomes larger for higher BER range. The complexity of the ISS-MA is much less than that of LISS and HSD. At the 9 dB of SNR, the ISS-MA achieves 60% complexity gain compared to the LISS and HSD. Though the ISS-MA has higher complexity than the ITS due to computation of the bias metric, the ISS-MA offers better performance and maintains large complexity gain over other two candidates.

V. CONCLUSIONS

In this paper, an efficient SISO detection algorithm was proposed for MIMO systems. In order to improve the sorting process in the \( M \)-algorithm, the new path metric was derived,
which accounts for the impact of undecided symbols through the linear MMSE estimate of undecided symbols. The final candidates found by the proposed ISS-MA are used to obtain the APP for the IDD systems. The performance improvement achieved by the ISS-MA was demonstrated through computer simulations. We also showed that the ISS-MA has better performance-complexity trade-off compared to the existing SISO tree detectors. The improved path metric can be applied to different tree detection algorithms so that the extensive study on application of the improved path metric would be an interesting issue for future research.

REFERENCES


