# A Low-complexity Near-ML Decoding Via Reduced Dimension Maximum Likelihood Search

Jun Won Choi, Byonghyo Shim, Nam Ik Cho, and Andrew C. Singer

***Abstract*— In this paper, we consider a low-complexity ML detection technique referred to as reduced dimension ML search (RD-MLS). RD-MLS directly refers the division of symbols into two groups viz. strong and weak group for searching over vector space with strong symbols instead of whole symbols. The inevitable performance loss, due to the exclusion of weak symbols, is compensated by 1) the list tree search which is an extended version of a single best searching algorithm called *sphere decoding* and 2) re-computation of the weak symbols for each strong symbols found at the list tree search. Furthermore, in order to lessen the computational burden in list tree search, we employ a tree pruning strategy that removes the unpromising sub-trees before they are being searched. Two pruning techniques, called list sphere decoding with probabilistic pruning (LSD-PP) and list stack algorithm with probabilistic pruning (LSA-PP), is proposed for this purpose. From the simulations performed on $M$-quadrature amplitude modulation (QAM) transmission through frequency non-selective multi-input-multi-output (MIMO) channels, we demonstrate that the RD-MLS shows near constant complexity over wide SNR range of interest ($10^{-1} \sim 10^{-4}$) while limiting the performance loss within a dB from the ML detection.**

***Index Terms*— Maximum likelihood (ML) decoding, Sphere decoding, Minimum mean square error (MMSE), Multiple input multiple output (MIMO), Stack algorithm, Dimension reduction, List tree search**

## I. Introduction

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

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

where $\mathbf{x}$ is the transmitted vector whose entries are chosen from a finite symbol alphabet, $\mathbf{y}$ and $\mathbf{w}$ are the received signal and noise vectors, respectively, and $\mathbf{H}$ is channel matrix. Multiple-input-multiple-output (MIMO) system is a typical example described by this model. In order to achieve the diversity and multiplexing gain promised by MIMO technologies [1], [2], a powerful MIMO detection scheme recovering the transmitted symbol with a minimal error is indispensable. In fact, due to the ability achieving minimum probability of error (MPE) performance and computational efficiency, an efficient maximum likelihood (ML) decoding algorithm referred to as *sphere decoding* (SD) has received much attention in recent years [3]–[7]. The key feature of the SD algorithm is the search space reduction using a sphere constraint. By restricting the search space within a sphere centered at the received vector, the reduction in number of lattice points that should be tested can be achieved. In spite of the substantial reduction in complexity, computational burden of the SD algorithm is still a major problem since the expected complexity is exponential with the problem size for a fixed signal-to-noise ratio (SNR) [11]. Considering a growing demand on high data rate services in future wireless systems, it is questionable to use the SD algorithm for a communication receiver supporting large dimension MIMO systems with high-order constellations.

There have been a number of approaches to reduce the complexity of SD algorithm including the Schnorr-Euchner enumeration [8]–[10], descending probabilistic ordering [14], increasing radius sphere decoder [19] and parallel competing branch algorithm [12]. Other approaches trading performance for complexity include the radius scheduling method [13], probabilistic tree pruning algorithm [15], [16], sequential Fano decoders [23], M-algorithm [20], K-algorithm [21] and semi-definite relaxation [22]. Refer [23] for the comprehensive and unified view on these algorithms.

In this paper, we introduce a near ML detection technique, referred to as *reduced dimension ML search* (RD-MLS) that provides significant savings in computational complexity, yet maintains a near ML performance. By reducing the dimension of search space from $n_t$ to $n_1$ ($n_1 \ll n_t$), the RD-MLS directly achieves the substantial reduction in size of search space from $M^{n_t}$ to $M^{n_1}$. Owing to the direct benefit on complexity, there have been a number of studies [24]–[29] on the combination of a linear processing and ML search. In a nutshell, whole symbol estimates in these methods are generated by the concatenation of symbol estimates obtained from the ML search and minimum mean square error (MMSE) estimation.

Our method is distinct from these approaches in two respects. First, instead of making a decision when the ML search is finished, we find multiple candidates by employing the list tree search (LTS). The LTS is applied once the linear processing, called dimension reduction operator, is finished. Dimension reduction operator performs the soft cancellation of the symbol not participating in the sphere search. In order to mitigate the performance loss due to the imperfection of the interference suppression, the LTS followed by the MMSE de-

J. W. Choi and A. C. Singer is with Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. B. Shim is with EECS Dept., Korea Univ., Seoul, Korea. N. I. Cho is with ECE Dept., Seoul National Univ., Seoul, Korea.

Contact info.: J. W. Choi and A. C. Singer [Address: 1308 W. Main St., CSL Building, 119, Urbana, IL, 61801, USA, E-mail: jwchoi,acsinger@illinois.edu, Phone: 1-217-377-7064, Fax: 1-217-244-1642], B. Shim [E-mail: bshim@korea.ac.kr, Phone: 82-2-3290-4842], N. I. Cho [E-mail: nicho@snu.ac.kr, Phone: 82-2-880-1810]

Please send your correspondence regarding this paper to Prof. Shim (email: bshim@korea.ac.kr).

A part of this paper was presented in IEEE Sensor Array and Multichannel Signal Processing (SAM) Workshop, Germany, 2008. Refer [34].

cision feedback (MMSE-DF) estimation is employed. Specifically, for each candidate generated by the LTS, the rest of symbols is estimated via MMSE-DF. Among concatenated symbols, the final solution is chosen as a minimizer of the $L_2$-norm based cost function. Second, in order to reduce the complexity increase and lessen the sensitivity on channel conditions, an additional complexity control method based on tree pruning of the LTS is introduced. Due to the selection on multiple lattice points, the choice of a search radius in the LTS depends more on the matrix structure and channel condition and thereby easily eat out the complexity gain of the dimension reduction. By prunning out the unpromising path before visit, a detection complexity comparable that of conventional SD algorithms can be achieved.

Through the asymptotic performance analysis, we show that the diversity gain of the RD-MLS is a function of the reduced dimension $n_1$ that lies between the full-dimensional ML ($n_r$) and MMSE ($n_r - n_t + 1$). In particular, from the Monte-Carlo simulations on the analysis, we show that the diversity equation of the RD-MLS approximates $n_r + n_1 - n_t$. The eventual performance loss over the exact ML detection due to the diversity gain diference is compensated by the LTS gain. In fact, from the simulation study on MIMO communication systems, we demonstrate that the complexity of RD-MLS is close to that of linear detection schemes (e.g., V-BLAST detector [17]) yet achieving the performance within a dB from the ML detector.

The rest of the paper is organized as follows. After describing the system model in Section II, we briefly review the SD algorithm and its computational complexity in Section III. In Section IV, we present the RD-MLS algorithm and analyze its performance. The simulation results are provided in Section V and we conclude in Section VI.

We briefly summarize the notations to be used for the rest of this paper. Uppercase and lowercase letters written in boldface are used for matrix and vector notations, respectively. The superscripts $(\cdot)^T$ and $(\cdot)^H$ denote a transpose and a conjugate transpose, respectively. $\|\cdot\|^2$ denotes an $L_2$-norm square of a vector. $\mathcal{X}_k^2$ denotes a Chi-square distribution with $k$ degree of freedom. $F_\chi(\cdot\,;k)$ and $F_\chi^{-1}(\cdot\,;k)$ are the cumulative density function (CDF) and the inverse CDF of the $\chi^2$-random variable with degree of freedom (DOF) $k$, respectively. $Q(x)$ denotes a $Q$-function defined as $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}}\exp\left(-x^2/2\right)$. $R_{\mathbf{xy}}$ denotes $E\left[\mathbf{x}\mathbf{y}^H\right] - E[\mathbf{x}]E[\mathbf{y}^H]$.

## II. Sphere decoding algorithm

### A. System Model

In the system model described in the previous section, if all the matrices and vectors are real and further $\mathbf{w}$ is the Gaussian noise vector ($w_i \sim N(0, \sigma_w^2)$), then the ML detection problem becomes

$$\mathbf{x}_{\mathrm{ml}} = \arg\min_{\mathbf{x}\in\mathcal{F}^{n_t}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \tag{2}$$

where $\mathbf{H}$ is the $n_r \times n_t$ channel matrix and $\mathbf{x}$ is the $n_t \times 1$ vector whose component is an element of the $M$-quadrature amplitude modulation (QAM) set $\mathcal{F}$ defined as

$$\mathcal{F} = \left\{ x_r + jx_i \middle|\ x_r, x_i \in \left\{ \frac{-\sqrt{M}+1}{\lambda}, \frac{-\sqrt{M}+3}{\lambda}, \cdots, \frac{\sqrt{M}-3}{\lambda}, \frac{\sqrt{M}-1}{\lambda} \right\} \right\}. \tag{3}$$

where $\lambda$ is chosen to satisfy the normalization condition $E\left[\mathbf{x}\mathbf{x}^H\right] = \mathbf{I}_{n_t}$. For example, $\lambda = \sqrt{10}$ for 16-QAM and $\lambda = \sqrt{42}$ for 64-QAM modulation, respectively.

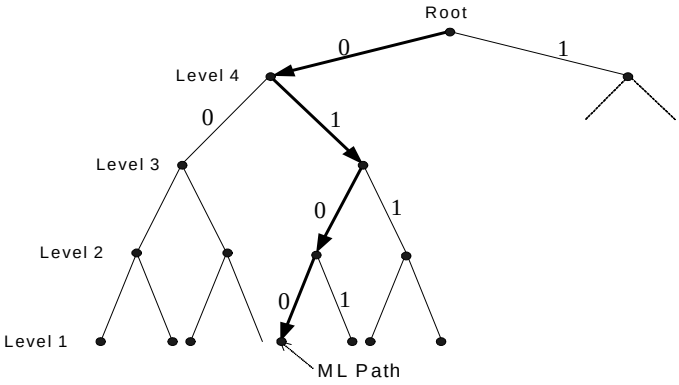


Fig. 1. Illustration of tree search for $2 \times 2$ MIMO with QPSK.

### B. SD Algorithm

The SD algorithm searches lattice points inside a hypersphere with the radius $\sqrt{B}$, centered at the received vector $\mathbf{y}$. The points inside the sphere satisfy

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \le B. \tag{4}$$

For a systematic tree search, QR decomposition of the channel matrix $\mathbf{H}$ is employed

$$\mathbf{H} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix} \tag{5}$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are the $n_r \times n_t$ and $n_r \times (n_r - n_t)$ matrix and $\mathbf{R}$ is the upper triangular matrix whose diagonal elements are positive. Since a norm operation is invariant to an orthogonal transform, the sphere constraint of (4) can be rewritten as

$$\|\mathbf{y}' - \mathbf{R}\mathbf{x}\|^2 = \sum_{i=1}^{n_t}\left| y_i' - \sum_{l=i}^{n_t} r_{i,l}x_l \right|^2 \le B' \tag{6}$$

where $\mathbf{y}' = \mathbf{Q}_1^T\mathbf{y}$, $B' = B - \|\mathbf{Q}_2^T\mathbf{y}\|^2$, and $\mathbf{y}' = \left[y_1', \cdots, y_{n_t}'\right]^T$. Emphasizing that each term in the left is a function of $x_i, \cdots x_{n_t}$, (6) becomes

$$B_1(x_1^{n_t}) + B_2(x_2^{n_t}) + \cdots + B_{n_t}(x_{n_t}^{n_t}) \le B' \tag{7}$$

where $x_i^{n_t}$ denotes the node associated with $x_i, \cdots x_{n_t}$, and $B_i(x_i^{n_t}) = \left|y_i' - \sum_{l=i}^{n_t} r_{i,l}x_l\right|^2$. Since the search starts from the bottom layer of the tree and moves upward, by defining the path metric $d_k\left(\mathbf{x}_k^{n_t}\right) = \sum_{i=k}^{n_t} B_i(x_i^{n_t})$, the update equation becomes

$$d\left(\mathbf{x}_k^{n_t}\right) = d\left(\mathbf{x}_{k+1}^{n_t}\right) + B_k(x_k^{n_t}). \tag{8}$$

Due to the fact that $d(\mathbf{x}_l^{n_t}) \geq d(\mathbf{x}_m^{n_t})$ for $l \leq m$, if a path metric violates the sphere condition, all leaf nodes of the subtree also violate the condition. Thus, the node $\mathbf{x}_k^{n_t}$ and all the subtrees are removed from the tree to keep the points satisfying the sphere condition.

