Efficient Soft-Input Soft-Output Tree Detection Via an Improved Path Metric

Jun Won Choi, Byonghyo Shim, *Senior Member, IEEE,* and Andrew C. Singer, *Fellow, IEEE*

Abstract

Tree detection techniques are often used to reduce the complexity of *a posteriori probability* (APP) detection in high dimensional multi-antenna wireless communication systems. In this paper, we introduce an efficient soft-input soft-output tree detection algorithm that employs a new type of look-ahead path metric in the computation of its branch pruning (or sorting). While conventional path metrics depend only on symbols on a visited path, the new path metric accounts for unvisited parts of the tree in advance through an unconstrained linear estimator and adds a bias term that reflects the contribution of as-yet undecided symbols. By applying the linear estimate-based look-ahead path metric to an M-algorithm that selects the best M paths for each level of the tree we develop a new soft-input soft-output tree detector, called an *improved soft-input soft-output M-algorithm* (ISS-MA). Based on an analysis of the probability of correct path loss, we show that the improved path metric offers substantial performance gain over the conventional path metric. We also demonstrate through simulations that the ISS-MA provides a better performance-complexity trade-off than existing soft-input soft-output detection algorithms.

I. Introduction

The relationship between the transmitted symbol and the received signal vector in many communication systems can be expressed in the form

$$\mathbf{y}_o = \mathbf{H}\mathbf{x} + \mathbf{n}_o, \tag{1}$$

where $\mathbf{x}$ is the $N \times 1$ transmitted vector whose entries are chosen from a finite symbol alphabet, $\mathbf{y}_o$ and $\mathbf{n}_o$ are the $L \times 1$ received signal and noise vectors, respectively, and $\mathbf{H}$ is $L \times N$ channel matrix. As

Dedicated to the memory of Ralf Koetter, with deep respect and admiration.

a practical decoding scheme when a code constraint is imposed, the *iterative detection and decoding* (IDD) has been applied to various digital communication systems including channel equalization [1], multi-input multi-output (MIMO) detection [2]–[4], and multi-user detection [5]. Motivated by the *turbo principle* [6], an IDD receiver exchanges soft information between a symbol detector and a channel decoder to achieve performance close to the channel capacity. The symbol detector computes *a posteriori* probabilities (APP) on the bits comprising $\mathbf{x}$, using *a priori* probabilities provided by the channel decoder and the observation $\mathbf{y}_o$. Then, the detector exchanges this soft information (so called *extrinsic* information) with a soft-input soft-output decoder such as max-log-MAP decoder [7]. In the sequel, we refer to this detector as an *APP detector*.

Direct computation of the APP involves marginalization over all configurations of the vector $\mathbf{x}$, leading to exponential complexity in the system size (e.g., number of antenna elements in MIMO systems). As a means of approximately performing the APP detection at reduced complexity, *tree detection techniques* have received much attention recently [3], [4], [8]–[16]. (Refer [17] for the overview of tree detection techniques.) The essence of these approaches is to produce a set of promising symbol candidates via a tree search for estimating the APP over this reduced set. Thus far, a variety of tree detection algorithms have been proposed. In [3], the *sphere decoding algorithm* (SDA) [18], [19] with a fixed radius was used to find symbol candidates. In [4], *a priori* information obtained from the channel decoder was exploited to improve the search efficiency of the SDA. In [8], a hard sphere decoder was employed to find a single *maximum a posteriori* probability (MAP) symbol estimate maximizing $P(\mathbf{x}|\mathbf{y}_o)$ and a candidate list was generated by flipping bits in the MAP estimate. In [9], the APPs of all bits in $\mathbf{x}$ are obtained simultaneously by modifying a bound tightening rule of a single sphere search. Additionally, a more sophisticated extension of this idea was introduced in [10]. The computational complexity of these tree detection algorithms varies depending on the channel and noise realizations, and in the worst-case the search complexity is the same as that of exhaustive search. In order to limit the worst case complexity of the tree detection approach, *fixed-complexity tree search* [20] has been proposed. For example, an M-algorithm was extended to soft-input soft-output detection in [11] and a smart candidate adding algorithm for improving efficiency of the M-algorithm was proposed in [12]. Besides, the stack algorithm was exploited for the list generation in combination with soft augmentation of tail bits of stack elements [13]. Other fixed-complexity soft-input soft-output detection algorithms can be found in [14]–[16].

The M-algorithm [11], [21], also known as *K-best algorithm* in the literature on MIMO detection [22], [23], selects only M best candidates for each layer of detection tree. The M-algorithm is

a practical candidate for soft-input soft-output detection due to its inherent nature to facilitate parallel and pipelined processing [23]. In spite of this benefit, the M-algorithm suffers from a poor performance-complexity trade-off due to the greedy nature of the algorithm. To be specific, the algorithm checks the validity of paths in the forward direction and never traverses back for reconsideration. Once the correct path is rejected, it will never be selected again in subsequent selections, resulting in wasteful search effort. Moreover, these erroneous decisions often occur in early candidate selection stages where the accumulated path metric considers only a few symbol spans. One way to alleviate such error propagation is symbol detection ordering [12], [24]. By processing each layer in an appropriate order, the chances of errors propagating to the next stage can be reduced. Nevertheless, error propagation severely limits the performance of the M-algorithm especially when a system size is large.

In this paper, we pursue an improvement of performance-complexity trade-off of soft-input soft-output M-algorithms. Towards this end, we propose a new path metric capturing the contribution of the entire symbol path. While the conventional path metric accounts for the contributions of symbols along the visited path only, the new path metric looks ahead to the unvisited paths and estimates their contributions through a soft unconstrained linear symbol estimate. In fact, a *bias term* reflecting the information from as-yet undecided symbols is incorporated into the conventional path metric for this purpose. In order to distinguish this improved path metric from the conventional path metric and other look-ahead metrics, we henceforth refer to it as a *linear estimate-based look-ahead (LE-LA) path metric*. We apply the LE-LA path metric to soft-input soft-output M-algorithm, introducing *improved soft-in soft-out M-algorithm* (ISS-MA). By sorting paths based on the LE-LA path metric, the ISS-MA lessens the chance of rejecting the correct path from the candidate list and eventually improves the detection performance especially for large size systems. Indeed, from an analysis of the probability of correct path loss (CPL), we show that the LE-LA path metric benefits the candidate selection process of the M-algorithm.

The idea of look-ahead path metric has been explored in artificial intelligence search problems [25] and can also be found in soft decoding of linear block codes [26], [27]. In [28], computationally efficient methods to obtain the bias term were investigated using semi-definite programming and H^∞ estimation techniques. In these approaches, the bias term is designed to guarantee the optimality of the tree search. Our approach is distinct from these schemes in the sense that linear estimation is exploited to obtain a useful bias term. The key advantage of using a linear estimator is that *a prior* information can be easily incorporated into the bias term so that the look-ahead operation benefits

from the decoder output in each iteration. It is also worth emphasizing the difference between the proposed path metric and Fano matric [29]. The Fano metric exploits the *a posteriori* probability of each path as its path metric. For a binary symmetric channel, the Fano metric introduces a bias term proportional to the path length to penalize paths of short length. The extension of the Fano metric to channels with memory or MIMO channels is not straightforward, since it involves marginalization over the distribution of the undecided symbols. Modification of the Fano metric is considered for equalization of intersymbol interference (ISI) channels in [30] and for multi-input multi-output detection in [17], [31]–[33]. In [13], the probability density of an observed signal estimated from separate tree search is used as a bias term. While these approaches assign an equal bias term for paths of the same length, the ISS-MA provides a distinct bias term for each path in the tree, allowing for the application of a breadth-first search including the M-algorithm. Henceforth, our path metric can be readily combined with any type of tree-based soft-input soft-output detector.

The rest of this paper is organized as follows. In Section II, we briefly review the IDD system and the tree detection algorithm. In Section III, we present the LE-LA path metric along with its efficient computation. We also describe the application of the LE-LA path metric to the soft-input soft-output M-algorithm. In Section IV, we present the performance analysis of the ISS-MA. In Section V, we provide simulation results and conclude in Section VI.

We briefly summarize the notation used in this paper. Uppercase and lowercase letters written in boldface denote matrices and vectors, respectively. The superscripts $(\cdot)^T$ and $(\cdot)^H$ denote transpose and conjugate transpose (hermitian operator), respectively. $\|\cdot\|^2$ denotes an L_2-norm square of a vector and $\mathrm{diag}\,(\cdot)$ is a diagonal matrix that has elements on the main diagonal. $\mathbf{0}_{M\times N}$ and $\mathbf{1}_{M\times N}$ are $M \times N$ matrix whose entries are all ones or zeros, respectively. The subscript is omitted if there is no risk of confusion. $\mathcal{CN}(m, \sigma^2)$ denotes a circular symmetric complex Gaussian density with mean m and variance σ^2. $E_x[\cdot]$ denotes expectation over the random variable x. $\mathrm{Cov}\,(\mathbf{x}, \mathbf{y})$ denotes $E\left[\mathbf{x}\mathbf{y}^H\right] - E[\mathbf{x}]E[\mathbf{y}^H]$. For a hermitian matrix $\mathbf{A}$, $\mathbf{A} \succeq 0$ (or $\mathbf{A} \succ 0$) means that $\mathbf{A}$ is semi-positive definite (or positive definite). $Pr(A)$ means probability of the event A. $f_{x_1,x_2,\cdots,x_n}(a_1, a_2, \cdots, a_n)$ denotes a joint probability density function (PDF) for the random variables $x_1, x_2, \cdots, x_n$.

II. Problem Description

In this section, we briefly review the IDD framework and then introduce the tree detection algorithms.

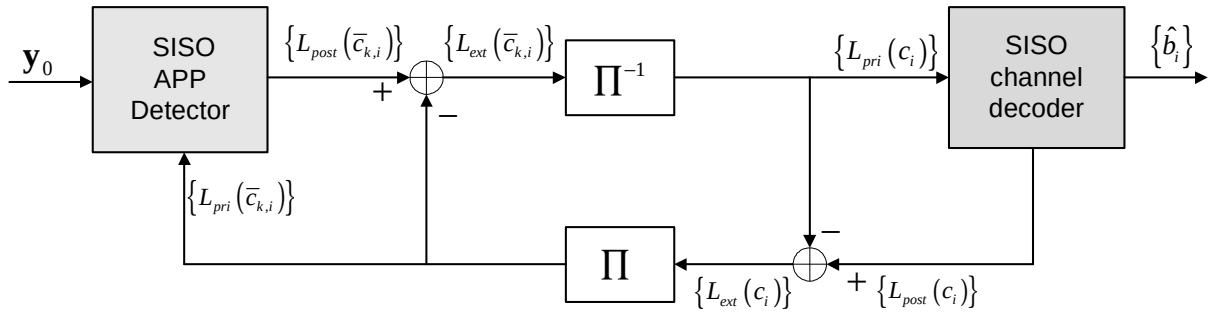


Fig. 1. Block diagram of the IDD system.

A. Iterative Detection and Decoding (IDD)

In a transmitter, a rate R_c channel encoder is used to convert a sequence of independent identically distributed (i.i.d.) binary information bits $\{b_i\}$ to an encoded sequence $\{c_i\}$. The bit sequence $\{c_i\}$ is permuted using a random interleaver $\prod$ and then mapped into a symbol vector using a 2^Q-ary quadrature amplitude modulation (QAM) symbol alphabet. We label the interleaved bits associated with the kth symbol x_k by $\bar{c}_{k,1}, \cdots, \bar{c}_{k,Q}$. Due to the interleaver, we can assume that these interleaved bits are mutually uncorrelated.

In the system model (1), $\mathbf{y}_o$ and $\mathbf{n}_o$ are the $L \times 1$ received signal and noise vectors, respectively. Each entry of the $N \times 1$ symbol vector $\mathbf{x}$ is drawn from a finite alphabet

$$\mathcal{F} = \Big\{ x_r + jx_i \big| \;\; x_r, x_i \in \Big\{ \frac{-2^{Q/2}+1}{P}, \frac{-2^{Q/2}+3}{P}, \cdots, \frac{2^{Q/2}-3}{P}, \frac{2^{Q/2}-1}{P} \Big\} \Big\}, \tag{2}$$

where P is chosen to satisfy the normalization condition $E\left[|x_k|^2\right] = 1$. For example, $P = \sqrt{10}$ for 16-QAM and $P = \sqrt{42}$ for 64-QAM modulation, respectively.

Fig. 1 depicts the basic structure of an IDD system. The receiver consists of two main blocks; the APP detector and the channel decoder. The APP detector generates the *a posteriori* log-likelihood ratio (LLR) of $\bar{c}_{k,i}$ using the observation $\mathbf{y}_o$ and *a priori* information delivered from the channel decoder. The *a posteriori* LLR is defined as

$$L_{\text{post}}\left(\bar{c}_{k,i}\right) = \ln \frac{Pr(\bar{c}_{k,i} = +1 | \mathbf{y}_o)}{Pr(\bar{c}_{k,i} = -1 | \mathbf{y}_o)}, \tag{3}$$

where we take $\bar{c}_{k,i} \in \{-1, 1\}$ rather than $\{0, 1\}$ by convention. With the standard noise model $\mathbf{n}_o \sim \mathcal{CN}\left(0, \sigma_n^2 \mathbf{I}\right)$, (3) can be rewritten [3]

$$L_{\text{post}}\left(\bar{c}_{k,i}\right) = \ln \frac{\sum_{\mathbf{x} \in X_{k,i}^{+1}} \exp\left(\psi\left(\mathbf{x}\right)\right)}{\sum_{\mathbf{x} \in X_{k,i}^{-1}} \exp\left(\psi\left(\mathbf{x}\right)\right)}, \tag{4}$$

where

$$\psi(\mathbf{x}) = -\frac{1}{\sigma_n^2}\left\|\mathbf{y}_o - \mathbf{H}\mathbf{x}\right\|^2 + \sum_{i=1}^{N}\sum_{j=1}^{Q} \ln Pr\left(\overline{c}_{i,j}\right),$$

$$Pr\left(\overline{c}_{i,j}\right) = \frac{1}{2}\left(1 + \overline{c}_{i,j}\tanh\left(\frac{L_{\rm pri}(\overline{c}_{i,j})}{2}\right)\right). \quad (5)$$

The set $X_{k,i}^{+1}$ is the set of all configurations of the vector $\mathbf{x}$ satisfying $\overline{c}_{k,i} = +1$ ($X_{k,i}^{-1}$ is defined similarly), and $L_{\rm pri}(\overline{c}_{k,i})$ is the *a priori* LLR defined as $L_{\rm pri}(\overline{c}_{k,i}) = \ln Pr(\overline{c}_{k,i} = +1) - \ln Pr(\overline{c}_{k,i} = -1)$. Once $L_{\rm post}(\overline{c}_{k,i})$ is computed, the extrinsic LLR is obtained from $L_{\rm ext}(\overline{c}_{k,i}) = L_{\rm post}(\overline{c}_{k,i}) - L_{\rm pri}(\overline{c}_{k,i})$. These extrinsic LLRs are de-interleaved and then delivered to the channel decoder. The channel decoder computes the extrinsic LLR for the coded bits $\{c_i\}$ and feeds them back to the APP detector. These operations are repeated until a suitably chosen convergence criterion is achieved [3].