In searching the nodes in a branch, Pohst enumeration [3] and SE enumeration [8] based ordering mechanisms are popularly used. In Pohst enumeration, natural spanning from the minimal to the maximal value is used between the interval,

$$X_k^{\min} \leq x_k \leq X_k^{\max} \tag{9}$$

where

$$X_k^{\max} = \left\lfloor \frac{1}{r_{k,k}} \left( y_k' - \xi_k + \sqrt{B' + d\left(\mathbf{x}_{k+1}^{n_t}\right)} \right) \right\rfloor, \tag{10}$$

$$X_k^{\min} = \left\lceil \frac{1}{r_{k,k}} \left( y_k' - \xi_k + \sqrt{B' - d\left(\mathbf{x}_{k+1}^{n_t}\right)} \right) \right\rceil, \tag{11}$$

where $\xi_k = \sum_{i=k+1}^{n_t} r_{k,i} x_i$. In contrast, SE enumeration spans the admissible point of $x_k$ in a zig-zag order from the mid-point $x_{k,\mathrm{mid}} = \left\lceil \frac{1}{r_{k,k}} (y_k' - \xi_k) \right\rfloor$. That is, the SE enumeration spans $x_{k,\mathrm{mid}}, x_{k,\mathrm{mid}}+1, x_{k,\mathrm{mid}}-1, x_{k,\mathrm{mid}}+2, \cdots$, when $\bar{y}_k' - \xi_k - r_{k,k} x_{k,\mathrm{mid}} \geq 0$, and $x_{k,\mathrm{mid}}, x_{k,\mathrm{mid}}-1, x_{k,\mathrm{mid}}+1, x_{k,\mathrm{mid}}-2, \cdots$, otherwise. By traversing the tree with this branch ordering mechanism, all lattice points inside the sphere are visited and a final lattice point having the minimum path metric becomes the ML point.

### C. Complexity of SD algorithm

Since neither the best nor the worst complexity can be a representative measure for evaluating complexity, expected complexity has been widely employed for assessing computational performance of the SD algorithm [6], [11]. Assuming that each node shares the same amount of operations, the lower bound on the expected number of nodes visited [11] is

$$E[N] \geq \frac{M^{\eta n_t} - 1}{\sqrt{M} - 1} \tag{12}$$

where $N$ is the number of the visited nodes and $\eta$ is the complexity exponent given by

$$\eta = \frac{1}{2} \left( 1 + \frac{4(M-1)}{3\lambda^2} \mathrm{SNR} \right)^{-1}. \tag{13}$$

Since $E[N]$ exponentially increases with $n_t$, the dimension reduction might be a powerful strategy for complexity reduction. However, direct reduction of the dimension will result in the performance loss so that a deliberate mechanism mitigating the performance loss is required. In the following section, we describe the detail of the proposed dimension reduction algorithm.

## III. Reduced dimension ML search (RD-MLS) algorithm

The structure of the RD-MLS system is shown in Fig. 2, where the preprocessing block reduces the system dimension by suppressing the interference, i.e., the contribution of symbols not participating in the search stage. The closest lattice point search is being performed in the reduced dimension


Fig. 2. System of RD-MLS technique.

system. Two key aspects of the search operation are 1) list tree search (LTS) and 2) probabilistic tree pruning. Since the ML solution of the reduced-dimension system is not necessarily equal to the ML solution of the original system, we find multiple candidates in the LTS and then choose the final output in the postprocessing stage. In order to lessen the complexity increase of the LTS, a probabilistic tree pruning method that snips the unlikely branch of the tree is introduced.

### A. Dimension Reduced ML problem

As a first step for the dimension reduction, the symbol vector is divided into two groups called strong symbol group $\mathbf{x}_1 \in \mathcal{F}^{n_1}$ and weak symbol group $\mathbf{x}_2 \in \mathcal{F}^{n_2}$ ($n_2 = n_t - n_1$). Then the ML solution can be expressed as

$$\begin{aligned} \mathbf{x}_{\mathrm{ml}} &= \arg \min_{\mathbf{x} \in \mathcal{F}^{n_t}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \\ &= \arg \min_{\mathbf{x}_1 \in \mathcal{F}^{n_1}, \mathbf{x}_2 \in \mathcal{F}^{n_2}} \|\mathbf{y} - \mathbf{H}_2\mathbf{x}_2 - \mathbf{H}_1\mathbf{x}_1\|^2 \end{aligned} \tag{14}$$

where $\mathbf{H}_1$ and $\mathbf{H}_2$ are the sub-matrices constructed by the $n_1$ and $n_2$ columns of $\mathbf{H}$, respectively. Denoting $\mathbf{x}_{\mathrm{ml}}^T = [\mathbf{x}_{1,\mathrm{ml}}^T \quad \mathbf{x}_{2,\mathrm{ml}}^T]^T$, $\mathbf{x}_{1,\mathrm{ml}}$ can be expressed as the two-stage processing given by

$$\mathbf{x}_{1,\mathrm{ml}} = \arg \min_{\mathbf{x}_1 \in \mathcal{F}^{n_1}} D(\mathbf{y} \,|\, \mathbf{x}_1) \tag{15}$$

$$D(\mathbf{y} \,|\, \mathbf{x}_1) = \min_{\mathbf{x}_2 \in \mathcal{F}^{n_2}} \|\mathbf{y} - \mathbf{H}_2\mathbf{x}_2 - \mathbf{H}_1\mathbf{x}_1\|^2. \tag{16}$$

Obviously, plugging (16) into (15) will return to (14) and no dimension reduction can be achieved. Indeed, in order to restrict the search space within $\mathbf{x}_1$, $\mathbf{x}_2$ needs to be estimated in a cost effective manner. In fact, reformulating an estimate of $D(\mathbf{y} \,|\, \mathbf{x}_1)$ without relying on an exhaustive search of $\mathbf{x}_2$ is the crux of the proposed approach. Employing the linear estimate $\hat{\mathbf{x}}_2$ of $\mathbf{x_2}$ for a given $\mathbf{x}_1$, (16) becomes

$$\widetilde{D}(\mathbf{y} \,|\, \mathbf{x}_1) = \|\mathbf{y} - \mathbf{H}_2\hat{\mathbf{x}}_2 - \mathbf{H}_1\mathbf{x}_1\|^2. \tag{17}$$

In estimating $\hat{\mathbf{x}}_2$, a linear minimum mean square error (LMMSE) method is being used. In a condition that $\mathbf{x}_1$ is given, the corresponding MMSE estimate of $\mathbf{x}_2$ is

$$\begin{aligned} \hat{\mathbf{x}}_2 &= R_{\mathbf{x}_2\mathbf{y}'} R_{\mathbf{y}'\mathbf{y}'}^{-1} (\mathbf{y} - E[\mathbf{y}]) \\ &= \mathbf{H}_2^H \left( \mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2 \mathbf{I} \right)^{-1} (\mathbf{y} - \mathbf{H}_1\mathbf{x}_1). \end{aligned} \tag{18}$$

Note that depending on the input of $\widetilde{D}(\mathbf{y} \,|\, \mathbf{x}_1)$ or $\mathbf{x}_1$, the different $\hat{\mathbf{x}}_2$ is used. Using (17) and (18), an approximate ML solution $\tilde{\mathbf{x}}_{1,\mathrm{ml}}$ becomes

$$\begin{aligned} \tilde{\mathbf{x}}_{1,\mathrm{ml}} &= \arg \min_{\mathbf{x}_1 \in \mathcal{F}^{n_1}} \widetilde{D}(\mathbf{y} \,|\, \mathbf{x}_1) \\ &= \arg \min_{\mathbf{x}_1 \in \mathcal{F}^{n_1}} \Big\| \mathbf{y} - \mathbf{H}_2\mathbf{H}_2^H \left( \mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2 \mathbf{I} \right)^{-1} \\ &\qquad (\mathbf{y} - \mathbf{H}_1\mathbf{x}_1) - \mathbf{H}_1\mathbf{x}_1 \Big\|^2. \end{aligned} \tag{19}$$

Denoting $\mathbf{Z} = \mathbf{I} - \mathbf{H}_2\mathbf{H}_2^H \left(\mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2\mathbf{I}\right)^{-1}$, one can easily show that

$$\tilde{\mathbf{x}}_{1,\mathrm{ml}} = \arg\min_{\mathbf{x}_1 \in \mathcal{F}^{n_1}} \|\mathbf{Z}\mathbf{y} - \mathbf{Z}\mathbf{H}_1\mathbf{x}_1\|^2. \quad (20)$$

Further, by denoting $\mathbf{z} = \mathbf{Z}\mathbf{y}$ and $\mathbf{G} = \mathbf{Z}\mathbf{H}_1$, we have

$$\tilde{\mathbf{x}}_{1,\mathrm{ml}} = \arg\min_{\mathbf{x}_1 \in \mathcal{F}^{n_1}} \|\mathbf{z} - \mathbf{G}\mathbf{x}_1\|^2. \quad (21)$$

Therefore, the solution can be searched over the $n_1$ dimensional search space.