B. Soft-input Soft-output Tree Detection

The direct computation of the *a posteriori* LLR in (4) involves marginalization over 2^{NQ} symbol candidates, which easily becomes infeasible for large systems employing high order modulations. A tree detection algorithm addresses this problem by searching a small set of promising symbol candidates over which *a posteriori* LLRs are estimated. Specifically, a small number of symbol vectors with large $\psi(\mathbf{x})$, equivalently, small $-\sigma_n^2\psi(\mathbf{x})$, are sought. In the sequel, we refer to $d_{\rm APP}(\mathbf{x}) = -\sigma_n^2\psi(\mathbf{x})$ as a *cost metric* for tree detection, where $-\sigma_n^2$ is a scaling factor. The goal of the tree detection algorithm is to find symbol vectors of small cost metric, and the best (minimum) among them corresponds to the *maximum a posteriori* (MAP) solution (denoted by $\mathbf{x}_{\rm MAP}$).

The tree detection algorithm relies on the tree representation of the search space spanned by $\mathbf{x} = (x_1, \cdots, x_N) \in \mathcal{F}^N$. Tree construction is performed from the root node as follows. First, representing the symbol realization for x_N, we extend 2^Q branches from the root (recall that we assume 2^Q-ary QAM modulation). For each such branch, 2^Q child branches are extended for the possible realization of the next symbol x_{N-1}. These branch extensions are repeated until all branches corresponding to $x_N, \cdots, x_1$ are generated. This yields a tree of the depth N, where each "complete" path from the root to a leaf corresponds to a realization of $\mathbf{x}$. In order to find the complete paths of small cost metric, the tree detection algorithm searches the tree using a systematic node visiting rule. For notational simplicity, we henceforth denote a path associated with a set of symbols $x_i, \cdots, x_j, (i < j)$ by a column vector $\mathbf{x}_i^j = [x_i, \cdots, x_j]^T$. Also, we call a level of tree associated with the symbol x_i "the ith level" (e.g., the bottom level associated with x_1 is the first level). For details on tree construction, see [19] and [34].

For a systematic search of symbol candidates, a *path metric* is assigned to each path $\mathbf{x}_i^N$. Towards this end, we perform a QR decomposition of $\mathbf{H}$ as

$$\mathbf{H} = \mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix} = [\mathbf{Q}_1 \;\; \mathbf{Q}_2]\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}, \tag{6}$$

where $\mathbf{R}$ has an $N \times N$ upper-triangular matrix whose diagonals are non-negative and $\mathbf{Q}$ is an $L \times N$ matrix satisfying $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$. Using the invariance of the norm to unitary transformations, we can define the cost metric $d_{\mathrm{APP}}(\mathbf{x})$ as

$$d_{\mathrm{APP}}(\mathbf{x}) = -\sigma_n^2\psi(\mathbf{x}) = \|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 - \sigma_n^2 \sum_{i=1}^{N}\sum_{j=1}^{Q} \ln Pr\left(\overline{c}_{i,j}\right) + C \tag{7}$$

$$= \sum_{i=1}^{N} b(\mathbf{x}_i^N) + C, \tag{8}$$

where $b(\mathbf{x}_i^N) = \left|y_i' - \sum_{j=i}^{N} r_{i,j}x_j\right|^2 - \sigma_n^2 \sum_{i=1}^{Q} \ln Pr\left(\overline{c}_{k,i}\right)$, and $\mathbf{y} = [y_1, \cdots, y_N]^T = \mathbf{Q}_1^H\mathbf{y}_o$ and $C = \left\|\mathbf{Q}_2^H\mathbf{y}_o\right\|^2$. The path metric associated with the path $\mathbf{x}_k^N$ can be defined as a partial sum in the cost metric [19], [34]

$$\gamma^{(c)}\left(\mathbf{x}_k^N\right) = \sum_{i=k}^{N} b(\mathbf{x}_i^N). \tag{9}$$

Whenever a new node is visited, the term $b(\mathbf{x}_i^N)$, referred to as a branch metric, is added to the path metric of the parent node. Since the branch metric is non-negative for all i, the path metric $\gamma^{(c)}\left(\mathbf{x}_k^N\right)$ becomes a lower bound of the cost metric $d_{\mathrm{APP}}(\mathbf{x})$. Using $\gamma^{(c)}\left(\mathbf{x}_k^N\right)$, the tree detection algorithm compares the reliability of distinctive paths and chooses the surviving paths. Since the path metric is determined by the visited path, we henceforth denote $\gamma^{(c)}\left(\mathbf{x}_k^N\right)$ as a *causal path metric*.

According to predefined node visiting rule [17], the tree detection algorithm finds the *complete paths* associated with smallest cost metric. Denoting the set of the corresponding symbol candidates as $\mathcal{L}$, an approximate APP can be expressed as

$$L_{\mathrm{post}}\left(\overline{c}_{k,i}\right) \approx \ln \frac{\sum_{\mathbf{x}\in\mathcal{L}\cap X_{k,i}^{+1}} \exp\left(\psi\left(\mathbf{x}\right)\right)}{\sum_{\mathbf{x}\in\mathcal{L}\cap X_{k,i}^{-1}} \exp\left(\psi\left(\mathbf{x}\right)\right)}. \tag{10}$$

Further simplification can be achieved using max-log approximation [7]

$$L_{\mathrm{post}}(\overline{c}_{k,i}) \approx \left(\max_{\mathbf{x}\in\mathcal{L}\cap X_{k,i}^{+1}} \psi\left(\mathbf{x}\right) - \max_{\mathbf{x}\in\mathcal{L}\cap X_{k,i}^{-1}} \psi\left(\mathbf{x}\right)\right). \tag{11}$$

Since $\mathcal{L}$ does not span whole symbol space, either $\mathcal{L}\cap X_{k,i}^{+1}$ or $\mathcal{L}\cap X_{k,i}^{-1}$ might be empty for some k values. If this case happens, the magnitude of $L_{\mathrm{post}}(\overline{c}_{k,i})$ is set to infinity, causing a bias in LLR values. One way to cope with this event is to clip the magnitude of $L_{\mathrm{post}}(\overline{c}_{k,i})$ to a constant value (e.g., ± 8) [3].

III. Improved Soft-input Soft-output M-algorithm (ISS-MA)

In this section, we present the ISS-MA which improves the candidate selection process of the soft-input soft-output tree detection algorithms. We first describe a genie-aided path metric that motivates our work and then introduce the new path metric that accounts for the information on the unvisited paths. We also discuss an efficient way to compute the new path metric.

A. Motivation

We begin our discussion with the following path metric:

Definition 3.1: A *genie-aided path metric* $\gamma^{(g)}\left(\mathbf{x}_k^N\right)$ is defined as

$$\gamma^{(g)}\left(\mathbf{x}_k^N\right) = \gamma^{(c)}\left(\mathbf{x}_k^N\right) + \underbrace{\min_{\mathbf{x}_1^{k-1}}\left(\sum_{i=1}^{k-1} b\left(\mathbf{x}_i^N\right)\right)}_{\text{bias term}}. \tag{12}$$

The genie-aided path metric is obtained by minimizing the sum of $b(\mathbf{x}_i^N)(1 \le i \le k-1)$ over all combinations of undecided symbols $\mathbf{x}_1^{k-1}$. This minimal term, which can be considered as a bias term, is added into the causal path metric. The genie-aided path metric can be used in the M-algorithm so that the best M candidates with the smallest genie-aided path metric are selected in each tree level. It is easy to show that the M-algorithm employing the genie-aided path metric finds the closest (best) path with probability one (even for $M = 1$). This can be readily shown using the fact that the genie-aided path metric provides the smallest cost metric among all tail paths. Note that the similar path metric has also appeared in [28].

Theorem 3.2: Given the actual transmitted symbol vector $\widetilde{\mathbf{x}}_k^N$ (i.e., $\mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N$), the bias term of the genie-aided path metric is

$$\min_{\mathbf{x}_1^{k-1}}\left(\sum_{i=1}^{k-1} b\left(\mathbf{x}_i^N\right)\right) = \sum_{i=1}^{k-1} b\left(\mathbf{x}_i^N\right)\Big|_{\mathbf{x}_1^{k-1}=\check{\mathbf{x}}_1^{k-1}}, \tag{13}$$

where the minimizer $\check{\mathbf{x}}_1^{k-1}$ is the MAP estimate of $\mathbf{x}_1^{k-1}$, i.e.,

$$\check{\mathbf{x}}_1^{k-1} = \arg\max_{\mathbf{x}_1^{k-1}} \ln Pr\left(\mathbf{x}_1^{k-1}\middle|\mathbf{y}, \mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N\right). \tag{14}$$

Proof: See Appendix A. ■

Theorem 3.2 implies that the bias term of the genie-aided path metric is obtained by computing $\sum_{i=1}^{k-1} b\left(\mathbf{x}_i^N\right)$ using the MAP estimate of $\mathbf{x}_1^{k-1}$. This MAP estimate is derived under the condition that the path associated with the actual transmitted symbols, $\widetilde{\mathbf{x}}_k^N$ is given. Though the genie-aided path metric offers a substantial performance gain, it is impractical to incorporate it into tree search due to the high complexity associated with the MAP estimation.

B. Derivation of Linear Estimate-Based Look-Ahead (LE-LA) Path Metric

In order to alleviate the complexity associated with MAP detection of $\mathbf{x}_1^{k-1}$ in the genie-aided path metric, we relax the finite alphabet constraint of $\mathbf{x}_1^{k-1}$ and then replace the MAP estimate by the linear MMSE estimate $\hat{\mathbf{x}}_1^{k-1}$. Note that when $\hat{\mathbf{x}}_1^{k-1}$ is assumed to be Gaussian, the MAP estimate is identical to the linear MMSE solution [35]. For a particular path visited $\mathbf{x}_k^N$, we first define the LE-LA path metric

Definition 3.3: The *linear estimate-based look-ahead path metric*, denoted by $\gamma^{(l)}\left(\mathbf{x}_k^N\right)$, is defined as

$$\gamma^{(l)}\left(\mathbf{x}_k^N\right) \triangleq \gamma^{(c)}\left(\mathbf{x}_k^N\right) + \underbrace{\sum_{i=1}^{k-1} b\left(\mathbf{x}_i^N\right)\Big|_{\mathbf{x}_1^{k-1}=\hat{\mathbf{x}}_1^{k-1}}}_{\text{bias term}}, \tag{15}$$

where $\hat{\mathbf{x}}_1^{k-1}$ is the linear MMSE estimate of $\mathbf{x}_1^{k-1}$.
Note that $\hat{\mathbf{x}}_1^{k-1}$ is obtained under the condition that $\mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N$. In the sequel, we denote this bias term as $\gamma^{(b)}\left(\mathbf{x}_k^N\right)$.

To derive the linear MMSE estimate $\hat{\mathbf{x}}_1^{k-1}$, we partition the vectors $\mathbf{y}$ and $\mathbf{n}(\triangleq \mathbf{Q}_1^H\mathbf{n}_o)$ to $(k-1)\times 1$ and $(N-k+1)\times 1$ vectors, i.e.,

$$\mathbf{y} = \begin{bmatrix}\mathbf{y}_1^{k-1}\\ \mathbf{y}_k^N\end{bmatrix} = \begin{bmatrix}\mathbf{R}_{11,k} & \mathbf{R}_{12,k}\\ \mathbf{0} & \mathbf{R}_{22,k}\end{bmatrix}\begin{bmatrix}\mathbf{x}_1^{k-1}\\ \mathbf{x}_k^N\end{bmatrix} + \begin{bmatrix}\mathbf{n}_1^{k-1}\\ \mathbf{n}_k^N\end{bmatrix}, \tag{16}$$

where $\mathbf{R}_{11,k}$, $\mathbf{R}_{12,k}$, and $\mathbf{R}_{22,k}$ are the adequately partitioned sub-matrices of $\mathbf{R}$. Using (16), $\gamma^{(l)}\left(\mathbf{x}_k^N\right)$ can be expressed as

$$\gamma^{(l)}\left(\mathbf{x}_k^N\right) = \gamma^{(c)}\left(\mathbf{x}_k^N\right) + \gamma^{(b)}\left(\mathbf{x}_k^N\right) \tag{17}$$

where

$$\gamma^{(c)}\left(\mathbf{x}_k^N\right) = \left\|\mathbf{y}_k^N - \mathbf{R}_{22,k}\mathbf{x}_k^N\right\|^2 + \xi\left(\mathbf{x}_k^N\right) \tag{18}$$

$$\gamma^{(b)}\left(\mathbf{x}_k^N\right) = \left\|\mathbf{y}_1^{k-1} - \mathbf{R}_{11,k}\hat{\mathbf{x}}_1^{k-1} - \mathbf{R}_{12,k}\mathbf{x}_k^N\right\|^2 \tag{19}$$

and $\xi\left(\mathbf{x}_k^N\right) = -\sigma_n^2\sum_{i=k}^{N}\sum_{j=1}^{Q}\ln Pr\left(\bar{c}_{i,j}\right)$. Note that the term generated by *a priori* information $\xi\left(\mathbf{x}_k^N\right)$ considers only $\mathbf{x}_k^N$ since the symbols $\mathbf{x}_1^{k-1}$ are undecided. Note also that the linear MMSE estimate of the non-causal symbols $\mathbf{x}_1^{k-1}$ is given by [35]