### B. Dimension Reduction Operator

In essence, the preprocessing operation consists of 1) application of the simple linear operator $\mathbf{Z}$ and 2) the ML search over this transformed system. By applying $\mathbf{Z}$, the linear system becomes

$$\mathbf{z} = \mathbf{G}\mathbf{x}_1 + \mathbf{Z}\mathbf{r} \quad (22)$$

where $\mathbf{r} = (\mathbf{w} + \mathbf{H}_2\mathbf{x}_2)$. Since $\mathbf{H}_2\mathbf{x}_2$ is an interference in detecting $\mathbf{x}_1$, the contribution of this term should be minimized by the preprocessing. In fact, it is clear from definition that

$$\mathbf{Z} = \sigma_w^2 \left(\mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2\mathbf{I}\right)^{-1} \quad (23)$$

and one can show that (23) equals to

$$\mathbf{Z} = \arg\min_{\mathbf{Z}'} E\left[\|\mathbf{w} - \mathbf{Z}'\mathbf{r}\|^2\right] = R_{\mathbf{wr}}R_{\mathbf{rr}}^{-1} \quad (24)$$

which is nothing but the LMMSE operator of $\mathbf{r}$ to estimate $\mathbf{w}$, i.e., $\hat{\mathbf{w}} = \mathbf{Z}\mathbf{r}$. Hence, (22) can be written as

$$\mathbf{z} = \mathbf{G}\mathbf{x}_1 + \hat{\mathbf{w}}, \quad (25)$$

where $\hat{\mathbf{w}} = \mathbf{w} + \mathbf{e}$ and $\mathbf{e}$ is the MMSE estimation error. Since $\mathbf{e}$ is colored, $\tilde{\mathbf{x}}_{1,\mathrm{ml}}$ is not strictly the ML solution of the system (22). However, by the proper choice of $\mathbf{H}_2$ and $\mathbf{x}_2$, the power of the colored noise component can be minimized and $\tilde{\mathbf{x}}_{1,\mathrm{ml}}$ becomes an approximate ML solution. A proper choice of $\mathbf{H}_1$ and $\mathbf{H}_2$ would be the one maximizing the receiver SNR of (25)

$$(\mathbf{H}_1, \mathbf{H}_2) = \arg\max_{\mathbf{H}_1', \mathbf{H}_2'} \mathrm{SNR}(\mathbf{H}_1', \mathbf{H}_2') \quad (26)$$

where

$$\begin{aligned}\mathrm{SNR}\left(\mathbf{H}_1', \mathbf{H}_2'\right) &= \frac{E(\,||\mathbf{G}\mathbf{x}_1||^2\,)}{E(\,||\hat{\mathbf{w}}||^2\,)} \\ &= \frac{\mathrm{tr}\left(\mathbf{Z}\mathbf{H}_1'\left(\mathbf{H}_1'\right)^H\mathbf{Z}^H\right)}{\mathrm{tr}\left(\sigma_w^4\left(\mathbf{H}_2'\left(\mathbf{H}_2'\right)^H + \sigma_w^2\mathbf{I}\right)^{-1}\right)}. \end{aligned} \quad (27)$$

For finding an optimal partition, $\frac{n_t!}{n_1!(n_t-n_1)!}$ choices should be examined, which is clearly too burdensome for a large $n_t$. Thus, simple schemes such as the V-BLAST symbol ordering [17] or probabilistic symbol ordering [14] might be preferred as a practical option.

### C. List Tree Search (LTS)

As mentioned, due to the reduced dimension search, $\widetilde{\mathbf{x}}_{1,\mathrm{ml}}$ is not guaranteed to be equal to $\mathbf{x}_{1,\mathrm{ml}}$ and the performance loss is unavoidable. In order to mitigate the loss, the LTS searching multiple candidates of $\mathbf{x}_1$ is employed. The list SD (LSD) algorithm proposed in [30] finds $N$ best lattice points by choosing a search radius large enough to keep more than $N$ points in the sphere. Note, contrary to the SD algorithm where the radius is updated dynamically for each candidate found, the LSD algorithm employs fixed radius until it finds $N$ best points. The radius is updated only when the list is filled and a new candidate substituting the existing one is found. In many cases, therefore, excessive number of lattice points might be visited, which can easily dilute the benefit of the dimension reduction. To maintain the effectiveness of the LTS and pursue the computational efficiency at the same time, we introduce a statistical pruning in the LTS.

*1) List Sphere Decoding with Probabilistic Pruning (LSD-PP):* After the preprocessing operation described in Section II.B, the sphere constraint of (21) becomes

$$d\left(\mathbf{x}_1^{n_1}\right) = \|\mathbf{z}' - \mathbf{R}\mathbf{x}_1\|^2 \le B' \quad (28)$$

where $\mathbf{G} = \mathbf{Q}\mathbf{R}$ and $\mathbf{z}' = \mathbf{Q}'\mathbf{z}$. Although $d\left(\mathbf{x}_1^{n_1}\right)$ should be used during the whole search, due to the causality of the operation, only $d\left(\mathbf{x}_k^{n_1}\right)$ are being used. In fact, the sphere constraint is very loose in the top level and gets tight as the search moves to the bottom of the tree. Hence, pruning of top paths with a large path metric is not possible as long as they satisfy the loose sphere constraint.

The proposed LSD-PP approach tightens the sphere constraint by adding the contribution of random noise into the path metric. Towards this end, the full path $\mathbf{x}_1^{n_1}$ is divided into the two parts: the already visited part (from the root to the current node) $\mathbf{x}_k^{n_1}$ and unvisited part $\mathbf{x}_1^{k-1}$. The corresponding path metric would be

$$d\left(\mathbf{x}_1^{n_1}\right) = d\left(\mathbf{x}_1^{k-1}\right) + d\left(\mathbf{x}_k^{n_1}\right) \quad (29)$$

$$= \sum_{i=1}^{k-1}\left(z_i' - \sum_{j=i}^{n_1} r_{i,j}x_j\right)^2 + \sum_{i=k}^{n_1'}\left(z_i' - \sum_{j=i}^{n_1} r_{i,j}x_j\right)^2. \quad (30)$$

To estimate the path metric $d\left(\mathbf{x}_1^{k-1}\right)$ of the unvisited part, we modify the sphere constraint by using the expected path metric

$$E\left[d\left(\mathbf{x}_1^{n_1}\right)\middle|\mathbf{x}_k^{n_1}\right] \le B'. \quad (31)$$

Since $E\left[d\left(\mathbf{x}_1^{n_1}\right)\middle|\mathbf{x}_k^{n_1}\right] = E\left[d_k\left(\mathbf{x}_1^{k-1}\right)\right] + d\left(\mathbf{x}_k^{n_1}\right)$, (31) becomes

$$E\left[d_k\left(\mathbf{x}_1^{k-1}\right)\right] + d\left(\mathbf{x}_k^{n_1}\right) \le B'. \quad (32)$$

Noting that $z_i' = \sum_{j=i}^{n_1} r_{i,j}\widetilde{x}_j + w_i'$ with $w_i'$ is zero-mean

Gaussian, $E\left[d_k\left(\mathbf{x}_1^{k-1}\right)\right]$ becomes

$$E\left[d\left(\mathbf{x}_1^{k-1}\right)\right] = \sum_{i=1}^{k-1} E\left[\left(w_i' - \sum_{j=1}^{n_1} r_{i,j}\left(x_j - \widetilde{x}_j\right)\right)^2\right] \quad (33)$$

$$\geq \sum_{i=1}^{k-1} E\left[\left(w_i'\right)^2\right] \quad (34)$$

$$= (k-1)\frac{\sigma_w^2}{2}. \quad (35)$$

Using (32) and (35), the modified necessary condition becomes

$$(k-1)\frac{\sigma_w^2}{2} + d_k\left(\mathbf{x}_k^{n_1}\right) \leq B'. \quad (36)$$

Note that the first term in the left-hand side of (36) represents the contribution of the unvisited nodes ($k-1$-th to the bottom layers) by which the sphere constraint gets tightened. Although using $E\left[d_k\left(\mathbf{x}_1^{k-1}\right)\right]$ might be a proper option for SD, such choice will be too harsh for LSD since $N$ best lattice points needs to be selected instead of single best. In order to relax the pruning condition, we add a control parameter $\mu_k$

$$\mu_k(k-1)\frac{\sigma_w^2}{2} + d_k\left(\mathbf{x}_k^{n_1}\right) \leq B' \quad (37)$$

where $\mu_k \leq 1$ scales the noise contribution on each layer. The condition of $\mu_k$ to prune the unlikely paths can be described as

$$Pr\left(\sum_{i=1}^{k-1}\left(w_i'\right)^2 \leq \mu_k(k-1)\frac{\sigma_w^2}{2}\right) = P_{\text{prun}}, \quad (38)$$

where $P_{\text{prun}}$ is the pruning probability that controls the pruning capability. If $w_k'$ is Gaussian then $\sum_{i=1}^{k-1}\left(w_i'\right)^2$ follows $\mathcal{X}_{k-1}^2$ distribution and (38) becomes

$$F_\chi\left(\mu_k(k-1); k-1\right) = P_{\text{prun}}. \quad (39)$$

By taking the inverse, we directly have $\mu_k = \frac{F_\chi^{-1}(P_{\text{prun}};k-1)}{(k-1)}$ and (37) becomes

$$d\left(\mathbf{x}_k^{n_1}\right) \leq B' - F_\chi^{-1}\left(P_{\text{prun}}; k-1\right)\frac{\sigma_w^2}{2}. \quad (40)$$

Since $F_\chi^{-1}(\cdot\,; k)$ is an increasing function of $k$, the effective search radius (right hand side in (40)) increases as the search moves to the bottom of the tree. As an extreme case, when it reaches to the bottom of the tree ($k = 1$), the radius returns to the initial radius $B'$.

Using this modified pruning constraint, the LTS is performed and $N$ closest points are selected as a candidate list. If the number of points found is less than $N$, the search is started over after updating the radius of (40) into $B'' = JB', (J > 1)$. In this case, the parameter $J$ effectively controls the increase of the number of lattice point for next trials. The LSD-PP algorithm is summarized in Appendix I.

*2) List Stack Algorithm with Probabilistic Pruning (LSA-PP):* As an alternative approach of the LSD-PP, we introduce a list stack algorithm with probabilistic pruning (LSA-PP). The stack algorithm [18], [19] is the best-first tree search method that explores a tree by extending a node with the minimum cost metric. Node information and cost metric are stored in the stack and updated whenever a new node is extended. Since the first path arriving at the final level becomes the ML solution, the original stack algorithm stops the search when a full path is visited once. The cost metric associated with a node $\mathbf{x}_k^{n_1}$ is given by

$$a_k\left(\mathbf{x}_k^{n_1}\right) = \min_{x_{k-1}\in H_k} d_{k-1}\left(\mathbf{x}_{k-1}^{n_1}\right), \quad (41)$$

where $H_k$ is the set of child nodes in $\mathbf{x}_k^{n_1}$ not generated yet. Thus, $a_k\left(\mathbf{x}_k^{n_1}\right)$ corresponds to the path metric of the best child node in $\mathbf{x}_k^{n_1}$.

In [31], the stack algorithm has been extended to LTS by exploiting the information stored in the stack and soft symbol information. The proposed list stack algorithm differs from that of [31] in that it finds the $K$ number of closest paths. The key feature of the list stack algorithm is that the stack continues node extension until it reaches the leaf level $K$ times instead of stopping at the first arrival. However, due to the increase in the numbers of path competing, large number of back-tracking operation occurs. Since the back-tracking operation is caused by the comparison of the cost metrics in different tree levels, the stack algorithm requires a bias term proportional to the tree depth to the cost metric [18]. The PTP principle, which is used to obtain tighter pruning bound in the LSD-PP, provides a way to determine the bias term. The modified cost metric including the PTP-based bias term is given by

$$a_k\left(\mathbf{x}_k^{n_1}\right) = \min_{x_{k-1}\in H_k} d_{k-1}\left(\mathbf{x}_{k-1}^{n_1}\right) + \mu_{k-1}(k-2)\frac{\sigma_w^2}{2} \quad (42)$$

where, similarly to the LST-PP, $\mu_k = \frac{F^{-1}(P_{\text{prun}};k-1)}{(k-1)}$ is determined from a design parameter $P_{\text{prun}}$.

It is worth mentioning that the search is in fact terminated without finding the fixed number of lattice points by the *fast stopping criterion*. Specifically, if the cost metric of a current path $\mathbf{x}_k^{n_1}$ is much larger than that of closest path $a_1$, i.e., $a_k\left(\mathbf{x}_k^{n_1}\right) > ma_1$ for $m > 1$, then the search is stopped instead of searching $K_{\max}$ lattice points and only $K$ ($< K_{\max}$) points are returned. Hence, only lattice points with dominantly small metric are kept in the list, causing less back-tracking operation. By introducing the stopping parameter $m$, the size of the candidate list can vary depending on a channel and noise condition, thereby improving the search complexity considerably. The LSA-PP algorithm is summarized in Appendix II.

### *D. Postprocessing*

The postprocessing operates in two steps. First, for each strong symbols obtained by the LTS, we generate the corresponding weak symbols by the MMSE-DF estimation followed by the slicing. Next, full-dimension symbols are generated by concatenating the sliced weak symbols and the corresponding strong symbols. Among multiple candidates in the list, the best one minimizing the Euclidean distance metric is chosen as the final output.

Since MMSE-DF estimation in this setup is well-known [32], [33], we skip the detail and present the resulting equations briefly. We let $\hat{\mathbf{x}}_1^i$ be the $i$-th element in the candidate list and $\hat{\mathbf{x}}_2^i$ be the corresponding MMSE-DF estimate of $\mathbf{x}_2$

given $\hat{\mathbf{x}}_1^i$. Assume that the columns of $\mathbf{H}_2$ and entries of $\mathbf{x}_2$ are arranged based on the V-BLAST detection ordering. Then the estimate $\hat{\mathbf{x}}_2^i$ is obtained as

$$\check{\mathbf{x}}_2^i = \mathbf{F}\left(\mathbf{y} - \mathbf{H}_1\hat{\mathbf{x}}_1^i\right) - \mathbf{B}\hat{\mathbf{x}}_2^i \tag{43}$$

$$\hat{\mathbf{x}}_2^i = \text{slicer}\left(\check{\mathbf{x}}_2^i\right) \tag{44}$$

where "slicer" does a function of mapping an input to the nearest constellation point. The feedforward filter $\mathbf{F}$ and feedback filter $\mathbf{B}$ minimize the MSE between $\mathbf{x}_2$ and $\hat{\mathbf{x}}_2^i$. Due to the causality in decision feedback operation, $\mathbf{B}$ should have a lower triangular shape. Further, let $\mathbf{h}_{2,k}$ be the $k$-th column of $\mathbf{H}_2$ given by $[\mathbf{h}_{2,1}, \cdots, \mathbf{h}_{2,k}]$ and $\mathbf{b}_k$ be the $k$-th column of $\mathbf{B}$ given by $[b_{k,1}, \cdots, b_{k,k-1}]^T$. Assuming $\hat{\mathbf{x}}_1^i$ equals $\mathbf{x}_1$, then $\mathbf{b}_k$ minimizing $E\left[\|\mathbf{x}_2 - \hat{\mathbf{x}}_2^i\|^2\right]$ is given by

$$\mathbf{b}_k = \mathbf{D}_{k-1}\mathbf{H}_{2,k-1}^H\mathbf{\Sigma}\,\mathbf{h}_{2,k} \tag{45}$$

where $\mathbf{D}_{k-1} = \left(\mathbf{I} - \mathbf{H}_{2,k-1}^H\mathbf{\Sigma}\mathbf{H}_{2,k-1},\right)^{-1}$ and $\mathbf{\Sigma} = \left(\mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2\mathbf{I}\right)^{-1}$. The optimal $\mathbf{F}$ becomes

$$\mathbf{F} = (\mathbf{B} + \mathbf{I})\,\mathbf{H}_2^H\mathbf{\Sigma}. \tag{46}$$

Once $\hat{\mathbf{x}}_2^i$ is obtained by (43)-(44), $\hat{\mathbf{x}}_1^i$ and $\hat{\mathbf{x}}_2^i$ are concatenated for the final search as

$$\mathcal{L} = \left\{\hat{\mathbf{x}}_{\text{ext}}^1, \cdots, \hat{\mathbf{x}}_{\text{ext}}^K\right\} = \left\{\begin{bmatrix}\hat{\mathbf{x}}_1^1\\ \hat{\mathbf{x}}_2^1\end{bmatrix}, \cdots, \begin{bmatrix}\hat{\mathbf{x}}_1^K\\ \hat{\mathbf{x}}_2^K\end{bmatrix}\right\} \tag{47}$$

and the element of $\mathcal{L}$ minimizing the cost function becomes the final output of the RD-MLS

$$\widetilde{\mathbf{x}}^{\text{ml}} = \arg\min_{\hat{\mathbf{x}}_{\text{ext}}\in\mathcal{L}} \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_{\text{ext}}\|^2. \tag{48}$$

## IV. Discussion

To achieve the goal of the RD-MLS, it is important to choose an appropriate value of $n_1$ providing the best tradeoff between performance and complexity. In this section, we provide the upper bound analysis of conditional error probability (CEP) that $\mathbf{x}_A$ is transmitted but not detected by the RD-MLS. The diversity gain obtained from the CEP analysis can be a useful reference for choosing the reduced dimension parameter $n_1$.

### A. Performance analysis

Recalling that the final output chosen from $\mathcal{L}$ is denoted as $\widetilde{\mathbf{x}}^{\text{ml}}$, $P_{\text{cer}}(\mathbf{x}_A)$ is defined as

$$P_{\text{cer}}(\mathbf{x}_A) = Pr_A\left(\widetilde{\mathbf{x}}^{\text{ml}} \neq \mathbf{x}_A\right) \tag{49}$$

$$= Pr_A\left(\widetilde{\mathbf{x}}^{\text{ml}} \neq \mathbf{x}_A \middle| \mathbf{x}_A \in \mathcal{L}\right) Pr_A\left(\mathbf{x}_A \in \mathcal{L}\right) + Pr_A\left(\widetilde{\mathbf{x}}^{\text{ml}} \neq \mathbf{x}_A \middle| \mathbf{x}_A \notin \mathcal{L}\right) Pr_A\left(\mathbf{x}_A \notin \mathcal{L}\right) \tag{50}$$

where $Pr_A(\cdot)$ refers the probability in condition that $\mathbf{x}_A$ is sent. The ML detection error probability $P_e^{\text{ML}}$ implies that there exists at least one symbol vector whose distance metric is smaller than $\|\mathbf{y} - \mathbf{H}\mathbf{x}_A\|$, so it is clear that $Pr_A\left(\widetilde{\mathbf{x}}^{\text{ml}} \neq \mathbf{x}_A \middle| \mathbf{x}_A \in \mathcal{L}\right) \leq P_e^{\text{ML}}$. It is also obvious that $Pr_A\left(\widetilde{\mathbf{x}}^{\text{ml}} \neq \mathbf{x}_A \middle| \mathbf{x}_A \notin \mathcal{L}\right) = 1$ and hence (50) becomes

$$\begin{aligned} P_{\text{cer}}(\mathbf{x}_A) &\leq P_e^{\text{ML}}\left(1 - Pr_A\left(\mathbf{x}_A \notin \mathcal{L}\right)\right) + Pr_A\left(\mathbf{x}_A \notin \mathcal{L}\right) \\ &\leq P_e^{\text{ML}} + Pr_A\left(\mathbf{x}_A \notin \mathcal{L}\right). \end{aligned} \tag{51}$$

Note, since $Pr_A\left(\mathbf{x}_A \notin \mathcal{L}\right) \ll 1$ in a moderate or high SNR regime, the inequality in (51) maintains the tightness. Evidently, the second term in the right-hand side of (51) explains the extra loss due to the sub-optimality of the RD-MLS.

Following the symbol classification method of the RD-MLS, $\mathbf{x}_A$ is divided into $\mathbf{x}_{A1}$ and $\mathbf{x}_{A2}$ and thus (51) becomes

$$\begin{aligned} P_{\text{cer}}(\mathbf{x}_A) &\leq P_e^{\text{ML}} + Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right) \\ &\quad + Pr_A\left(\mathbf{x}_A \notin \mathcal{L} | \mathbf{x}_{A1} \in \mathcal{L}_1\right) Pr_A\left(\mathbf{x}_{A1} \in \mathcal{L}_1\right) \quad (52) \\ &\leq P_e^{\text{ML}} + Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right) + Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right). \quad (53) \end{aligned}$$

where $Pr_A\left(\mathbf{x}_A \notin \mathcal{L} | \mathbf{x}_{A1} \in \mathcal{L}_1\right) = Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right)$ and $Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right) \ll 1$ in a moderate and high SNR regime. The second and third term in the right-hand side in (53) are the probability that the candidate list found by the LTS does not contain $\mathbf{x}_{A1}$ and the probability that the MMSE-DF makes a decision error on the condition that $\mathbf{x}_{A1} \in \mathcal{L}$, respectively.

In LTS with the stopping criterion parameter $m$, we can show that $Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right)$ is upper bounded by

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right) \leq E_{\mathbf{H}_2}\left[\prod_{i=1}^{n_r} \frac{\frac{\lambda_1}{\lambda_i^2}}{\frac{\lambda_1}{\lambda_i^2} + \frac{m}{\lambda^2}}\right], \tag{54}$$

where $\lambda$ is a normalization factor of the QAM modulation and $\lambda_1 \leq \cdots \leq \lambda_{n_r}$ are the eigenvalues of $\left(\mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2\mathbf{I}\right)^{-1}$. See Appendix III for the proof of (54). For an intuitive explanation on (54), it is worth considering a high SNR regime where $\sigma_w^2 \to 0$. Since the quantities $\frac{\lambda_1}{\lambda_{n_r-n_2+1}^2}, \cdots, \frac{\lambda_1}{\lambda_{n_r}^2}$ are arbitrarily large, we have

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right) \leq E_{\mathbf{H}_2}\left[\prod_{i=1}^{n_r-n_2} \frac{\frac{\lambda_1}{\lambda_i^2}}{\frac{\lambda_1}{\lambda_i^2} + \frac{m}{\lambda^2}}\right] \tag{55}$$

$$\leq \left(\frac{\lambda^2}{m}\right)^{n_r-n_2} E_{\mathbf{H}_2}\left[\prod_{i=1}^{n_r-n_2} \frac{\lambda_1}{\lambda_i^2}\right]. \tag{56}$$

The fact that the upper bound is a function of the list stopping parameter $m$ is matching with our intuition because, when $m$ increases, the number of candidates in the list also increases, and the effect of additional loss decreases. Since $m$ is multiplied into this minimum squared distance $4/\lambda^2$ of QAM, an additional performance gain is obtained over the single point tree search.