$$\hat{\mathbf{x}}_1^{k-1} = \mathbf{F}_k\left(\mathbf{y}_1^{k-1} - E\left[\mathbf{y}_1^{k-1}\middle|\mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N\right]\right) + E\left[\mathbf{x}_1^{k-1}\middle|\mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N\right] \tag{20}$$

$$= \mathbf{F}_k\left(\mathbf{y}_1^{k-1} - \mathbf{R}_{11,k}\overline{\mathbf{x}}_1^{k-1} - \mathbf{R}_{12,k}\mathbf{x}_k^N\right) + \overline{\mathbf{x}}_1^{k-1}, \tag{21}$$

where $\overline{\mathbf{x}}_1^{k-1} = E[\mathbf{x}_1^{k-1}]$ and $\mathbf{F}_k = \mathrm{Cov}(\mathbf{x}_1^{k-1}, \mathbf{y}_1^{k-1} | \mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N)\ \mathrm{Cov}^{-1}(\mathbf{y}_1^{k-1} | \mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N)$. We can obtain $\overline{\mathbf{x}}_1^{k-1}$ and $\mathbf{F}_k$ from *a priori* LLRs as [2]

$$\overline{\mathbf{x}}_1^{k-1} = \begin{bmatrix} \sum_{\theta \in \Theta} \theta \prod_{j=1}^{Q} \frac{1}{2} \left(1 + \overline{c}_{1,j} \tanh\left(\frac{L_{\mathrm{pri}}(\overline{c}_{1,j})}{2}\right)\right) \\ \vdots \\ \sum_{\theta \in \Theta} \theta \prod_{j=1}^{Q} \frac{1}{2} \left(1 + \overline{c}_{k-1,j} \tanh\left(\frac{L_{\mathrm{pri}}(\overline{c}_{k-1,j})}{2}\right)\right) \end{bmatrix} \tag{22}$$

$$\mathbf{F}_k = \mathbf{\Lambda}_k (\mathbf{R}_{11,k})^H \left((\mathbf{R}_{11,k})\, \mathbf{\Lambda}_k (\mathbf{R}_{11,k})^H + \sigma_n^2 \mathbf{I} \right)^{-1}, \tag{23}$$

where $\mathbf{\Lambda}_k = \mathrm{diag}\,(\lambda_1, \cdots, \lambda_{k-1})$ and $\lambda_i = \sum_{\theta \in \Theta} |\theta - \overline{x}_i|^2 \prod_{q=1}^{Q} \frac{1}{2}(1 + \overline{c}_{i,q} \tanh(\frac{L_{\mathrm{pri}}(\overline{c}_{i,q})}{2}))$. The set Θ includes all possible constellation points. In the first iteration of the IDD where *a priori* LLRs, $L_{\mathrm{pri}}(\overline{c}_{i,q})$ are unavailable, $\mathbf{\Lambda}_k = \mathbf{I}$ and $\overline{\mathbf{x}}_1^{k-1} = \mathbf{0}$.

Using (19) and (21), $\gamma^{(b)}\left(\mathbf{x}_k^N\right)$ can be rewritten

$$\gamma^{(b)}\left(\mathbf{x}_k^N\right) = \left\| (\mathbf{I} - \mathbf{R}_{11,k}\mathbf{F}_k) \left(\mathbf{y}_1^{k-1} - \mathbf{R}_{11,k} \overline{\mathbf{x}}_1^{k-1} - \mathbf{R}_{12,k} \mathbf{x}_k^N \right) \right\|^2 \tag{24}$$

$$= \left\| \mathbf{Z}_k \left(\mathbf{y}_1^{k-1} - \mathbf{R}_{11,k} \overline{\mathbf{x}}_1^{k-1} - \mathbf{R}_{12,k} \mathbf{x}_k^N \right) \right\|^2, \tag{25}$$

where

$$\mathbf{Z}_k = \mathbf{I} - \mathbf{R}_{11,k}\mathbf{F}_k \tag{26}$$

$$= \sigma_n^2 \left(\mathbf{R}_{11,k} \mathbf{\Lambda}_k (\mathbf{R}_{11,k})^H + \sigma_n^2 \mathbf{I} \right)^{-1}. \tag{27}$$

Further, denoting $\mathbf{q}_k = \mathbf{Z}_k(\mathbf{y}_1^{k-1} - \mathbf{R}_{11,k}\overline{\mathbf{x}}_1^{k-1})$ and $\mathbf{P}_k = \mathbf{Z}_k\mathbf{R}_{12,k}$, $\gamma^{(l)}\left(\mathbf{x}_k^N\right)$ can be simply expressed

$$\gamma^{(l)}\left(\mathbf{x}_k^N\right) = \gamma^{(c)}\left(\mathbf{x}_k^N\right) + \underbrace{\left\| \mathbf{q}_k - \mathbf{P}_k \mathbf{x}_k^N \right\|^2}_{\text{bias term}}. \tag{28}$$

Note that the bias term $\|\mathbf{q}_k - \mathbf{P}_k \mathbf{x}_k^N\|^2$ of the LE-LA path metric can be computed with only linear operations. Note also that *a priori* information obtained from the channel decoder is reflected through $\overline{\mathbf{x}}_1^{k-1}$ and $\mathbf{\Lambda}_k$ in the bias term.

C. Efficient Computation of Path Metric

In this subsection, we discuss how the LE-LA path metric can be computed efficiently. Recalling that the bias term is expressed as $\|\mathbf{Z}_k(\mathbf{y}_1^{k-1} - \mathbf{R}_{11,k}\overline{\mathbf{x}}_1^{k-1} - \mathbf{R}_{12,k}\mathbf{x}_k^N)\|^2$, computation of the path metric is divided into two steps; 1) computation of $\mathbf{Z}_k$ for all k prior to the tree search and 2) recursive update of the path metric for each branch extension during the search.

First, using a matrix inversion formula for block matrices [36, *Appendix 1.1.3*], the operators $\mathbf{Z}_k (k = 1, \cdots, N)$ in (27) can be computed recursively. Denoting

$$\mathbf{R}_{11,k+1} = \begin{bmatrix} \mathbf{R}_{11,k} & \mathbf{r}_{k+1} \\ \mathbf{0} & r_{k+1,k+1} \end{bmatrix}, \ \mathbf{\Lambda}_{k+1} = \begin{bmatrix} \mathbf{\Lambda}_k & \mathbf{0} \\ \mathbf{0} & \lambda_{k+1} \end{bmatrix}, \tag{29}$$

and $\mathbf{r}_{k+1} = [r_{1,k+1}, \cdots, r_{k,k+1}]^T$, then $\mathbf{Z}_{k+1}$ is expressed as a function of $\mathbf{Z}_k$ as

$$\mathbf{Z}_{k+1} = \begin{bmatrix} \mathbf{Z}_k - K\lambda_{k+1}\mathbf{Z}_k\mathbf{r}_{k+1}\mathbf{r}_{k+1}^H\mathbf{Z}_k & -K\lambda_{k+1}r_{k+1,k+1}\mathbf{Z}_k\mathbf{r}_{k+1} \\ -K\lambda_{k+1}r_{k+1,k+1}\mathbf{r}_{k+1}^H\mathbf{Z}_k & K\left(\lambda_{k+1}\mathbf{r}_{k+1}^H\mathbf{Z}_k\mathbf{r}_{k+1} + \sigma_n^2\right) \end{bmatrix}, \tag{30}$$

where

$$K = \frac{1}{\lambda_{k+1}\left(\mathbf{r}_{k+1}^H\mathbf{Z}_k\mathbf{r}_{k+1} + r_{k+1,k+1}^2\right) + \sigma_n^2}. \tag{31}$$

In particular, $\mathbf{Z}_2 = \frac{\sigma_n^2}{\lambda_1 r_{1,1}^2 + \sigma_n^2}$. See Appendix B for the derivation of (30). If the *a priori* LLRs are all zero, $\mathbf{Z}_k$ does not need to be computed for every symbol time as long as the channel remains constant. If the *a priori* LLRs are non-zero, these steps are performed for each symbol. However, the required computations can be further reduced by replacing the instantaneous covariance matrix $\mathbf{\Lambda}_k$ by its time-average over a coherent time [1].

Next, the LE-LA path metric can be recursively updated for each tree extension. At the root node, a vector is defined such that $\mathbf{a}_{N+1} = \mathbf{y} - \mathbf{R}\overline{\mathbf{x}}_1^N$. The vector $\mathbf{a}_k$ is updated from that of its parent node as

$$\begin{bmatrix} \mathbf{a}_k \\ v_k \end{bmatrix} = \mathbf{a}_{k+1} - \begin{bmatrix} r_{1,k} & \cdots & r_{k,k} \end{bmatrix}^T (x_k - \overline{x}_k), \tag{32}$$

where v_k is a scaler variable. Using the vector for each path $\mathbf{x}_k^N$, the LE-LA path metric can be obtained as

$$\gamma^{(l)}\left(\mathbf{x}_k^N\right) = \gamma^{(c)}\left(\mathbf{x}_k^N\right) + p^{(b)}\left(\mathbf{x}_k^N\right) \tag{33}$$

$$\gamma^{(c)}\left(\mathbf{x}_k^N\right) = \gamma^{(c)}\left(\mathbf{x}_{k+1}^N\right) + |v_k|^2 + \xi\left(x_k\right) \tag{34}$$

$$\gamma^{(b)}\left(\mathbf{x}_k^N\right) = \left\|\mathbf{Z}_k \cdot \mathbf{a}_k\right\|^2, \tag{35}$$

where $\gamma^{(c)}\left(\mathbf{x}_{N+1}^N\right) = 0$. Noting that the dimension of the matrix $\mathbf{Z}_k$ is $(k-1) \times (k-1)$, the number of complex multiplications for the bias term computation is proportional to $(k-1)^2$. In order to reduce the complexity, we can look ahead only $N_l (< k-1)$ symbols instead of all non-causal symbols.

TABLE I

SUMMARY OF ISS-MA

Output: $\{L_{\text{post}}(\overline{c}_{k,i})\}_{k=[1:N],i=[1:Q]}$

Input: $\mathbf{y}$, $\mathbf{H}$, $\{L_{\text{pri}}(\overline{c}_{k,i})\}_{k=[1:N],i=[1:Q]}$, N_l and J

STEP 1: (Preprocessing) Order $\mathbf{x}$ and $\mathbf{H}$ according to V-BLAST ordering [24] or B-Chase preprocessing [12]. Then, compute $\mathbf{Z}_k$ for all k.

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 ith level.

STEP 4: If $i > 1$, choose the M best paths with the smallest $\gamma^{(l)}\left(\mathbf{x}_i^N\right)$ and go to STEP 3 with $i = i - 1$. Otherwise, store all $2^Q M$ survival candidates into the list $\mathcal{L}$ and go to STEP 5.

STEP 5: (List extension & APP calculation) For each value of k and i, compute $\{L_{\text{post}}(\overline{c}_{k,i})\}$ based on $\mathcal{L}$. If the value of $\overline{c}_{k,i}$ for all elements of $\mathcal{L}$ is either +1 or -1, the value of $\overline{c}_{k,i}$ of the best J candidates (associated with the minimum cost metric) is flipped and these counter-hypothesis candidates are added to $\mathcal{L}$ to generate the extended list $\mathcal{L}_k^{\text{ext}}$. The APP is calculated over the extended list based on (37).

Towards this goal, we set $\alpha = \max(0, k - N_l)$ and repartition the system as

$$\begin{bmatrix} \mathbf{y}_\alpha^{k-1} \\ \mathbf{y}_k^N \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{R}}_{11,k} & \overline{\mathbf{R}}_{12,k} \\ \mathbf{0} & \overline{\mathbf{R}}_{22,k} \end{bmatrix} \begin{bmatrix} \mathbf{x}_\alpha^{k-1} \\ \mathbf{x}_k^N \end{bmatrix} + \begin{bmatrix} \mathbf{n}_\alpha^{k-1} \\ \mathbf{n}_k^N \end{bmatrix}, \tag{36}$$

where $\overline{\mathbf{R}}_{11,k}$ and $\overline{\mathbf{R}}_{12,k}$ are the redefined sub-matrices of (16), respectively. In this case, the bias term defined in Section III-B needs to be modified based on this partitioning. In doing so, the dimension of $\overline{\mathbf{R}}_{11,k}$ and $\mathbf{Z}_k$ is reduced from $(k-1) \times (k-1)$ to $N_l \times N_l$. The recursive computation of $\mathbf{Z}_k$ employing the new partitioning can be derived without matrix inversion (see [37, *Section III. A*]). In addition, in (35), we only need to multiply $\mathbf{Z}_k$ with the last α elements of $\mathbf{a}_k$. Overall, by using only N_l non-causal symbols for the bias term, the number of operations for the bias computation can be reduced from $M \sum_{k=1}^{N} (k-1)^2 (= M/6 \cdot (2N^3 - 3N^2 + N))$ to $MN \cdot N_l^2$.

D. Application to APP Detection

In this section, we introduce the soft-input soft-output tree detection algorithm employing the LE-LA path metric. To reduce errors in early detection stage, symbol detection ordering is performed

first. The V-BLAST ordering [24] or B-Chase preprocessing [12] can be adopted. Note that the B-Chase preprocessing is preferred when M is larger than the constellation size 2^Q. In each level of the tree, $\gamma^{(l)}\left(\mathbf{x}_k^N\right)$ of $2^Q M$ survival paths are compared and then the M best paths are selected. Starting from the root node, this candidate selection procedure continues to the bottom level, eventually producing $2^Q M$ complete paths. The symbol vectors corresponding to these complete paths generate a candidate list $\mathcal{L}$, over which the extrinsic LLR for each bit is calculated. In the event that a particular bit in each of the candidates takes the same value (all one or zero), the magnitude of the generated LLR might become unduly large, limiting the error-correction capability of the channel decoder [38]. In order to prevent this situation, whenever this occurs for the kth bit of the candidate list, the kth bits of the best J candidates ($J \leq 2^Q M$) are flipped and added into the candidate list $\mathcal{L}$, generating an extended list $\mathcal{L}_k^{\text{ext}}$. As a result, the size of $\mathcal{L}_k^{\text{ext}}$ becomes $2^Q M + J$. Using the list $\mathcal{L}_k^{\text{ext}}$ together with the max-log approximation, the APP becomes

$$L_{\text{post}}\left(\overline{c}_{k,i}\right) \approx \max_{\mathbf{x} \in \mathcal{L}_k^{\text{ext}} \cap X_{k,i}^{+1}} \psi\left(\mathbf{x}\right) - \max_{\mathbf{x} \in \mathcal{L}_k^{\text{ext}} \cap X_{k,i}^{-1}} \psi\left(\mathbf{x}\right). \tag{37}$$

A summary of the ISS-MA is provided in Table I.