In general, it is hard to find the closed-form expression on (54) due to the expectation on $\mathbf{H}_2$. Hence, for a quantitative result, we need to evaluate the expectation over the fading realizations of $\mathbf{H}_2$. Since the expectation is well approximated via Monte-Carlo (MC) simulation, we use MC method in evaluating the diversity gain ($= \lim_{SNR\to\infty} -\frac{\Delta BER}{\Delta SNR}$). In Table I, the diversity gains for MIMO and system size variations are

TABLE I

DIVERSITY GAIN OBTAINED FROM (54) FOR SEVERAL $n_1$ VALUES USING 16-QAM MODULATION AND V-BLAST SYMBOL ORDERING.

| | $n_1=1$ | $n_1=2$ | $n_1=3$ | $n_1=4$ | $n_1=5$ | $n_1=6$ | $n_1=7$ |
|---|---|---|---|---|---|---|---|
| $n_r=6, n_t=6$ | 0.97 | 1.97 | 2.96 | 3.94 | 4.94 | NA | NA |
| $n_r=8, n_t=8$ | 0.93 | 1.95 | 2.91 | 3.93 | 4.91 | 5.90 | 6.92 |
| $n_r=8, n_t=6$ | 2.96 | 3.95 | 4.94 | 5.92 | 6.91 | NA | NA |

summarized, from which we get $n_r+n_1-n_t$ as an approximate equation.

We next present the upper bound on $Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right)$, which is the error probability due to the MMSE-DF. Let $\widetilde{\mathbf{H}}_2 = \left[\mathbf{h}_{i_1}, \cdots, \mathbf{h}_{i_{n_2}}\right]$ be the reordering of the columns of $\mathbf{H}_2$ and let $\widetilde{\mathbf{H}}_{2,[i:j]}$ be the submatrix from the $i$ to $j$-th columns of $\widetilde{\mathbf{H}}_2$. Then $Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right)$ is upper bounded by

$$Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right) \leq 1 - E_{\mathbf{H}_2}\left[\prod_{k=1}^{n_2}\left(1-4\left(1-\frac{1}{\sqrt{M}}\right)Q\left(\sqrt{\frac{2}{\sigma_{u_k}^2\lambda^2}}\right)\right)\right], \tag{57}$$

where $\sigma_{u_k}^2 = 1 - \mathbf{h}_{i_k}\left(\widetilde{\mathbf{H}}_{2,[k:n_2]}\widetilde{\mathbf{H}}_{2,[k:n_2]}^H + \sigma_w^2\mathbf{I}\right)^{-1}\mathbf{h}_{i_k}^H$. See Appendix IV. Performing the MC simulations for realizations of $\mathbf{H}_2$ in (57), we observe that the diversity gain is close to $n_r-n_2+1 = n_r+n_1-n_t+1$ which is equal to the diversity gain of unordered DF detection [36].

Using (54) and (57), we obtain the upper bound of $P_{\mathrm{cer}}(\mathbf{x}_A)$ as

$$P_{\mathrm{cer}}(\mathbf{x}_A) \leq P_{\mathrm{er}}^{\mathrm{ub}} = P_e^{\mathrm{ML}} + \underbrace{E_{\mathbf{H}_2}\left[\prod_{i=1}^{n_r}\frac{\frac{\lambda_1}{\lambda_i^2}}{\frac{\lambda_1}{\lambda_i^2}+\frac{m}{\lambda^2}}\right]}_{P_e^1} + \underbrace{1 - E_{\mathbf{H}_2}\left[\prod_{k=1}^{n_2}\left(1-4\left(1-\frac{1}{\sqrt{M}}\right)Q\left(\sqrt{\frac{2}{\sigma_{u_k}^2\lambda^2}}\right)\right)\right]}_{P_e^2} \tag{58}$$

Recalling $SNR^{-m_{\min}} \sim \sum_i SNR^{-m_i}$ for large $SNR$, the diversity gain of $P_{\mathrm{cer}}(\mathbf{x}_A)$ would be dominated by $n_r+n_1-n_t$, which is the minimum of the diversity gains of $P_e^{\mathrm{ML}}$, $P_e^1$, and $P_e^2$, or, $\min\left(n_r, n_r+n_1-n_t\ \ , n_r+n_1-n_t+1\right)$. Hence, we expect that the performance loss over the ML detection, in particular for high SNR regime, is unavoidable. Nonetheless, due to the gain from the stopping parameter in the LTS, performance loss of the RD-MLS is mitigated in the wide operational regime, as shown in the simulation results in Section V.

### B. Comments on complexity

The overall complexity of the RD-MLS algorithm consists of those for 1) preprocessing, 2) tree search, and 3) post-processing operations. Since our focus is on the complexity comparison between the RD-MLS and standard ML tree search, we do not count the QR decomposition and symbol classification complexities, which are common for both. In our analysis, the complexity is evaluated by the number of floating-point operations (FLOPS) counted per channel use. The number of FLOPS $N$ needed for the tree search operation is expressed as

$$N = \sum_{k=1}^{n_t}\sum_{\mathbf{x}_k^{n_t}} N_k I\left(\mathbf{x}_k^{n_t}\right), \tag{59}$$

where $I\left(\mathbf{x}_k^{n_t}\right)$ outputs 1 if $\mathbf{x}_k^{n_t}$ is visited, and 0 otherwise. $N_k$ is the number of FLOPS per node at the tree level $k$. The total number of FLOPS depends on the number of nodes visited. TABLE II tabulates the number of FLOPS in the preprocessing and postprocessing steps and per node computations of the LSD-PP and LSA-PP. Employing this and the number of node visited information, averaged number of FLOPS of the RD-MLS system can be measured.

## V. SIMULATIONS

In this section, we observe the complexity and performance of the proposed RD-MLS over full-dimensional SD and V-BLAST detector. The simulation setup is based on $M$-QAM transmission over MIMO systems in quasi-static Rayleigh fading channel where the elements of $\mathbf{H}$ are modeled by independent Gaussian random variables. As a measure for the performance and complexity, bit error rate (BER) and the average number of FLOPS per channel use are used. Total $10^6$ information bits are generated for each point in the simulations. The following algorithms are compared;

1) Full dimensional SD (FD-SD) algorithm: *Algorithm II* [5], which guarantees an exact ML performance.
2) RD-MLS I: RD-MLS algorithm with LSD-PP. We set parameter $J$ to 1.5 and $P_{\mathrm{prun}}$ to 0.4. The initial radius is set to infinity and only two maximal search failure is allowed.
3) RD-MLS II: RD-MLS algorithm with LSA-PP. We fixed $P_{\mathrm{prun}}$ to 0.6.
4) V-BLAST detector [17].

The convention of $(n_r, n_t, n_1)$ is used to represent a configuration of the RD-MLS where the design parameter $n_1$ is chosen based on the analysis in Section IV.

First, we consider 16-QAM transmission for $(6,6,4)$, $(8,8,6)$ and $(10,10,7)$ systems. Fig. 3 shows the average number of FLOPS and BER for each detection algorithm in terms of SNR. We set $m=4$, $K=3$ and $K_{\max}=4$ for the $(6,6,4)$ and $(8,8,6)$ systems and $m=6$, $K=5$ and $K_{\max}=6$ for the $(10,10,7)$ system. These parameters are chosen empirically based on the simulation results. For almost entire SNR range of interest (except the range from 22 to 24 dB in $(6,6,4)$ system for the RD-MLS I), the RD-MLS algorithms exhibit lower complexity than the FD-SD. In particular, at 19

TABLE II

COMPLEXITY OF RD-MLS ALGORITHM.

| Block | # of real additions | # of real multiplications |
|---|---|---|
| Preprocessing step | $4n_r^2 - 2n_r$ | $4n_r^2$ |
| Postprocessing step | $K\left(12n_t - 4n_1 + 4\right)n_r - K\left(2n_t - 2n_1 + 2\right)$ | $K\left(12n_t - 4n_1 + 4\right)n_r$ |
| $N_k$ for LSD-PP alg. | $2n_t - k + 5$ | $2n_t - k + 3$ |
| $N_k$ for LSA-PP alg. | $2n_t - k + 7$ | $2n_t - k + 5$ |

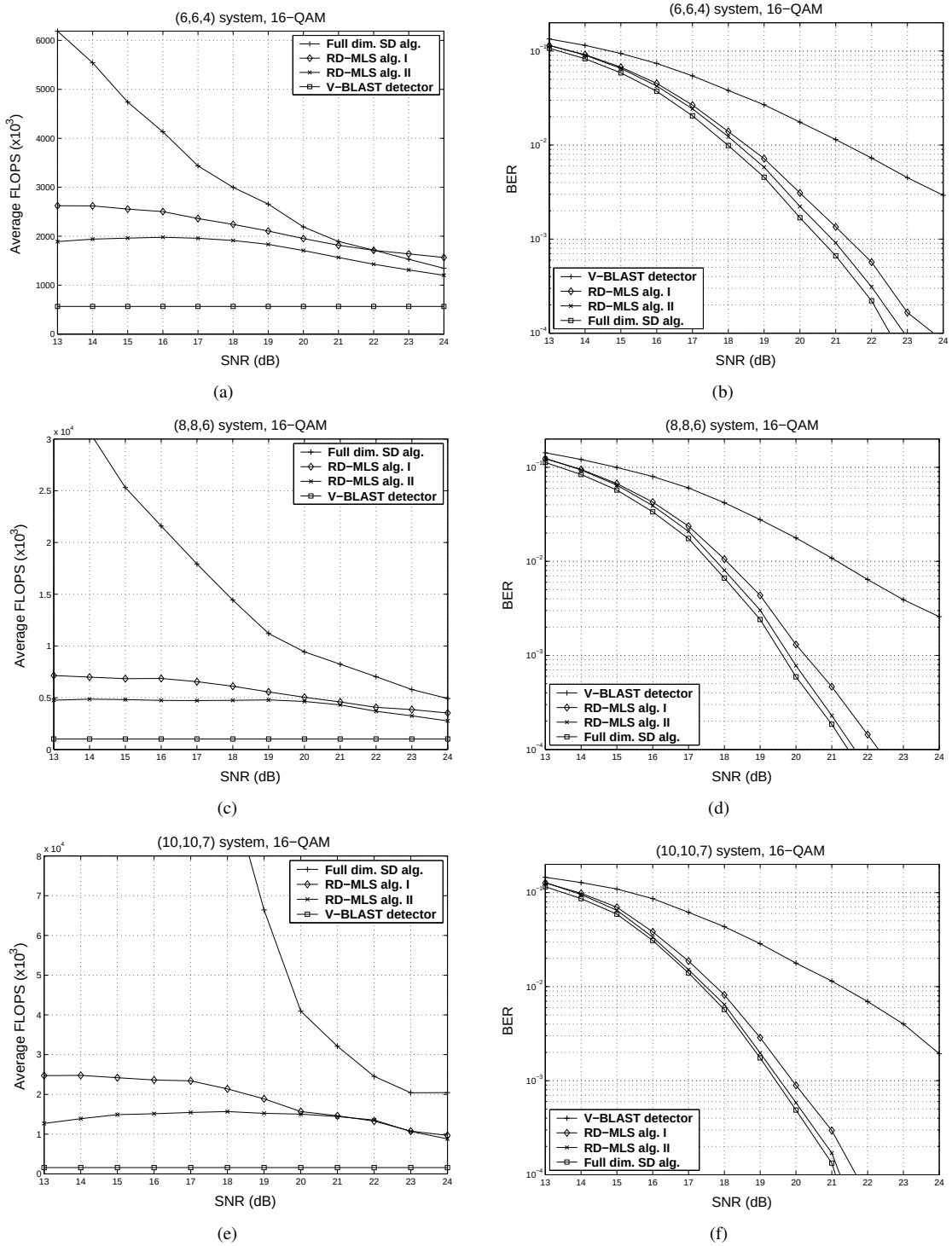


Fig. 3. Plots of average FLOPS and BER of (a) and (b) $(6, 6, 4)$ system, (c) and (d) $(8, 8, 6)$ system, and (e) and (f) $(10, 10, 7)$ system. 16-QAM case is considered.

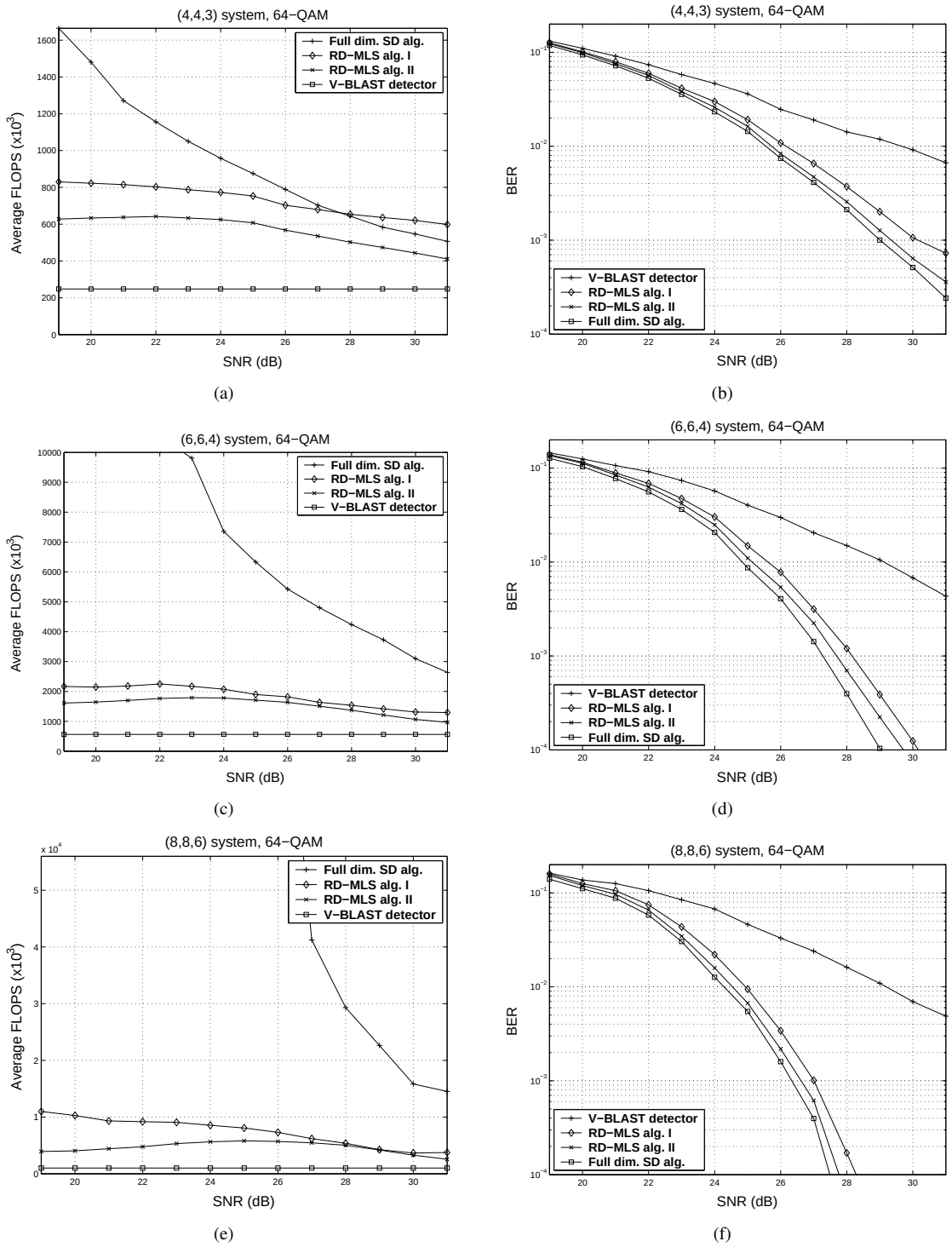


Fig. 4. Plots of average FLOPS and BER of (a) and (b) $(4, 4, 3)$ system, (c) and (d) $(6, 6, 4)$ system, and (e) and (f) $(8, 8, 6)$ system. 64-QAM case is considered.

dB of SNR, the RD-MLS II achieves 27% to 75% complexity reduction while the RD-MLS I achieves slightly less reduction than those of the RD-MLS II. On the contrary, the performance gap from the exact ML performance is less than 1 dB for all cases. As the system size $n_t$ increases, the achieved complexity reduction gets increasing significantly, showing the potential of the RD-MLS for large size systems. It is noteworthy that the RD-MLS II maintains a low level of complexity even for low SNR range. The reason we observe this result is that the fast stopping criterion of the RD-MLS II reduces the size of candidate list adaptively in response to different channel realizations and noise variance. Somewhat surprisingly, the RD-MLS II shows slightly decreasing tendency in complexity for decreasing SNR between 13 dB and 16 dB.

In Fig. 4, the performance and complexity curves for the 64-QAM transmission are provided. We include the results of

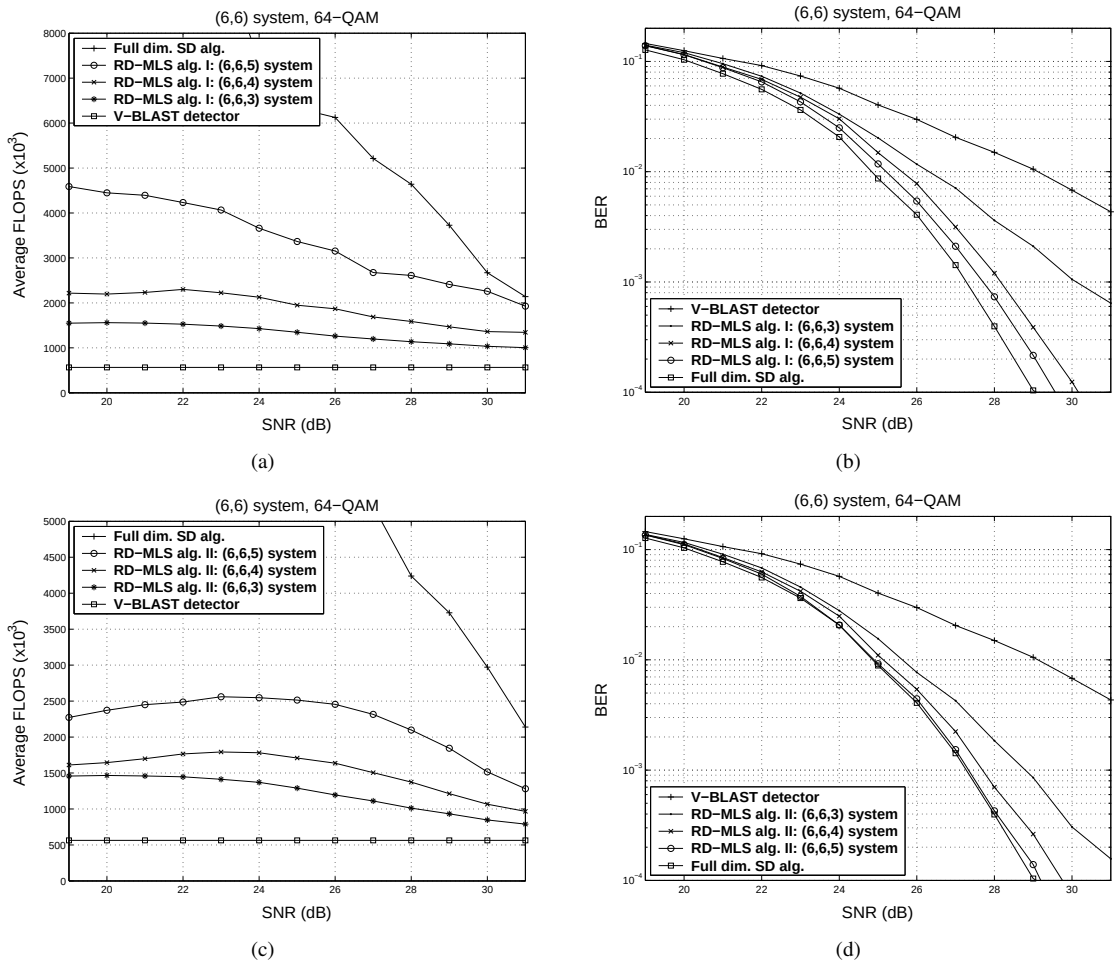


Fig. 5. The complexity and performance of (a) and (b) RD-MLS I and (c) and (d) RD-MLS II for $n_1 = 3, 4$ and 5. The $(6, 6)$ system and 64-QAM case are considered.

the $(4, 4, 3)$, $(6, 6, 4)$ and $(8, 8, 6)$ systems. We set $m = 4$, $K = 3$ and $K_{\max} = 5$ for the $(4, 4, 3)$ system, $m = 5$, $K = 4$ and $K_{\max} = 5$ for the $(6, 6, 4)$ system, and $m = 6$, $K = 5$ and $K_{\max} = 7$ for the $(8, 8, 6)$ system. The RD-MLS algorithms yield lower complexity than the FD-SD over almost entire SNR range of interest. At 24 dB of SNR, the RD-MLS II achieves the $24\%$ complexity reduction for the $(4, 4, 3)$ system and $73\%$ for the $(6, 6, 4)$ system. For the $(8, 8, 6)$ system, the complexity reduction achieved by the RD-MLS improves dramatically. The complexity of the RD-MLS I was slightly worse than the RD-MLS II but still better than the FD-SD. In addition, for the same size of system, the performance gain for the 64-QAM case is larger than that for the 16-QAM case. These results match our intuition that a search complexity would be scaled by the factor of $M^{n_t - n_1}$ by the reduction of search space dimension. Notice that the performance loss of the RD-MLD is within 1 dB over the FD-SD.

Fig. 5 shows how complexity and performance are traded off through $n_1$. Only 64-QAM case is considered and the performance and complexity results of the $(6, 6)$ system are provided for $n_1 = 3, 4$ and 5. For all cases, we fix the parameters as $m = 4$, $K = 4$ and $K_{\max} = 4$. Fig. 5 (a) and (b) shows the results of the RD-MLS I and Fig. 5 (c) and (d) those of the RD-MLS II. As $n_1$ decreases, more complexity reduction is achieved while performance loss increases. When $n_1 = 3$, the RD-MLS can reduce the complexity substantially while more than 1 dB performance loss should be sacrificed for the RD-MLS II and 2 dB for the RD-MLS I. As $n_1$ increases, the performance gap from the exact ML performance decreases and becomes very close when $n_1 = 5$. This result explains that the RD-MLS fills the gap between a high complexity ML detector and a linear sub-optimal detector. Owing to the stopping criterion used in the RD-MLS II, it yields better performance-complexity trade-off than the RD-MLS I. This gain of the RD-MLS II is conspicuous in high SNR region where the algorithm does not have to use a large size of candidate list for obtaining a near-ML performance.