IV. Performance Analysis

We discussed in the previous section that the transmitted symbols are always found with $M = 1$ if the genie-aided path metric is used. Relaxation of the finite alphabet constraint and Gaussian approximation are made for undecided symbols to derive the LE-LA path metric. In this section, we analyze the performance of the proposed M-algorithm employing the LE-LA path metric. As a measure for performance, we consider the probability of a CPL event, i.e., the probability that the tree search rejects a path associated with the transmitted symbols. In order to make the analysis tractable, we focus on the case when $M = 1$. Although our analysis focuses only on the case $M = 1$, it is clear that lower CPL rate for $M = 1$ implies more chances of a correct symbol being selected for $M > 1$ as well. The performance analysis for $M > 1$ is presented via computer simulations in Section V-B.

Given the channel matrix $\mathbf{R}$ and the *a priori* LLRs, the probability of CPL can be expressed as

$$P_{\text{CPL}} = 1 - \Pr\left(\widetilde{\mathbf{x}} \in \mathcal{L} | \widetilde{\mathbf{x}} \text{ is sent}\right) \tag{38}$$

$$= 1 - \prod_{k=1}^{N} \overline{\Pr}\left(\widetilde{\mathbf{x}}_k^N \in \mathcal{L}_k \middle| \widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}\right) \tag{39}$$

$$= 1 - \prod_{k=1}^{N} \left(1 - \overline{\Pr}\left(\widetilde{\mathbf{x}}_k^N \notin \mathcal{L}_k \middle| \widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}\right)\right), \tag{40}$$

where $\mathcal{L}_k$ denotes the set of the paths selected at the kth level and $\overline{\Pr}(\cdot)$ is the probability given that $\widetilde{\mathbf{x}}$ is sent. Since we consider the case of $M = 1$, $\widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}$ implies that a correct path has been selected up to the $(k+1)$th level. With this setup and from (16), (18), and (25), one can show that $\gamma^{(l)}\left(\mathbf{x}_k^N\right)$ is given by

$$\gamma^{(l)}\left(\mathbf{x}_k^N\right) = \underbrace{\left\| \mathbf{y}_k^N - \mathbf{R}_{22,k} \begin{bmatrix} x_k \\ \widetilde{\mathbf{x}}_{k+1}^N \end{bmatrix} \right\|^2 + \xi\left(x_k\right) + \xi\left(\widetilde{\mathbf{x}}_{k+1}^N\right)}_{\gamma^{(c)}\left(\mathbf{x}_k^N\right)} + \underbrace{\left\| \mathbf{q}_k - \mathbf{P}_k \begin{bmatrix} x_k \\ \widetilde{\mathbf{x}}_{k+1}^N \end{bmatrix} \right\|^2}_{\gamma^{(b)}\left(\mathbf{x}_k^N\right)} \tag{41}$$

$$= \left| r_{k,k}\left(\widetilde{x}_k - x_k\right) + n_k \right|^2 + \sum_{i=k+1}^{N} |n_i|^2 + \left\| \mathbf{Z}_k \mathbf{r}_k \left(\widetilde{x}_k - x_k\right) + \mathbf{Z}_k \mathbf{b}_k \right\|^2 + \xi\left(x_k\right) + \xi\left(\widetilde{\mathbf{x}}_{k+1}^N\right)$$

$$= \left\| \begin{bmatrix} \mathbf{Z}_k \mathbf{r}_k \\ r_{k,k} \end{bmatrix} \left(\widetilde{x}_k - x_k\right) + \begin{bmatrix} \mathbf{Z}_k \mathbf{b}_k \\ n_k \end{bmatrix} \right\|^2 + \xi(x_k) + \sum_{i=k+1}^{N} |n_i|^2 + \xi\left(\widetilde{\mathbf{x}}_{k+1}^N\right) \tag{42}$$

$$= \left\| \sqrt{\mathbf{r}_k^H \mathbf{Z}_k \mathbf{r}_k + |r_{k,k}|^2}\left(\widetilde{x}_k - x_k\right) + \frac{\begin{bmatrix} \mathbf{Z}_k \mathbf{r}_k & r_{k,k} \end{bmatrix}}{\sqrt{\mathbf{r}_k^H \mathbf{Z}_k \mathbf{r}_k + |r_{k,k}|^2}} \begin{bmatrix} \mathbf{Z}_k \mathbf{b}_k \\ n_k \end{bmatrix} \right\|^2 + \xi(x_k) + C \tag{43}$$

where $\mathbf{b}_k = \mathbf{R}_{11,k}\left(\widetilde{\mathbf{x}}_1^{k-1} - \overline{\mathbf{x}}_1^{k-1}\right) + \mathbf{n}_1^{k-1}$, and $\mathbf{r}_k = \mathbf{R}_{12,k}\mathbf{e}_1 = [r_{1,k}, \cdots, r_{k-1,k}]^T$. Note that C is independent of the selection of x_k. The first term in (43) can be interpreted as the distance metric between the output of a scalar additive noise channel

$$\xi_k = \sqrt{\mathbf{r}_k^H \mathbf{Z}_k \mathbf{r}_k + |r_{k,k}|^2}\,\widetilde{x}_k + \frac{\begin{bmatrix} \mathbf{Z}_k \mathbf{r}_k & r_{k,k} \end{bmatrix}}{\sqrt{\mathbf{r}_k^H \mathbf{Z}_k \mathbf{r}_k + |r_{k,k}|^2}} \begin{bmatrix} \left(\mathbf{Z}_k \mathbf{b}_k\right)^T & n_k \end{bmatrix}^T \tag{44}$$

and a symbol candidate $\sqrt{\mathbf{r}_k^H \mathbf{Z}_k \mathbf{r}_k + |r_{k,k}|^2}\,x_k$. The ISS-MA chooses the M best symbols x_k according to the cost metric in (43). Since the *a priori* term $\xi(x_k)$ in (43) leads to better detection, we ignore the impact of it in our discussion. If we let $E\left[\mathbf{b}_k \mathbf{b}_k^H\right] = \mathbf{\Sigma}_k = \left(\mathbf{R}_{11,k} \mathbf{\Lambda}_k \mathbf{R}_{11,k}^H + \sigma_n^2 \mathbf{I}\right)$ and $\mathbf{Z}_k = \sigma_n^2 \mathbf{\Sigma}_k^{-1}$, then the signal to interference plus noise ratio (SINR) of the scalar additive noise channel is given

by

$$\text{SINR} = \frac{\left(\mathbf{r}_k^H \mathbf{Z}_k \mathbf{r}_k + |r_{k,k}|^2\right)^2}{\mathbf{r}_k^H \mathbf{Z}_k^2 E\left[\mathbf{b}_k \mathbf{b}_k^H\right] \mathbf{Z}_k^2 \mathbf{r}_k + \sigma_n^2 |r_{k,k}|^2} \tag{45}$$

$$= \frac{1}{\sigma_n^2} \frac{\left(\mathbf{r}_k^H \left(\sigma_n^4 \mathbf{\Sigma}_k^{-2}\right) \mathbf{r}_k + |r_{k,k}|^2\right)^2}{\mathbf{r}_k^H \left(\sigma_n^6 \mathbf{\Sigma}_k^{-3}\right) \mathbf{r}_k + |r_{k,k}|^2}. \tag{46}$$

Lemma 4.1: The SINR in (46) is bounded by

$$\sigma_n^2 \mathbf{r}_k^H \mathbf{\Sigma}_k^{-2} \mathbf{r}_k + \frac{|r_{k,k}|^2}{\sigma_n^2} \leq \text{SINR} \leq \mathbf{r}_k^H \mathbf{\Sigma}_k^{-1} \mathbf{r}_k + \frac{|r_{k,k}|^2}{\sigma_n^2}. \tag{47}$$

Proof: See Appendix C. ■

Taking similar steps, one can show that the SINR for the causal path metric $\gamma^{(c)}\left(\mathbf{x}_k^N\right)$ is $\frac{|r_{k,k}|^2}{\sigma_n^2}$. Hence, $\mathbf{r}_k^H \mathbf{\Sigma}_k^{-1} \mathbf{r}_k^H$ and $\sigma_n^2 \mathbf{r}_k^H \mathbf{\Sigma}_k^{-2} \mathbf{r}_k^H$ can be regarded as upper and lower bounds on the SINR gain achieved by the LE-LA path metric, respectively. It is of interest to check the behavior of the upper and lower bound of SINR gain for high dimensional systems. Suppose that $N, L \to \infty$ with a fixed aspect ratio $\beta = N/L$ $(0 < \beta \leq 1)$, and let $\lambda_{\min}$ and $\lambda_{\max}$ be smallest and largest diagonals of $\mathbf{\Lambda}_k$, respectively. Then, we attain a looser bound on the SINR is

$$\underbrace{\sigma_n^2 \mathbf{r}_k^H (\sigma_n^2 \mathbf{I} + \lambda_{\max} \mathbf{R}_{11,k} \mathbf{R}_{11,k}^H)^{-2} \mathbf{r}_k}_{B_k^{\text{lower}}} + \frac{|r_{k,k}|^2}{\sigma_n^2} \leq \text{SINR} \leq \underbrace{\mathbf{r}_k^H (\sigma_n^2 \mathbf{I} + \lambda_{\min} \mathbf{R}_{11,k} \mathbf{R}_{11,k}^H)^{-1} \mathbf{r}_k}_{B_k^{\text{upper}}} + \frac{|r_{k,k}|^2}{\sigma_n^2}, \tag{48}$$

where the upper and lower bound of the SINR gain are denoted as B_k^{upper} and B_k^{lower}, respectively. Note that (48) can be shown by the relationship $\mathbf{B} \preceq \mathbf{\Sigma}_k \preceq \mathbf{A}$ (equivalently, $\mathbf{\Sigma}_k^{-1} \preceq \mathbf{B}^{-1}$ and $\mathbf{A}^{-2} \preceq \mathbf{\Sigma}_k^{-2}$), where $\mathbf{A} = \sigma_n^2 \mathbf{I} + \lambda_{\max} \mathbf{R}_{11,k} \mathbf{R}_{11,k}^H$ and $\mathbf{B} = \sigma_n^2 \mathbf{I} + \lambda_{\min} \mathbf{R}_{11,k} \mathbf{R}_{11,k}^H$ and $X \succeq 0$ implies that the matrix X is positive semi-definite.

Theorem 4.2: For an $L \times N$ matrix $\mathbf{H}$ whose elements are i.i.d. random variables with zero mean and variance $\frac{1}{L}$, the upper and lower bound of the SINR gain for the level $k = \gamma N + 1$ $(0 < \gamma < 1)$ converge to

$$B_k^{\text{upper}} \longrightarrow B_k^{\text{upper},\infty} = \frac{1}{2\lambda_{\min}} \left(-1 - (1 - \gamma\beta) \frac{\lambda_{\min}}{\sigma_n^2} + G\left(\frac{\lambda_{\min}}{\sigma_n^2}, \gamma\beta \right) \right) \tag{49}$$

$$B_k^{\text{lower}} \longrightarrow B_k^{\text{lower},\infty} = \frac{1}{2\sigma_n^2} \left(-(1 - \gamma\beta) + \frac{1 + \gamma\beta + (1 - \gamma\beta)^2 \frac{\lambda_{\max}}{\sigma_n^2}}{G\left(\frac{\lambda_{\max}}{\sigma_n^2}, \gamma\beta \right)} \right) \tag{50}$$

as $N, L \to \infty$ with $\beta = N/L$, where $G(x, b) = \sqrt{1 + 2(1 + b)x + (1 - b)^2 x^2}$.

Proof: See Appendix D. ■

Corollary 4.3: As $\sigma_n^2 \to 0$, we have

$$B_k^{\text{upper},\infty} \longrightarrow \lambda_{\min} \frac{\gamma\beta}{(1-\gamma\beta)} \tag{51}$$

$$B_k^{\text{lower},\infty} \longrightarrow 0. \tag{52}$$

In addition, as $\sigma_n^2 \to 0$, $B_k^{\text{upper},\infty}$ monotonically increases and approaches $\lambda_{\min}\frac{\gamma\beta}{(1-\gamma\beta)}$.

We can deduce from (51) and (52) that the actual SINR gain approaches a deterministic value between $[0, \lambda_{\min}\frac{\gamma\beta}{(1-\gamma\beta)}]$. One can also show that both $B_k^{\text{upper},\infty}$ and $B_k^{\text{lower},\infty}$ are an increasing functions of $\gamma\beta \in (0,1)$. Noting that γ indicates an index for tree depth, the SINR bounds achieve their maximum at the top level of the tree ($k = N$).

Next, we analyze the probability of CPL using the SINR obtained. It is worth taking a close look at the term $\mathbf{Z}_k\mathbf{b}_k$ in (44). Recalling that $\mathbf{b}_k = \mathbf{R}_{11,k}\left(\widetilde{\mathbf{x}}_1^{k-1} - \overline{\mathbf{x}}_1^{k-1}\right) + \mathbf{n}_1^{k-1}$ and $\mathbf{Z}_k = \sigma_n^2\left(\mathbf{R}_{11,k}\mathbf{\Lambda}_k(\mathbf{R}_{11,k})^H + \sigma_n^2\mathbf{I}\right)^{-1}$, $\mathbf{Z}_k\mathbf{b}_k$ is an MMSE estimate of $\mathbf{n}_1^{k-1}$ [35]. In order to make the derivation of the CPL probability more tractable, we use a Gaussian approximation for the MMSE estimation error $(\mathbf{n}_k - \mathbf{Z}_k\mathbf{b}_k)$ or equivalently, $\mathbf{Z}_k\mathbf{b}_k$. Under this approximation, we can assume that the interference plus noise of the scalar channel is Gaussian. The validity of this approximation has been supported in many asymptotic scenarios in [39] and [40]. In particular, it is shown that the Gaussian approximation is highly accurate for large problem size N [41].

Using the SINR in (46), the probability of CPL for the kth level detection can be expressed as [42]

$$\underline{\Pr}\left(\widetilde{\mathbf{x}}_k^N \notin \mathcal{L}_k \middle| \widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}, \mathbf{H}\right) \leq 4\left(1 - \frac{1}{\sqrt{2^Q}}\right) Q\left(\sqrt{K\frac{1}{\sigma_n^2}\frac{\left(\mathbf{r}_k^H\left(\sigma_n^4\mathbf{\Sigma}_k^{-2}\right)\mathbf{r}_k + |r_{k,k}|^2\right)^2}{\mathbf{r}_k^H\left(\sigma_n^6\mathbf{\Sigma}_k^{-3}\right)\mathbf{r}_k + |r_{k,k}|^2}}\right), \tag{53}$$

where $K = \frac{3}{(2^Q-1)}$. The inequality in (53) follows from the existence of *a priori* terms in (43), which lowers the actual CPL probability. From (47), we have

$$\underline{\Pr}\left(\widetilde{\mathbf{x}}_k^N \notin \mathcal{L}_k \middle| \widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}, \mathbf{H}\right) \leq 4\left(1 - \frac{1}{\sqrt{2^Q}}\right) Q\left(\sqrt{K\left(\sigma_n^2\mathbf{r}_k^H\mathbf{\Sigma}_k^{-2}\mathbf{r}_k + \frac{|r_{k,k}|^2}{\sigma_n^2}\right)}\right). \tag{54}$$

Using (54), we can analyze an average probability of CPL for a random channel $\mathbf{H}$ whose elements are independent complex Gaussian with $\mathcal{CN}(0,1)$. The average probability of CPL, denoted as $\overline{P}_{\text{CPL}}$, is given by

$$\overline{P}_{\text{CPL}} = 1 - E_{\mathbf{H}}\left[\prod_{k=1}^{N}\left(1 - \underline{\Pr}\left(\widetilde{\mathbf{x}}_k^N \notin \mathcal{L}_k \middle| \widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}, \mathbf{H}\right)\right)\right] \tag{55}$$

$$= \sum_{k=1}^{N} E_{\mathbf{H}}\left[\underline{\Pr}\left(\widetilde{\mathbf{x}}_k^N \notin \mathcal{L}_k \middle| \widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}, \mathbf{H}\right)\right] + \text{higher order terms}, \tag{56}$$

where $E_{\mathbf{H}}[\cdot]$ denotes the expectation over $\mathbf{H}$. The average CPL probability is obtained after evaluating $E_{\mathbf{H}}\left[\underline{\Pr}\left(\widetilde{\mathbf{x}}_k^N \notin \mathcal{L}_k \middle| \widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}, \mathbf{H}\right)\right]$ for all k. In our analysis, we do not put our emphasis on the higher order terms since they become negligible in the high SNR regime. Using the relationship $Q(\sqrt{x+y}) \leq Q(\sqrt{x}) \exp\left(-\frac{y}{2}\right)$ for $x, y > 0$ and from (54), we have

$$E_{\mathbf{H}}\left[\underline{\Pr}\left(\widetilde{\mathbf{x}}_k^N \notin \mathcal{L}_k \middle| \widetilde{\mathbf{x}}_{k+1}^N \in \mathcal{L}_{k+1}, \mathbf{H}\right)\right] \tag{57}$$

$$\leq 4\left(1 - \frac{1}{\sqrt{2^Q}}\right) E_{\mathbf{H}}\left[Q\left(\sqrt{K\frac{|r_{k,k}|^2}{\sigma_n^2}}\right) \exp\left(-\frac{K\sigma_n^2 \mathbf{r}_k^H \mathbf{\Sigma}_k^{-2} \mathbf{r}_k}{2}\right)\right] \tag{58}$$

$$= 4\left(1 - \frac{1}{\sqrt{2^Q}}\right) E_{\mathbf{H}}\left[Q\left(\sqrt{K\frac{|r_{k,k}|^2}{\sigma_n^2}}\right)\right] E_{\mathbf{H}}\left[\exp\left(-\frac{K\sigma_n^2 \mathbf{r}_k^H \mathbf{\Sigma}_k^{-2} \mathbf{r}_k}{2}\right)\right], \tag{59}$$

where (59) follows from independence of $r_{k,k}$ and $\mathbf{r}_k$. Noting that $r_{k,k}$ has a Chi-square distribution with $2(L-k+1)$ degrees of freedom and $\mathbf{r}_k$ has independent complex Gaussian elements [43, *Lemma 2.1*], we have [44],

$$E_{\mathbf{H}}\left[Q\left(\sqrt{\frac{K}{\sigma_n^2}|r_{k,k}|^2}\right)\right] = \left(\frac{1}{2} - \frac{1}{2}\sqrt{\frac{K}{K+2\sigma_n^2}}\right)^{L-k+1} \sum_{l=0}^{L-k} \binom{L-k+l}{l} \left(\frac{1}{2} + \frac{1}{2}\sqrt{\frac{K}{K+2\sigma_n^2}}\right)^l \tag{60}$$

Lemma 4.4: An upper bound on the scaling gain in (59) is given by

$$E_{\mathbf{H}}\left[\exp\left(-\frac{K\sigma_n^2 \mathbf{r}_k^H \mathbf{\Sigma}_k^{-2} \mathbf{r}_k}{2}\right)\right] \leq \int_0^\infty \cdots \int_0^\infty \left(\prod_{i=1}^{k-1} \frac{1}{1 + \frac{K}{2}\frac{\sigma_n^2}{(\lambda_{\max} x_i + \sigma_n^2)^2}}\right) \times f_{\eta_1, \cdots, \eta_{k-1}}(x_1, \cdots, x_{k-1})\, dx_1 \cdots dx_{k-1}, \tag{61}$$

where

$$f_{\eta_1, \cdots, \eta_{k-1}}(x_1, \cdots, x_{k-1}) = \frac{1}{(k-1)!} \exp\left(-\sum_{i=1}^{k-1} x_i\right) \prod_{i=1}^{k-1} \frac{x_i^{L-k+1}}{(k-1-i)!(L-i)!} \prod_{i<j}^{k-1} (x_i - x_j)^2.$$

Proof: See Appendix E. ∎

While $\exp\left(-\frac{K\sigma_n^2 \mathbf{r}_k^H \mathbf{\Sigma}_k^{-2} \mathbf{r}_k}{2}\right)$ in (61) tends to one as $\sigma_n^2 \to 0$, (60) decreases to zero with a slope $\lim_{\sigma_n^2 \to 0} \ln(P_e)/\ln(\sigma_n^2) = L-k+1$. Therefore, at high SNR, the probability of CPL for the top level ($k = N$) would dominate, i.e.,

$$\overline{P}_{\text{CPL}} \lessapprox 4\left(1 - \frac{1}{\sqrt{2^Q}}\right) E_{\mathbf{H}}\left[Q\left(\sqrt{K\frac{|r_{N,N}|^2}{\sigma_n^2}}\right)\right] E_{\mathbf{H}}\left[\exp\left(-\frac{K\sigma_n^2 \mathbf{r}_N^H \mathbf{\Sigma}_N^{-2} \mathbf{r}_N}{2}\right)\right], \tag{62}$$

where the right-hand side is obtained from (60) and (61). Following similar steps, we can also show that the upper bound of the average CPL probability for the causal path metric becomes

$$\overline{P}_{\text{CPL}}^{\text{causal}} \lessapprox 4\left(1 - \frac{1}{\sqrt{2^Q}}\right) E_{\mathbf{H}}\left[Q\left(\sqrt{K\frac{|r_{N,N}|^2}{\sigma_n^2}}\right)\right]. \tag{63}$$

We observe from (62) that the average CPL probability of the LE-LA path metric is smaller than that of the causal path metric by the factor of $E_{\mathbf{H}}\left[\exp\left(-\frac{K}{2}\sigma_n^2\mathbf{r}_N^H\boldsymbol{\Sigma}_N^{-2}\mathbf{r}_N\right)\right]$. Since this term is strictly less than unity, it corresponds to the scaling gain obtained from the LE-LA path metric.

In Fig. 2, we provide the plot of the average CPL probability versus SNR for several system sizes ($N = 5, 10, 15$, and 20). We assume uncoded QPSK transmission. The average CPL rate and its upper bound are obtained from (53) and (54). For a comprehensive view, we also include the average CPL rate for the causal path metric in (63). For all cases considered, the CPL expression in (53) is quite close to that obtained from the simulation results, supporting the accuracy of the analytic bound we obtained. In particular, the upper bound of the average CPL rate appears tight at high SNR. Fig. 3 shows how the scaling gain in (61) varies as a function of SNR and system size. We observe that the performance gain of the LE-LA path metric improves with system size and the maximum is achieved in low to moderate SNR range (10 dB $\sim$ 20 dB). Notice that this behavior is desirable for IDD, since the performance in low-to-mid SNR range is critical in triggering performance improvement though iterations [45].

V. Simulation and Discussion

In this section, we evaluate the performance of the ISS-MA through computer simulations. First, we observe the performance of the soft-input soft-output M-algorithm employing the LE-LA path metric and that employing the conventional path metric. Note that the LE-LA path metric is not restricted to a particular search scheme and can be extended to more sophisticated breadth-first search algorithms (such as [12] and [23]). Next, we compare the performance-complexity trade-off of the ISS-MA with the existing soft-input soft-output detectors.

A. Simulation Setup

The simulation setup for the IDD system is as follows. A total of 2×10^5 information bits are randomly generated. A rate $R = 1/2$ recursive systematic convolutional (RSC) code with feedback polynomial $1 + D + D^2$ and feedforward polynomial $1 + D^2$. We use a random interleaver of size of $12,000$ bits. We use a gray mapping for QAM modulation. We assume fast fading channels where each entry of $\mathbf{H}$ is i.i.d. complex Gaussian $\mathcal{CN}(0, 1)$ and perfect knowledge of the channel state at the receiver is assumed. For the channel decoding, a *max-log-MAP* decoder [7] is employed. The SNR is defined as SNR $= 10\log_{10}(N/\sigma_n^2)$. Computational complexity of detectors is measured by

counting the average number of complex multiplications per symbol period and per iteration.[1]

B. Simulation Results

First, we compare the performance of the causal path metric and the LE-LA path metric. We consider the 12×12 16-QAM MIMO system, which requires high detection complexity. For fair comparison, we employ the same candidate extension strategy with $J = 16$ (described in Section III-D) for both algorithms. The parameter N_l is set to 5 for the LE-LA path metric. In Fig 4, the plots of bit error rate (BER) versus SNR are provided for several M values ($M = 4, 6, 8$ and 12). Each plot shows the BER curves obtained after different number of iterations. The ISS-MA outperforms the conventional M-algorithm for all M values and after each iteration. In particular, with $M = 4$, the ISS-MA shows remarkable performance gain (more than 5 dB gain). Then, the performance gap decreases as M increases. Note that the ISS-MA maintains strong performance even with small M (e.g. $M = 4$). Table II provides computational complexity of both algorithms along with the SNR required to achieve the BER of 10^{-2} for the same setup. The SNR is measured after the 7th iteration. In order to compare performance-complexity trade-off, it is worth looking at the performance of the ISS-MA with $M = 4$ and the M-algorithm with $M = 8$, where both algorithms require similar computational complexity. In these cases, the ISS-MA achieves almost 1 dB performance gain. Besides, we can additionally observe that the performance of the ISS-MA converges faster than the conventional M-algorithm, which might also help reducing the complexity of the ISS-MA by the early termination of the iterations.

Next, we take a look at how performance gap between the ISS-MA and the conventional M-algorithm changes in terms of different system size. Table III presents the SNR at 10^{-3} BER and complexity of both algorithms for $N = L = 6, 8, 10$, and 12. 16-QAM is used and N_l and M are set to 5 and 6 for all cases. The performance is measured after the 7th iteration. We observe that the performance gain due to the LE-LA path metric increases with system size. In particular, the gain of the ISS-MA for 6×6 system is 0.5 dB and that increases to 1.75 dB for the 12×12 system. This clearly demonstrates that future cost plays a key role for large systems.

We next investigate the performance of the ISS-MA as a function of the parameter N_l (see Section III-C). In our simulations, the 12×12 16-QAM transmission is considered and M is set to 8. Fig. 5

[1]The complexity for QR decomposition and detection ordering is not considered since they are common in all detection algorithms under consideration.

(a) and (b) show the performance and complexity of the ISS-MA for different N_l. Note that the ISS-MA with $N_l = 0$ reduces to the conventional M-algorithm. As the parameter N_l increases, the ISS-MA accounts for the further future cost so that the computational complexity increases and performance improves. The ISS-MA offers performance-complexity trade-off through N_l. While the performance of the ISS-MA improves much for small N_l values, the effect of N_l diminishes with larger N_l. It is shown that the choice of $N_l = 5$ is sufficient to achieve the maximal performance gain offered by the LE-LA path metric.

Finally, we check the performance-complexity trade-off of the ISS-MA along with those of the existing soft-input soft-output tree detection algorithms. For comprehensive picture, we consider the following algorithms;

1) MMSE-PIC algorithm; MMSE parallel interference cancellation detector [1], [2]. This detector subtracts *a priori* estimates of the interfering symbols from the received vector and then applies a linear MMSE estimator to obtain soft estimate of the symbols.
2) LISS algorithm ($|S|, |S_x|$); List sequential stack algorithm [13]. It is characterized by the size of stack $|S|$ and that of auxiliary stack $|S_x|$.
3) LFCSD algorithm ($N_{\mathcal{L}}, N_{\mathcal{S}_e}$); List fixed complexity sphere decoder [14]. A candidate list is found by the fixed complexity sphere decoder proposed in [20]. This detector is characterized by $N_{\mathcal{L}}$ and $N_{\mathcal{S}_e}$, which represent the size of the candidate list and the number of paths fully extended, respectively.
4) ITS algorithm (M); Iterative tree search [11]. This detector uses the conventional M-algorithm to find the candidate list.