## VI. Conclusions

A low-complexity near-ML decoding technique referred to as RD-MLS is presented in this paper. On top of the dimension reduction which directly saves the complexity, two main ideas mitigating the performance loss have been addressed. Firstly, the detection on the reduced dimension system is modified from the ML search to the LTS. Therefore, instead of detecting one best strong symbol, multiple candidates are found in the

LTS stage. Secondly, for each strong symbols, corresponding weak symbols are estimated via MMSE-DF algorithm. Among concatenated symbols list, the final output is chosen as a minimizer of the $L_2$-norm based cost function. We have found, from the CEP analysis, that the diversity gain of the RD-MLS lies between that of ML and MMSE but with additional performance gain offered by the LTS. Therefore, in spite of the eventual difference in performance (due to the diversity gain difference), we could observe from the simulations that the BER performance of the RD-MLS is comparable to the FD-ML search (original SD) for the wide range of practical interests.

There are several interesting directions worth pursuing. The curves obtained from simulation shows that the RD-MLS complexity curve is close to constant, i.e., insensitive to SNR variations. Although it is quite promising from the practical perspective, we are not clear if this behavior can be extended into very high dimensional systems (such as $n_t > 50$). Henceforth, more rigorous and analytic guidance might be needed to uncover a comprehensive trend. As a related topic, one might also consider partitioning into several groups instead of two. Furthermore, one might improve the performance (achieve real close-to-ML performance) by iteratively applying the LTS and MMSE-DF operation.

## APPENDIX I
## SUMMARY OF LSD-PP ALGORITHM

In describing the LSD-PP algorithm, following definitions are used.

- $\Phi(k)$: path metric of a current node at the $(n_1 - k + 1)$ tree level,
- $\hat{x}_k$: current value of $x_k$.
- $n_f$: the number of search failures. A search failure implies that the number of candidate points found is less than $K$.
- XLIST: memory to store the lattice points found.
- $b(k) = F^{-1}\left(P_{\text{prun}}; k\right) \frac{\sigma_w^2}{2}$.

If the LSD-PP encounters a search failure, it increases the radius by $JB'n_f$ and restarts the search. The LSD-PP algorithm is summarized;

**STEP 1** : (initialization)
$\Phi(n_t) = 0$, $\xi_{n_t} = 0$, $\{\hat{x}_1, \cdots, \hat{x}_{n_t}\} = \{0, \cdots, 0\}$, $n_f = 0$ and $k = n_t$.
**STEP 2** : compute $\hat{x}_k = \lfloor (y'_k - \xi_k)/r_{k,k} \rceil$ and $\delta_k = \text{sign}\left(y'_k - \xi_k - r_{k,k}x_k\right)$, then go to STEP 4.
**STEP 3** : (SE enumeration)
compute $\hat{x}_k = \hat{x}_k + \delta_k$ and $\delta_k = -\delta_k - \text{sign}\left(\delta_k\right)$, then go to STEP 4.
**STEP 4** : (main routine)
compute $c = |y'_k - \xi_k - r_{k,k}\hat{x}_k|^2$.
If $\Phi\left(k\right) + c > B'\left(1 + Jn_f\right) - b(k-1)$,
  if $k = n'_1$
    if the size of XLIST is less than $K$
      $n_f = n_f + 1$ and go to STEP 1.
    else
      go to STEP 5.
    end
  else set $k = k + 1$, then go to STEP 3.
  end
else
  if $\hat{x}_k$ is outside the signal set boundary, go to STEP 3.
  elseif $k > 1$,
    set $\Phi(k-1) = \Phi(k) + c$, $\xi_{k-1} = \sum_{i=k}^{n_t} r_{k-1,i}\hat{x}_i$,
    $k = k - 1$, then go to STEP 2.
  else
    if $\Phi(1) + c < B'$
      set $B' = \Phi(1) + c$ (update the search radius)
    end
    XLIST $\leftarrow \{\hat{x}_1, \cdots, \hat{x}_{n_t}\}$.
    $k = k + 1$, then go to STEP 3.
  end
end
**STEP 5.** (finish)
In the XLIST, choose the $K$ elements with smallest path metric and output them. ■

## APPENDIX II
## SUMMARY OF LSA-PP ALGORITHM

The following definitions are used in LSA-PP algorithm.

- STACK: memory storing the generated nodes.
- XLIST: memory storing the candidate points found.
- $\check{x}_k^{n_1}$: top node of the stack after sorting.
- $\text{bcn}\left(x_k^{n_1}\right)$: best child node of $x_k^{n_1}$ not generated yet.

The LSA-PP algorithm is summarized;
**STEP 1** : (initialization)
STACK $\leftarrow$ root,
**STEP 2** : (main routine)
Let $\check{x}_k^{n_1}$ be the top node.
if $k = 2$
  if the stoping criterion is met
    go to STEP 4.
  end
  XLIST $\leftarrow \left(\text{bcn}\left(\check{x}_k^{n_1}\right), \check{x}_k^{n_1}\right)$ and update $a_k\left(\check{x}_k^{n_1}\right)$.
  remove $\check{x}_k^{n_1}$ from STACK.
else
  STACK $\leftarrow \left(\text{bcn}\left(\check{x}_k^{n_1}\right), \check{x}_k^{n_1}\right)$ and update $a_k\left(\check{x}_k^{n_1}\right)$.
  if all child nodes of $\check{x}_k^{n_1}$ have been generated,
    remove $\check{x}_k^{n_1}$ from STACK.
  else
    update $\text{bcn}\left(\check{x}_k^{n_1}\right)$, and $a_k\left(\check{x}_k^{n_1}\right)$.
  end
end
**STEP 3** : (stack sorting)
set the node with the minimum cost metric as a top node.
go to STEP 2.
**STEP 4** : (finish)
output XLIST. ■

## APPENDIX III
## PROOF OF (54)

Referring (25), the transformed observation is expressed as

$$\mathbf{z} = \mathbf{G}\mathbf{x}_{A1} + \hat{\mathbf{w}}, \tag{60}$$

To make the analysis tractable, we assume that $\hat{\mathbf{w}}$ is a proper Gaussian [37] with zero mean and the covariance matrix $\mathbf{\Phi} = E[(\hat{\mathbf{w}})(\hat{\mathbf{w}})^H] = \sigma_w^4 \left(\mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2\mathbf{I}\right)^{-1}$. This assumption is not strictly true, but it has been shown that the residual error of the MMSE estimation can be well approximated by Gaussian model [35]. In addition, let $\hat{\mathbf{x}}_1^{\max}$ and $\hat{\mathbf{x}}_1^{\min}$ be the elements of $\mathcal{L}_1$ that correspond to the maximizer and minimizer of the cost function $J(\mathbf{x}) = \|\mathbf{z} - \mathbf{G}\mathbf{x}\|^2$, respectively. In this setup, $Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right)$ is expressed as

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right) = Pr_A\left(\mathbf{x}_{A1} \notin \left\{\hat{\mathbf{x}}_1^1 \cdots \hat{\mathbf{x}}_1^K\right\} \middle| \mathbf{H}\right) \tag{61}$$

$$= Pr_A\left(\|\mathbf{z} - \mathbf{G}\mathbf{x}_{A1}\|^2 > \|\mathbf{z} - \mathbf{G}\hat{\mathbf{x}}_1^{\max}\|^2 \middle| \mathbf{H}\right) \tag{62}$$

$$= Pr\left(\|\mathbf{v}\|^2 > \|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right) + \mathbf{v}\|^2 \middle| \mathbf{H}\right) \tag{63}$$

$$= Pr\left(\Re\left\{\left(\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\right)^H \mathbf{v}\right\} < -\frac{1}{2}\left\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\right\|^2 \middle| \mathbf{H}\right) \tag{64}$$

An affine transform of a proper Gaussian variable is also proper Gaussian [37] so that we can assume that $\left(\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\right)^H \mathbf{v}$ is proper Gaussian. The variance of its real part is given by

$$\frac{1}{2}\left(\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\right)^H \mathbf{\Phi}\left(\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\right) = \frac{1}{2}\left\|\mathbf{Q}\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\right\|^2 \tag{65}$$

where $\mathbf{Q}$ is a square root of $\mathbf{\Phi}$, i.e., $\mathbf{\Phi} = \mathbf{Q}^H\mathbf{Q}$. Recall that $Pr(z < -\beta) = Pr(z > \beta) = Q(\frac{\beta}{\sigma})$ for Gaussian RV $z$ with variance $\sigma$. Then $Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right)$ is expressed as

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right) = Q\left(\frac{\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\|^2}{\sqrt{2}\left\|\mathbf{Q}\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\right\|}\right) \tag{66}$$

By using the definition of a matrix norm, $\|\mathbf{A}\|_2 = \max_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\| / \|\mathbf{x}\|$, we have

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right) \le Q\left(\sqrt{\frac{\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\|^2}{2\|\mathbf{\Phi}\|_2}}\right), \tag{67}$$

where we use $\|\mathbf{\Phi}\|_2 = \|\mathbf{Q}\|_2^2$.

To simplify the analysis, we relax the stopping criterion such that we can store as many lattice points as possible in the list until a lattice point whose distance is larger than $ma_1 (= m\left\|\mathbf{z} - \mathbf{G}\hat{\mathbf{x}}_1^{\min}\right\|^2)$ is found[1]. In addition, we include the finally found lattice point into the list, which is the first lattice point violating the stopping criterion. Although it might sacrifice the practical condition (finite size list) little bit, this assumption greatly simplifies our analysis. With this assumption, the list can be expressed as $\mathcal{L}_1 = \{\hat{\mathbf{x}}_1^1, \hat{\mathbf{x}}_1^2, \cdots, \hat{\mathbf{x}}_1^{\mathrm{u}-1}, \hat{\mathbf{x}}_1^{\mathrm{u}}\}$ where $\hat{\mathbf{x}}_1^{\min} = \hat{\mathbf{x}}_1^1$ and $\hat{\mathbf{x}}_1^{\max} = \hat{\mathbf{x}}_1^{\mathrm{u}}$. Also, we have

$$\|\mathbf{z} - \mathbf{G}\hat{\mathbf{x}}_1^{\max}\|^2 \ge m\left\|\mathbf{z} - \mathbf{G}\hat{\mathbf{x}}_1^{\min}\right\|^2. \tag{68}$$

[1]In fact, it is highly possible that the list contains a finite number of lattice points in the moderate and high SNR regime.