Although the LSD [3] and single (parallel) tree search (STS) [9], [10] are considered as powerful detection schemes, their complexities grow so rapidly with a problem size hence they are infeasible for the 12×12 system. For this reason, we only consider fixed-complexity detectors. In Fig. 6, the performance and complexity of each algorithm are drawn in the same plot to compare the performance-complexity trade-off of the detectors. Due to a linear structure, the MMSE-PIC achieves the lowest complexity among all candidates. In addition, the performance of the MMSE-PIC is better than that of the LISS, LFCSD, and ITS. This would be why the performance of the latter detectors depends on a candidate size and the size is not large enough to achieve good performance in the 12×12 system. In particular, due to the limited stack size, the stack memory used in the LISS easily gets full before reaching a leaf of the tree tso that the LISS often fails to find reliable candidates. Fig. 6 shows

that only the ISS-MA can achieve the better performance than the MMSE-PIC. Due to improved candidate selection process, the ISS-MA finds reliable candidates only with small candidate size, thereby yielding the best BER performance while maintaining reasonable complexity. In conclusion, the ISS-MA achieves the best performance-complexity trade-off among all tree detectors considered. In addition, the ISS-MA provides the performance gain over the MMSE-PIC at the expense of higher but manageable complexity.

VI. Conclusions

In this paper, we discussed a new path metric, which shows great promise in terms of its performance-complexity trade-off for soft-input soft-output tree detection in an IDD system. By accounting for non-causal symbols in the *linear estimate-based look-ahead* (LE-LA) path metric, the performance gains over the existing causal path metric are achieved. We apply the LE-LA path metric to the soft-input soft-output M-algorithm. By adopting the sorting mechanism exploiting the LE-LA path metric, we could improve the chance of selecting the correct path dramatically, thereby achieving good detection and decoding performance with fewer iterations. From CPL probability analysis, we observed that the LE-LA path metric reflects the reliability of path much better than causal path metric. Computer simulations confirm that the proposed ISS-MA can be a promising candidate for soft-input soft-output detection in high dimensional systems.

Appendix A
Proof of Theorem 3.2

The transformed vector $\mathbf{y}$ can be expressed as $\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{n}$, where $\mathbf{n} = \mathbf{Q}_1\mathbf{n}_o$. Let k be the current layer being searched then $\mathbf{y}$, $\mathbf{x}$, and $\mathbf{n}$ can be partitioned into two $(k-1)\times 1$ and $(N-k+1)\times 1$ vectors, i.e.,

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^{k-1} \\ \mathbf{y}_k^N \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{11,k} & \mathbf{R}_{12,k} \\ \mathbf{0} & \mathbf{R}_{22,k} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^{k-1} \\ \mathbf{x}_k^N \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1^{k-1} \\ \mathbf{n}_k^N \end{bmatrix}, \tag{64}$$

where the upper triangular matrix $\mathbf{R}$ is partitioned into four sub-matrices. Given the transmitted symbol $\mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N$, *a posteriori* probability of $\mathbf{x}_1^{k-1}$ is given by

$$\ln Pr\left(\mathbf{x}_1^{k-1}\middle|\mathbf{y}, \mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N\right) \tag{65}$$

$$= \ln Pr\left(\mathbf{y}\middle|\mathbf{x}_1^{k-1}, \mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N\right) + \ln Pr(\mathbf{x}_1^{k-1}) \tag{66}$$

$$= -\ln\left(\sqrt{2\pi}\sigma_n\right) - \frac{1}{\sigma_n^2}\left\|\mathbf{y} - \mathbf{R}\begin{bmatrix}\mathbf{x}_1^{k-1}\\ \widetilde{\mathbf{x}}_k^N\end{bmatrix}\right\|^2 + \ln Pr\left(\mathbf{x}_1^{k-1}\right) \tag{67}$$

$$= -\ln\left(\sqrt{2\pi}\sigma_n\right) - \frac{1}{\sigma_n^2}\left\|\mathbf{y}_1^{k-1} - \mathbf{R}_{12,k}\widetilde{\mathbf{x}}_k^N - \mathbf{R}_{11,k}\mathbf{x}_1^{k-1}\right\|^2 - \frac{1}{\sigma_n^2}\left\|\mathbf{y}_k^N - \mathbf{R}_{22,k}\widetilde{\mathbf{x}}_k^N\right\|^2 + \ln Pr\left(\mathbf{x}_1^{k-1}\right). \tag{68}$$

Hence, we can show that

$$\check{\mathbf{x}}_1^{k-1} = \arg\max \ln Pr\left(\mathbf{x}_1^{k-1}\middle|\mathbf{y}, \mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N\right) \tag{69}$$

$$= \arg\min_{\mathbf{x}_1^{k-1}}\left\|\mathbf{y}_1^{k-1} - \mathbf{R}_{12,k}\widetilde{\mathbf{x}}_k^N - \mathbf{R}_{11,k}\mathbf{x}_1^{k-1}\right\|^2 - \sigma_n^2 \ln Pr\left(\mathbf{x}_1^{k-1}\right) \tag{70}$$

$$= \arg\min_{\mathbf{x}_1^{k-1}}\sum_{i=1}^{k-1}\left(\left|y_i - \sum_{j=k}^{N} r_{i,j}\widetilde{x}_j - \sum_{j=i}^{k-1} r_{i,j}x_j\right|^2 - \sigma_n^2\sum_{j=1}^{Q}\ln Pr\left(\widetilde{c}_{i,j}\right)\right) \tag{71}$$

$$= \arg\min_{\mathbf{x}_1^{k-1}}\sum_{i=1}^{k-1} b\left(\mathbf{x}_i^N\right) \tag{72}$$

where the equation (72) follows from the definition of the branch metric. Hence, for $\mathbf{x}_k^N = \widetilde{\mathbf{x}}_k^N$, we have $\min_{\mathbf{x}_1^{k-1}}\sum_{i=1}^{k-1} b\left(\mathbf{x}_i^N\right) = \sum_{i=1}^{k-1} b\left(\mathbf{x}_i^N\right)\big|_{\mathbf{x}_1^{k-1}=\check{\mathbf{x}}_1^{k-1}}$.

Appendix B
Proof of (30)

We can express $\mathbf{Z}_{k+1}$ in (27) as

$$\mathbf{Z}_{k+1} = \sigma_n^2\left[\mathbf{R}_{11,k+1}\mathbf{\Lambda}_{k+1}(\mathbf{R}_{11,k+1})^H + \sigma_n^2\mathbf{I}\right]^{-1} \tag{73}$$

$$= \sigma_n^2\left(\begin{bmatrix}\mathbf{R}_{11,k} & \mathbf{r}_{k+1}\\ \mathbf{0} & r_{k+1,k+1}\end{bmatrix}\begin{bmatrix}\mathbf{\Lambda}_k & \mathbf{0}\\ \mathbf{0} & \lambda_{k+1}\end{bmatrix}\begin{bmatrix}\mathbf{R}_{11,k} & \mathbf{r}_{k+1}\\ \mathbf{0} & r_{k+1,k+1}\end{bmatrix}^H + \begin{bmatrix}\sigma_n^2\mathbf{I}_k & \mathbf{0}\\ \mathbf{0} & \sigma_n^2\end{bmatrix}\right)^{-1} \tag{74}$$

$$= \sigma_n^2\left(\begin{bmatrix}\sigma_n^2\left(\mathbf{Z}_k\right)^{-1} + \lambda_{k+1}\mathbf{r}_{k+1}\mathbf{r}_{k+1}^T & \lambda_{k+1}r_{k+1,k+1}\mathbf{r}_{k+1}\\ \lambda_{k+1}r_{k+1,k+1}\mathbf{r}_{k+1}^T & \lambda_{k+1}r_{k+1,k+1}^2 + \sigma_n^2\end{bmatrix}\right)^{-1} \tag{75}$$

To obtain the update formula, for partitioned matrices, $\mathbf{A}$ given by [36, *Appendix 1.1.3*]

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

we have

$$\mathbf{A}^{-1} = \begin{bmatrix} \left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right)^{-1} & -\left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right)^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\left(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\right)^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \left(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\right)^{-1} \end{bmatrix}.$$

Let $\mathbf{A}_{11} = \sigma_n^2 (\mathbf{Z}_k)^{-1} + \lambda_{k+1}\mathbf{r}_{k+1}\mathbf{r}_{k+1}^T$, $\mathbf{A}_{12} = \lambda_{k+1} r_{k+1,k+1}\mathbf{r}_{k+1}$, $\mathbf{A}_{21} = \lambda_{k+1} r_{k+1,k+1}\mathbf{r}_{k+1}^T$, and $\mathbf{A}_{22} = \lambda_{k+1} r_{k+1,k+1}^2 + \sigma_n^2$, then (75) becomes

$$\mathbf{Z}_{k+1} = \begin{bmatrix} \mathbf{Z}_k - K\lambda_{k+1}\mathbf{Z}_k\mathbf{r}_{k+1}\mathbf{r}_{k+1}^H\mathbf{Z}_k & -K\lambda_{k+1} r_{k+1,k+1}\mathbf{Z}_k\mathbf{r}_{k+1} \\ -K\lambda_{k+1} r_{k+1,k+1}\mathbf{r}_{k+1}^H\mathbf{Z}_k & K\left(\lambda_{k+1}\mathbf{r}_{k+1}^H\mathbf{Z}_k\mathbf{r}_{k+1} + \sigma_n^2\right) \end{bmatrix}, \tag{76}$$

where

$$K = \frac{1}{\lambda_{k+1}\mathbf{r}_{k+1}^T\mathbf{Z}_k\mathbf{r}_{k+1} + \lambda_{k+1} r_{k+1,k+1}^2 + \sigma_n^2}. \tag{77}$$

Appendix C
Proof of Lemma 4.1

Let $\mathbf{\Sigma}_k$ be decomposed to $\mathbf{U}\mathbf{\Phi}_k\mathbf{U}^H$, where $\phi_1 \geq \phi_2 \geq \cdots \geq \phi_k$ are the eigenvalues of $\mathbf{\Sigma}_k$. Then, the upper bound of the SINR is given by

$$\text{SINR} = \frac{1}{\sigma_n^2} \frac{\left(\mathbf{r}_k^H\left(\sigma_n^4\mathbf{\Sigma}_k^{-2}\right)\mathbf{r}_k + |r_{k,k}|^2\right)^2}{\mathbf{r}_k^H\left(\sigma_n^6\mathbf{\Sigma}_k^{-3}\right)\mathbf{r}_k + |r_{k,k}|^2} \tag{78}$$

$$= \frac{1}{\sigma_n^2} \frac{\left(\sum_{i=1}^{k-1}\sigma_n^4\phi_i^{-2}|r_{i,k}^{'}|^2 + |r_{k,k}|^2\right)^2}{\sum_{i=1}^{k-1}\sigma_n^6\phi_i^{-3}|r_{i,k}^{'}|^2 + |r_{k,k}|^2} \tag{79}$$

$$\leq \sum_{i=1}^{k-1}\phi_i^{-1}|r_{i,k}^{'}|^2 + \frac{|r_{k,k}|^2}{\sigma_n^2} \tag{80}$$

$$= \mathbf{r}_k^H\mathbf{\Sigma}_k^{-1}\mathbf{r}_k + \frac{|r_{k,k}|^2}{\sigma_n^2}, \tag{81}$$

where $\mathbf{r}_k' = \left[r_{1,k}', \cdots, r_{k-1,k}'\right]^T = \mathbf{U}\mathbf{r}_k$ and (80) is from the *Cauchy-Schwarz* inequality.

Next, with $A = \mathbf{r}_k^H\left(\sigma_n^4\mathbf{\Sigma}_k^{-2}\right)\mathbf{r}_k + |r_{k,k}|^2$ and $B = \mathbf{r}_k^H\left(\sigma_n^6\mathbf{\Sigma}_k^{-3}\right)\mathbf{r}_k + |r_{k,k}|^2$, we can show

$$A - B = \mathbf{r}_k^H\left(\sigma_n^4\mathbf{\Sigma}_k^{-2} - \sigma_n^6\mathbf{\Sigma}_k^{-3}\right)\mathbf{r}_k \tag{82}$$

$$= \sum_{i=1}^{k-1}\left(\sigma_n^4\phi_i^{-2} - \sigma_n^6\phi_i^{-3}\right)|r_{i,k}^{'}|^2 \geq 0 \tag{83}$$

where (83) follows from $\sigma_n^2\phi_i^{-1} = \sigma_n^2/(\sigma_n^2+\epsilon) \le 1$. Hence,

$$\text{SINR} = \frac{1}{\sigma_n^2}\frac{A^2}{B} \ge \frac{1}{\sigma_n^2}A = \sigma_n^2\mathbf{r}_k^H\boldsymbol{\Sigma}_k^{-2}\mathbf{r}_k + \frac{|r_{k,k}|^2}{\sigma_n^2}. \tag{84}$$

This becomes the lower bound of the SINR.

Appendix D
Proof of Theorem 4.2

Let $\mathbf{H}_{1:k-1}$ be a matrix generated from the first $k-1$ columns of $\mathbf{H}$. Since $\mathbf{H}_{1:k-1} = \mathbf{Q}\begin{bmatrix}\mathbf{R}_{11,k}\\ \mathbf{0}\end{bmatrix}$, the matrices $\mathbf{R}_{11,k}\mathbf{R}_{11,k}^H$ and $\mathbf{H}_{1:k-1}^H\mathbf{H}_{1:k-1}$ share same eigenvalues. For an i.i.d. random matrix $\mathbf{H}$, the elements of $\mathbf{r}_k$ are zero-mean and independent with variance of $\frac{1}{L}$. According to [43, *Lemma 2.29*], as $N, L \to \infty$ with $\beta = L/N$,

$$B_k^{\text{upper}} = \mathbf{r}_k^H\left(\sigma_n^2\mathbf{I} + \lambda_{\min}\mathbf{R}_{11,k}\mathbf{R}_{11,k}^H\right)^{-1}\mathbf{r}_k \longrightarrow \frac{\gamma\beta}{\sigma_n^2}\int_0^\infty \frac{1}{1+\frac{\lambda_{\min}}{\sigma_n^2}x}f_\eta(x)dx \tag{85}$$

where $f_\eta(x)$ is an empirical eigenvalue distribution of $\mathbf{H}_{1:k-1}^H\mathbf{H}_{1:k-1}$. According to the Marcenko-Pastur law [43, *Theorem 2.35*], as $N, L \to \infty$ with $\beta = L/N$, $f_\eta(x)$ converges to

$$f_\eta(x) \longrightarrow f_\eta^o(x) = \frac{\sqrt{\left(x-(1-\sqrt{\gamma\beta})^2\right)^+\left((1+\sqrt{\gamma\beta})^2-x\right)^+}}{2\pi\gamma\beta x}, \tag{86}$$

where $(x)^+ = \max(0,x)$. Hence, from (85) and (86), we obtain

$$B_k^{\text{upper}} \longrightarrow \frac{\gamma\beta}{\sigma_n^2}\int_0^\infty \frac{1}{1+\frac{\lambda_{\min}}{\sigma_n^2}x}f_\eta^o(x)dx = \frac{1}{2\lambda_{\min}}\left(-1-(1-\gamma\beta)\frac{\lambda_{\min}}{\sigma_n^2}+G\left(\frac{\lambda_{\min}}{\sigma_n^2},\gamma\beta\right)\right). \tag{87}$$

In a similar manner, the lower bound converges to

$$B_k^{\text{lower}} = \sigma_n^2\mathbf{r}_k^H(\sigma_n^2\mathbf{I}+\lambda_{\max}\mathbf{R}_{11,k}\mathbf{R}_{11,k}^H)^{-2}\mathbf{r}_k \longrightarrow \frac{\gamma\beta}{\sigma_n^2}\int_0^\infty \frac{1}{\left(1+\frac{\lambda_{\max}}{\sigma_n^2}x\right)^2}f_\eta^o(x)dx \tag{88}$$

$$= \frac{1}{2\sigma_n^2}\left(-(1-\gamma\beta)+\frac{1+\gamma\beta+(1-\gamma\beta)^2\frac{\lambda_{\max}}{\sigma_n^2}}{G\left(\frac{\lambda_{\max}}{\sigma_n^2},\gamma\beta\right)}\right). \tag{89}$$

Appendix E
Proof of Lemma 4.4

Let $\eta_1, \eta_2, \cdots, \eta_{k-1}$ be the unordered eigenvalues of $\mathbf{R}_{11,k}\mathbf{R}_{11,k}^H$. The scaling gain in (59) can be expressed as

$$E_{\mathbf{H}}\left[\exp\left(-\frac{K}{2}\sigma_n^2\mathbf{r}_k^H\boldsymbol{\Sigma}_k^{-2}\mathbf{r}_k\right)\right] \leq E_{\mathbf{H}}\left[\exp\left(-\frac{K}{2}\sigma_n^2\mathbf{r}_k^H\left(\sigma_n^2\mathbf{I}+\lambda_{\max}\mathbf{R}_{11,k}\mathbf{R}_{11,k}^H\right)^{-2}\mathbf{r}_k\right)\right] \tag{90}$$

$$= E_{\mathbf{H}}\left[\prod_{i=1}^{k-1}\exp\left(-\frac{K}{2}\frac{\sigma_n^2}{\left(\lambda_{\max}\eta_i+\sigma_n^2\right)^2}|r_{i,k}|^2\right)\right] \tag{91}$$

$$= E_{\mathbf{R}_{11,k}}\left[\prod_{i=1}^{k-1}E_{r_{i,k}}\left[\exp\left(-\frac{K}{2}\frac{\sigma_n^2}{\left(\lambda_{\max}\eta_i+\sigma_n^2\right)^2}|r_{i,k}|^2\right)\middle|\,\mathbf{R}_{11,k}\right]\right] \tag{92}$$

$$= E_{\mathbf{R}_{11,k}}\left[\prod_{i=1}^{k-1}\frac{1}{1+\frac{K}{2}\frac{\sigma_n^2}{\left(\lambda_{\max}\eta_i+\sigma_n^2\right)^2}}\right], \tag{93}$$

where (92) is from $E[x] = E[E[x|y]]$ and (93) follows from $r_{i,k}$ being $\mathcal{CN}(0,1)$ and independent of $\mathbf{R}_{11,k}$. Let $\mathbf{H}_{1:k-1}$ be a matrix generated from the first $k-1$ columns of $\mathbf{H}$, then the matrices $\mathbf{R}_{11,k}\mathbf{R}_{11,k}^H$ and $\mathbf{H}_{1:k-1}^H\mathbf{H}_{1:k-1}$ share same eigenvalues. The pdf of the unordered eigenvalues of $\mathbf{H}_{1:k-1}^H\mathbf{H}_{1:k-1}$ is given by [46]

$$f_{\eta_1,\cdots,\eta_{k-1}}(x_1,\cdots,x_{k-1}) = \frac{1}{(k-1)!}\exp\left(-\sum_{i=1}^{k-1}x_i\right)\prod_{i=1}^{k-1}\frac{x_i^{L-k+1}}{(k-1-i)!(L-i)!}\prod_{i<j}^{k-1}(x_i-x_j)^2, \tag{94}$$

which completes the proof.

References

[1] M. Tüchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *IEEE Trans. Commun.*, vol. 50, pp. 754-767, May 2002.

[2] M. Sellathurai and S. Haykin, "Turbo-BLAST for wireless communications: theory and experiments," *IEEE Trans. Signal Processing*, vol. 50, pp. 2538-2546, Oct. 2002.

[3] B. Hochwald and S. T. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, pp. 389-399, March 2003.

[4] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 2299-2311, Nov. 2004.

[5] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 1046-1061, July 1999.

[6] C. Berrou and A. Glavieux "Near optimum error-correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp 1261-1271, Oct. 1996.

[7] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. on Telecommun.*, vol. 8, pp. 119-125, March 1997.

[8] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with soft detection based on hard sphere decoding," *IEEE Trans. Commun.*, vol. 54, pp. 587-590, April 2006.

[9] J. Jaldén and B. Ottersten, "Parallel implementation of a soft output sphere decoder," *Proc. IEEE Ashilomar Conference on Signals, Systems, and Computers*, Nov. 2005, pp. 581-585.

[10] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *IEEE Journal on Selected Areas in Commun.*, vol. 26, pp. 290-300, Feb. 2008.

[11] Y. L. C. de Jong and T. J. Wilink, "Iterative tree search detection for MIMO wireless systems," *IEEE Trans. Commun.*, vol. 53, pp. 930-935, June 2005.

[12] D. L. Milliner, E. Zimmermann, J. R. Barry, and G. Fettweis, "A fixed-compelxity smart candidate adding algorithm for soft output MIMO detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, pp. 1016-1025, Dec. 2009.

[13] J. Hagenauer and C. Kuhn, "The list-sequential (LISS) algorithm and its application," *IEEE Trans. Commun.*, vol. 55, pp. 918-928, May 2007.

[14] L. G. Barbero and T. S. Thompson, "Extending a fixed-complexity sphere decoder to obtain likelihood information for turbo-MIMO systems," *IEEE trans. Veh. Technol.*, vol. 57, no. 5, pp. 2804-2814, Sep. 2008.

[15] E. G. Larsson and J. Jaldén, "Fixed-complexity soft MIMO detection via partial marginalization," *IEEE Trans. Signal Processing*, vol. 56, pp. 3397-3407, Aug. 2008.

[16] D. Wu, J. Eilert, R. Asghar and D. Liu, "VLSI implementation of a fixed-complexity soft-output MIMO detector for high-speed wireless," *EURASIP Journal on Wireless Commun. and Networking*, vol. 2010, pp. 1-13, 2010.

[17] A. D. Murugan, H. E. Gamal, M. O. Damen, and G. Caire, "A unified framework for tree search decoding: rediscovering the sequential decoder," *IEEE Trans. Information Theory*, vol. 52, pp. 933-953, March 2006.

[18] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, pp. 463-471, April 1985.

[19] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Information Theory*, vol. 49, pp. 2389-2402, Oct. 2003.

[20] J. Jaldén, L. G. Babero, B. Ottersten, and J. S. Thompson, "Full diversity detection in MIMO systems with a fixed-complexity sphere decoder," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2007, pp. III-49-III-52.

[21] J. B. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Commun.*, vol. COM-32, no. 2, pp. 169-176, Feb. 1984.

[22] K. Wong, C. Tsui, R. S. Cheng, and W. Mow, "A VLSI architecture of the K-best lattice decoding algorithm for MIMO channels," *IEEE ISCAS, Scottsdale, AZ, USA*, May 2002, pp. 273-276.

[23] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Commun.*, vol. 24, pp. 491-503, March 2006.

[24] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizaing very high data rates over the rich-scattering wireless channel," *Proc. URSI Int. Symp. Signals, Syst., Electron.*, Sep. 1998, pp. 295-300.

[25] J. N. Nilsson, *Principle of Artificial Intelligence.* Palo Alto, CA: Tioga Publishing Co., 1980.

[26] Y. S. Han, C. R. P. Hartmann, and C. Chen, "Efficient priority-first search maximum-likelihood soft-decision decoding of linear block codes," *IEEE Trans. Information Theory,* vol. 39, pp. 1514-1523, Sep. 1993.

[27] L. Ekroot and S. Dolinar, "A* decoding of block codes," *IEEE Trans. Commun.*, vol. 44, pp. 1052-1056, Sep. 1996.

[28] M. Stojnic, H. Vikalo, and B. Hassibi, "Speeding up the sphere decoder with H^{∞} and SDP inspired lower bounds," *IEEE Trans. Signal Processing,*, vol. 56, pp. 712-726, Feb. 2008.

[29] R. Johannesson and K. S. Zigangirov, *Fundamentals of convolutional coding*, Wiley-IEEE Press, 1999.

[30] F. Xiong, A. Zerik, and E. Shwedyk, "Sequentional sequence estimation for channels with intersymbol interference of finite or infinite length," *IEEE Trans. Commun.*, vol. 38, pp. 795-804, June 1990.

[31] R. Gowaikar and B. Hassibi, "Statistical pruning for near-Maximum Likelihood Decoding," *IEEE Trans. Signal Processing*, vol. 55, pp. 2661-2675, June 2007

[32] B. Shim and I. Kang, "Sphere decoding with a probabilistic tree pruning," *IEEE Trans. Signal Processing*, vol. 56, pp. 4867-4878, Oct. 2008.

[33] T. Cui, T. Ho and C. Tellambura, "Statistical pruning for near maximum likelihood detection of MIMO systems," *IEEE International Conference on Commun. (ICC)*, June 2007, pp. 5462-5467.

[34] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Processing*, vol. 53, pp. 2806-2818, Aug. 2005.

[35] H. V. Poor, *An Introduction to Signal Detection and Estimation, 2nd Edition*, Springer, 1994.

[36] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory.* Addison Wesley Longman, 1993.

[37] M. Tüchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using *a priori* information," *IEEE Trans. Signal Processing*, vol. 50, pp. 673-683, March 2002.

[38] D. Milliner, E. Zimmermann, J. R. Barry, G. Fettweis, "Channel state information based LLR clipping in list MIMO detection," *Proc. IEEE PIMRC*, Sept. 2008, pp. 15-18.

[39] H. V. Poor and S. Verdu, "Probability of error in MMSE multiuser detection," *IEEE Trans. Information Theory*, vol. 43, pp. 858-871, May 1997.

[40] P. Li, D. Paul, R. Narasimhan, and J. Cioffi, "On the distribution of SINR for the MMSE MIMO receiver and performance analysis," *IEEE Trans. Information Theory*, vol. 52, pp. 271-286, Jan. 2006.

[41] D. Guo, S. Verdu, and L. K. Rasmussen, "Asymptotic normality of linear multiuser receiver outputs," *IEEE Trans. Information Theory*, vol. 48, pp. 3080-3095, Dec. 2002.

[42] J. Proakis, *Digital communications: 4th edition.* McGraw-Hill, 2001.

[43] A. M. Tulino and S. Verdu, *Random matrix theory and wireless communications.* Foundations and Trends in Communications and Information Theory, 2004.

[44] D. Tse and P. Viswanath, *Fundamentals of wireless communication.* Cambridge University Press, 2005.

[45] S. T. Brink, "Convergence of iterative decoding," *Electron. Lett.*, vol. 35, pp. 806-808, May 1999.

[46] A. Edelman, "Eigenvalues and condition numbers of random matrices," Ph. D. thesis, Dept. Math., Massachusetts Inst. Technol., Cambridge, 1989.

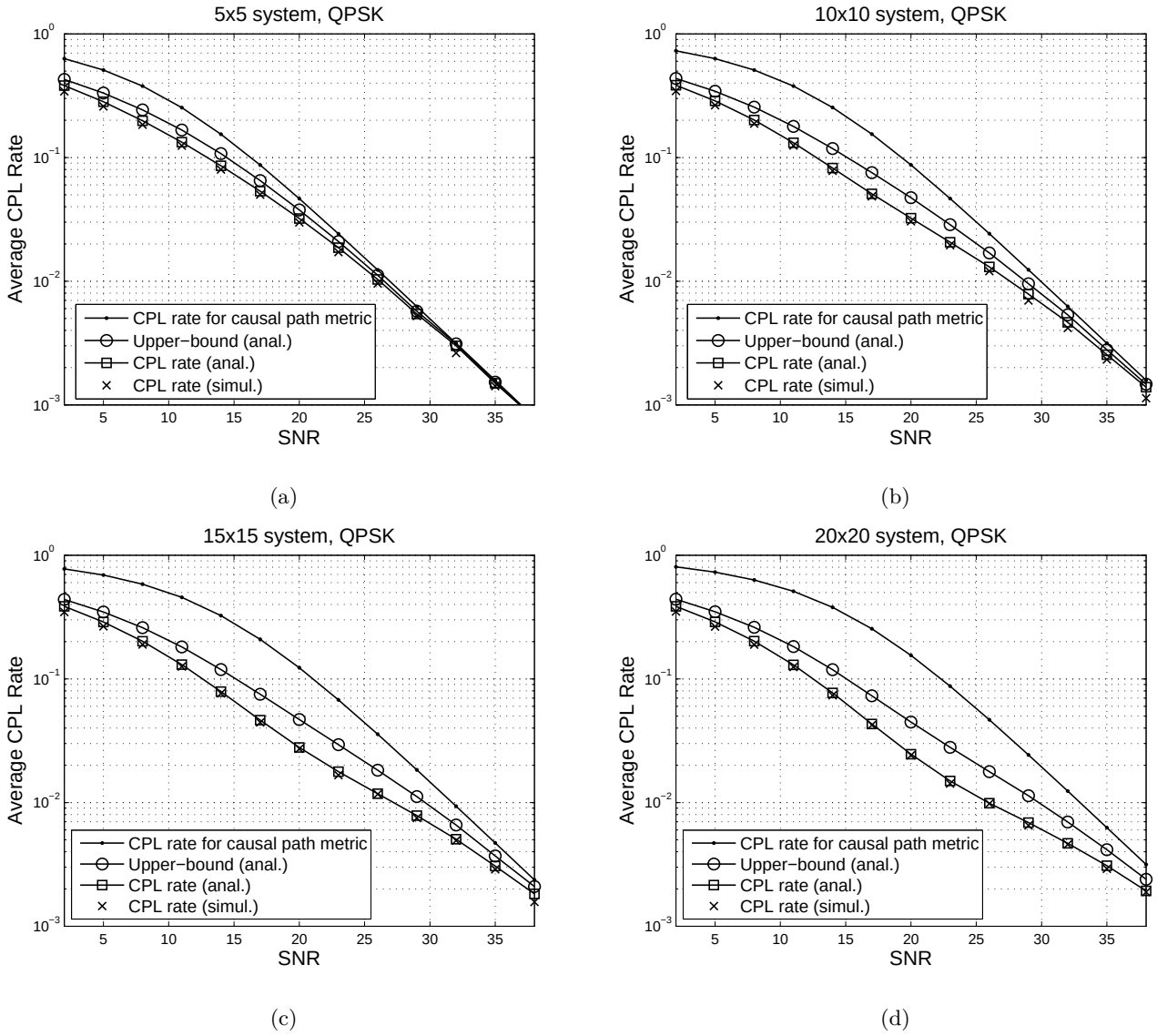


Fig. 2. Average CPL probability versus SNR for the (a) 5×5, (b) 10×10, (c) 15×15, and (d) 20×20 systems. QPSK uncoded transmission is considered. The curves for the "CPL rate (anal.)" are obtained by Monte-Carlo averaging of (53) over i.i.d. Gaussian channels.

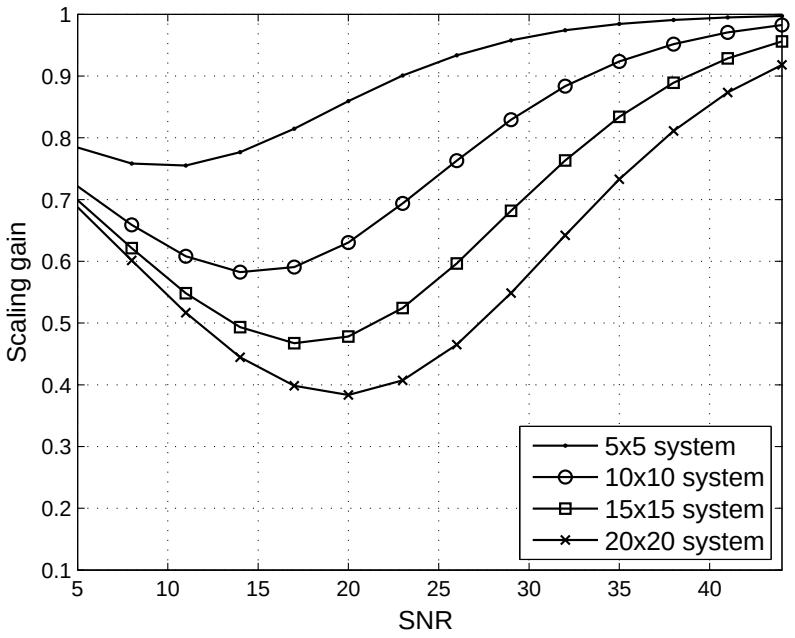
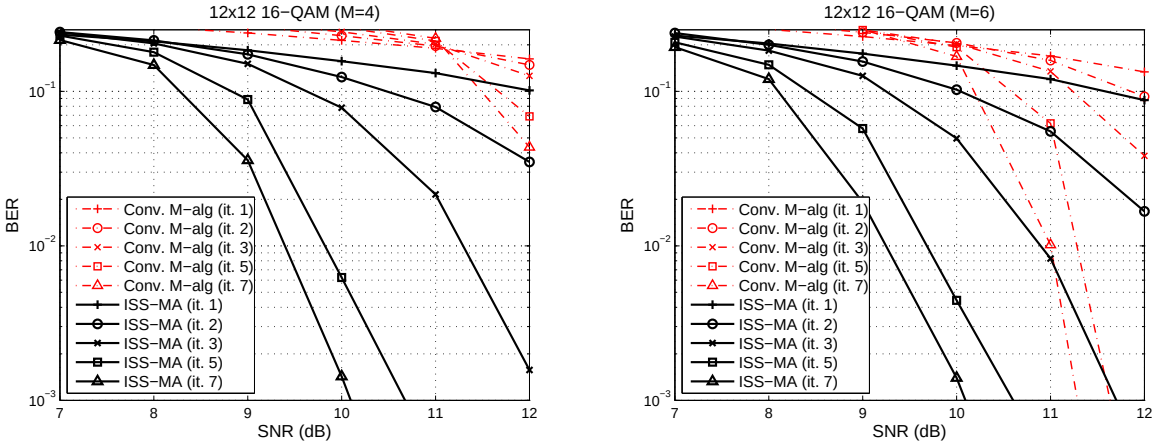
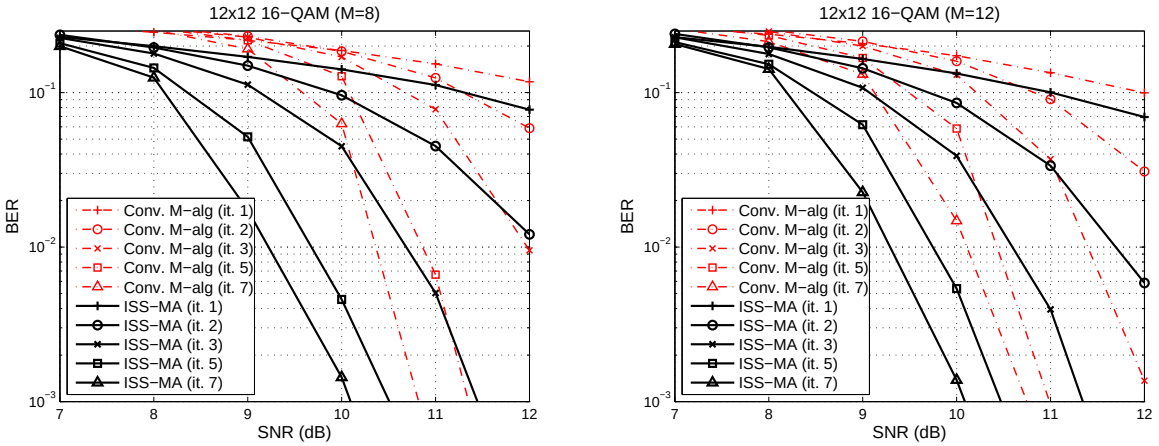


Fig. 3. Scaling gain versus SNR for different system sizes $N = 5, 10, 15$, and 20.



(a) (b)



(c) (d)

Fig. 4. Comparison between the causal path metric and LE-LA path metric for the 12×12 16-QAM system with (a) $M = 4$, (b) $M = 6$, (c) $M = 8$, and (d) $M = 12$.

TABLE II

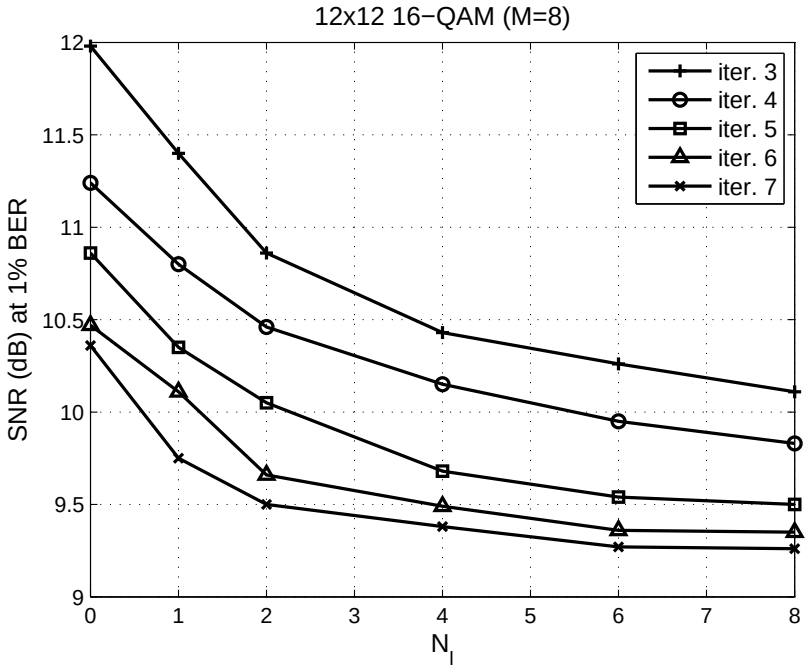
PERFORMANCE/COMPLEXITY OF 12×12 16-QAM SYSTEM FOR DIFFERENT M VALUES.

	LE-LA path metric		Conventional path metric	
	SNR (at BER = 1%)	# of multiplications	SNR (at BER = 1%)	# of multiplications
$M = 4$	**9.40** dB	**133.77**k	12.50 dB	120.23k
$M = 6$	9.25 dB	145.30k	11.00 dB	125.50k
$M = 8$	9.22 dB	157.53k	**10.36** dB	**132.00**k
$M = 12$	9.29 dB	183.12k	10.11 dB	145.98k

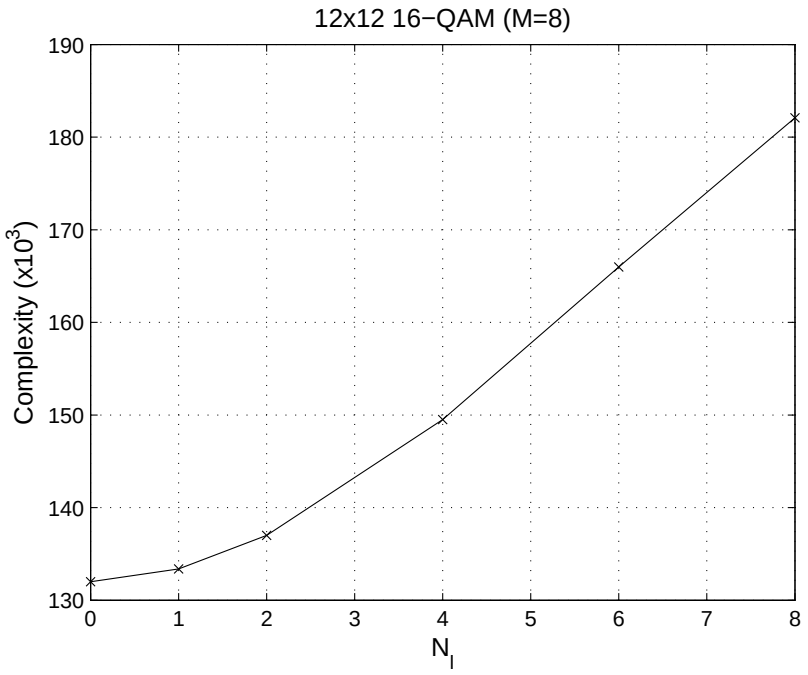
TABLE III

PERFORMANCE/COMPLEXITY FOR DIFFERENT PROBLEM SIZES. M IS SET TO 6.

	LE-LA path metric		Conventional path metric	
	SNR (at BER = 1%)	# of multiplications	SNR (at BER = 1%)	# of multiplications
6×6 16-QAM	8.80 dB	24.30k	9.29 dB	18.79k
8×8 16-QAM	8.97 dB	50.79k	9.59 dB	40.87k
10×10 16-QAM	9.22 dB	90.07k	10.39 dB	75.19k
12×12 16-QAM	9.25 dB	145.30k	10.11 dB	125.50k



(a)



(b)

Fig. 5. (a) Performance and (b) complexity of the ISS-MA versus N_l for 12×12 16-QAM system (M is set to 8).

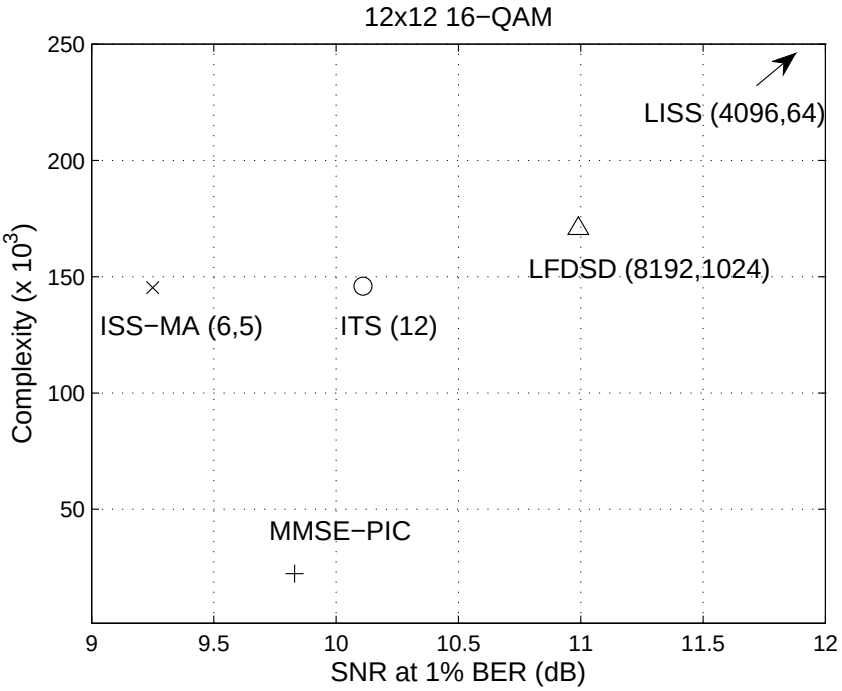


Fig. 6. Comparison of the several soft-input soft-output detectors. The numbers in the parenthesis represent the parameters of the detectors.