By taking an expectation on (68), we have

$$E\left[\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right) + \mathbf{v}\|^2 \middle| \mathbf{H}\right] \ge mE\left[\left\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\min}\right) + \mathbf{v}\right\|^2 \middle| \mathbf{H}\right]. \tag{69}$$

After some manipulation, (69) becomes

$$\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\max}\right)\|^2 \ge m\left\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\min}\right)\right\|^2 + (m-1)\mathrm{tr}\left(\mathbf{\Phi}\right) \tag{70}$$

$$\ge m\left\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\min}\right)\right\|^2. \tag{71}$$

Using (67) and (71),

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right) \le Q\left(\sqrt{\frac{m\left\|\mathbf{G}\left(\mathbf{x}_{A1} - \hat{\mathbf{x}}_1^{\min}\right)\right\|^2}{2\|\mathbf{\Phi}\|_2}}\right). \tag{72}$$

Assume that $\left(\mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2\mathbf{I}\right)^{-1}$ is decomposed into $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$, where $\mathbf{U}$ is a unitary matrix and $\mathbf{\Lambda}$ is the diagonal matrix. The eigenvalues of $\left(\mathbf{H}_2\mathbf{H}_2^H + \sigma_w^2\mathbf{I}\right)^{-1}$ denoted as $\lambda_1 \ge \cdots \ge \lambda_{n_r}$ are on the main diagonal of $\mathbf{\Lambda}$. According to (23), $\mathbf{Z}$ can be decomposed into $\mathbf{U}\left(\sigma_w^2\mathbf{\Lambda}\right)\mathbf{U}^H$. In addition, $\|\mathbf{\Phi}\|_2$ equals $\sigma_w^4\lambda_1$ and hence

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right) \le Q\left(\sqrt{\frac{m\left\|\mathbf{\Lambda}\widetilde{\mathbf{H}}_1\left(\mathbf{x}_{1A} - \hat{\mathbf{x}}_1^{\min}\right)\right\|^2}{2\lambda_1}}\right). \tag{73}$$

where $\widetilde{\mathbf{H}}_1 = \mathbf{U}^H\mathbf{H}_1$.

To remove conditioning, $Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right)$ needs to be averaged over $\widetilde{\mathbf{H}}_1$ and $\mathbf{H}_2$ as

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right) = E_{\mathbf{H}_2}\left[E_{\widetilde{\mathbf{H}}_1}\left[Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right) \middle| \mathbf{H}_2\right]\right]. \tag{74}$$

Further, employing the property $Q(x) \le \exp\left(-\frac{x^2}{2}\right)$, we have

$$E_{\widetilde{\mathbf{H}}_1}\left[Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right) \middle| \mathbf{H}_2\right] \le E_{\widetilde{\mathbf{H}}_1}\exp\left(-\frac{m\sum_{i=1}^{n_r}\lambda_i^2\left\|\tilde{\mathbf{h}}_i^H\left(\mathbf{x}_{1A} - \hat{\mathbf{x}}_1^{\min}\right)\right\|^2}{4\lambda_1}\right), \tag{75}$$

where $\tilde{\mathbf{h}}_i^H$ is the $i$th row vector of $\widetilde{\mathbf{H}}_1$. With the assumption that the elements of $\widetilde{\mathbf{H}}_1$ (equivalently $\mathbf{H}_1$) is i.i.d. complex Gaussian, one can show that (75) is

$$E_{\widetilde{\mathbf{H}}_1}\left[Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1 \middle| \mathbf{H}\right) \middle| \mathbf{H}_2\right] \le \prod_{i=1}^{n_r}\frac{4\lambda_1}{4\lambda_1 + m\lambda_i^2\left\|\mathbf{x}_{1A} - \hat{\mathbf{x}}_1^{\min}\right\|^2}. \tag{76}$$

In practice, the distribution of $\mathbf{H}_1$ is not strictly Gaussian due to the symbol classification operation on $\mathbf{H}$. In performing the symbol classification, we hope that a partition of $\mathbf{H}$ leads to less average CEP or $E_{[\mathbf{H}_1,\mathbf{H}_2]}\left[Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right)\right]$ than without classification. Hence, we assume that i.i.d. Gaussian assumption on $\mathbf{H}_1$ does not affect the inequality of (75). Since the minimum distance between two constellation points is $2/\lambda$ from (3), $\left\|\mathbf{x}_{1A} - \hat{\mathbf{x}}_1^{\min}\right\|^2 \ge 4/\lambda^2$ and therefore (74) becomes

$$Pr_A\left(\mathbf{x}_{A1} \notin \mathcal{L}_1\right) \le E_{\mathbf{H}_2}\left[\prod_{i=1}^{n_r}\frac{\frac{\lambda_1}{\lambda_i^2}}{\frac{\lambda_1}{\lambda_i^2} + \frac{m}{\lambda^2}}\right]. \tag{77}$$

## APPENDIX IV
## PROOF OF (55)

Assuming that $\hat{\mathbf{x}}_1^i$, the $i$-th element of $\mathcal{L}_1$ equals $\mathbf{x}_{A1}$, then $Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right)$ becomes the probability that $\hat{\mathbf{x}}_2^i \neq \mathbf{x}_{A2}$. Under this condition, $\hat{\mathbf{x}}_1^i = \mathbf{x}_{A1}$ and the effect of $\mathbf{x}_{A1}$ is subtracted from $\mathbf{y}$ in (44) so that we can focus on the performance of the reduced size-system $\mathbf{y}' = \mathbf{H}_2\mathbf{x}_{A2} + \mathbf{w}$ to evaluate $Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right)$. In [36], the error analysis of DF detector without detection ordering has been provided. Let $\hat{\mathbf{x}}_2^i(i_k)$ be the $i_k$th element of $\mathbf{x}_{A2}$ which is detected in the $k$-th order by the MMSE-DF detector. Given $\mathbf{H}_2$, the error probability of the MMSE-DF detector is

$$
\begin{aligned}
&Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1, \mathbf{H}_2\right) \\
&= Pr_A\left(\hat{\mathbf{x}}_2^i \neq \mathbf{x}_{A2} | \hat{\mathbf{x}}_1^i = \mathbf{x}_{A1}, \mathbf{H}_2\right) \qquad (78) \\
&= 1 - Pr_A\left(\hat{\mathbf{x}}_2^i = \mathbf{x}_{A2} | \hat{\mathbf{x}}_1^i = \mathbf{x}_{A1}, \mathbf{H}_2\right) \qquad (79) \\
&= 1 - \prod_{k=1}^{n_2} Pr_A\left(\hat{\mathbf{x}}_2^i(i_k) = \mathbf{x}_{A2}(i_k) | \hat{\mathbf{x}}_2^i(i_1) = \mathbf{x}_{A2}(i_1),\right. \\
&\left. \cdots, \hat{\mathbf{x}}_2^i(i_{k-1}) = \mathbf{x}_{A2}(i_{k-1}), \hat{\mathbf{x}}_1^i = \mathbf{x}_{A1}, \mathbf{H}_2\right) \qquad (80)
\end{aligned}
$$

The second term in (80) refers the product of the probabilities of correct symbol detection under the perfect feedback assumption. We can express the MMSE estimate $\hat{\mathbf{x}}_2^i\left(i_k\right)$ as $\hat{\mathbf{x}}_2^i\left(i_k\right) = \mathbf{x}_{1A}\left(i_k\right) + u_k$, where $u_k$ is the MMSE estimation error. One can easily show that the variance of $u_k$, denoted as $\sigma_{u_k}^2$, is given by

$$
\sigma_{u_k}^2 = 1 - \mathbf{h}_{i_k}\left(\mathbf{H}_{2,\left[i_k:i_{n_2}\right]}\mathbf{H}_{2,\left[i_k:i_{n_2}\right]}^H + \sigma_w^2\mathbf{I}\right)^{-1}\mathbf{h}_{i_k}^H. \qquad (81)
$$

We assume that the MMSE estimation error $u_k$ is zero-mean Gaussian with the variance $\sigma_{u_k}^2$. Then, the bound of the conditional error probability is

$$
\begin{aligned}
&Pr_A\left(\hat{\mathbf{x}}_2^i(i_k) = \mathbf{x}_{A2}(i_k) \middle| \hat{\mathbf{x}}_2^i(i_1) = \mathbf{x}_{A2}(i_1),\right. \\
&\left. \cdots, \hat{\mathbf{x}}_2^i(i_{k-1}) = \mathbf{x}_{A2}(i_{k-1}), \hat{\mathbf{x}}_1^i = \mathbf{x}_{A1}, \mathbf{H}_2\right) \\
&\geq 1 - 4\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{2}{\sigma_{u_k}^2\lambda^2}}\right). \qquad (82)
\end{aligned}
$$

This equals the error probability for detection of a QAM signal corrupted by Gaussian noise with the variance $\sigma_{u_k}^2$ [38]. By using (80) and (82), the upper bound of $Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right)$ is expressed as

$$
\begin{aligned}
&Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1\right) \\
&= E_{\mathbf{H}_2}\left[Pr_A\left(\mathbf{x}_{A2} \notin \mathcal{L}_2 | \mathbf{x}_{A1} \in \mathcal{L}_1, \mathbf{H}_2\right)\right] \qquad (83) \\
&\leq 1 - E_{\mathbf{H}_2}\left[\prod_{k=1}^{n_2}\left(1 - 4\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{2}{\sigma_{u_k}^2\lambda^2}}\right)\right)\right]. \qquad (84)
\end{aligned}
$$

## REFERENCES


[1] G. J. Foschini and M. J. Gans, "On Limits of Wireless Communication a Fading Environment when Using Multiple Antennas," *Wireless Personal Communications*, vol. 6, pp. 311-335, March 1998.

[2] E. Telatar, "Capacity of multi-antenna Gaussian channels,". *European Transactions on Telecommunications*, vol. 10 pp. 585-596, Nov. 1999.

[3] 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, Apr. 1985.

[4] E. Viterbo and E. Giglieri, "A universal lattice decoder," in *GRESTSI 14-eme Colloque*, Juan-les-Pins, France, Sep. 1993.

[5] M. O. Damen, H. E. Gamel, 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.

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

[7] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm II. Generalizations, second-order statistics, and applications to communications," *IEEE Trans. Signal Processing*, vol. 53, pp. 2819-2834, Aug. 2005.

[8] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181-191, 1994.

[9] E. Agrell, T. Eriksson, A. Vardy, and K. Zegar, "Closet point search in lattices," *IEEE Trans. Information Theory*, vol. 48, pp. 2201-2214, Aug. 2002.

[10] A. M. Chan and I. Lee, "A new reduced-complexity sphere decoder for multiple antenna systems," *Proc. Int. Conf. Commun.*, April 2002, pp. 460-464.

[11] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communication," *IEEE Trans. Signal Proceesing*, vol. 53, pp. 1474-1484, April 2005.

[12] A. Li, W. Xu, Y. wang, Z. Zhou, and J. Wang, "A faster ML sphere decoder with competing branches," *Proc. IEEE VTC Conf.*, June 2005, pp. 438-441.

[13] W. Zhao and G. B. Giannakis, "Sphere decoding algorithms with improved radius search," *IEEE Trans. Commun.*, vol. 53, pp. 1104-1109, July 2005.

[14] W. Zhao and G. B. Giannakis, "Reduced complexity closest point decoding algorithms for random lattices," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 101-111, Jan. 2006.

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

[16] B. Shim and I. Kang, "Radius adaptive sphere decoding via proabilistic tree pruning," *Proc. IEEE SPAWC*, June 2007, pp. 1-5.

[17] 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.*, pp. 295-300, Sep. 1998.

[18] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, vol. 13, pp. 675-685, Nov. 1969.

[19] W. Xu, Y. Wang, Z. Zhou, and J. Wang, "A computationally efficient exact ML sphere decoder," *Proc. IEEE Global Telecommun. Conf.*, Nov. 2004, pp. 2594-2598.

[20] 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.

[21] M. Kokkonen and K. Kalliojarvi, "Soft-decision decoding of binary linear codes using the t-algorithm, *in Proc. IEEE 8th Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC)*, Finland, Sep. 1997, pp. 1181-1185.

[22] W. K. Ma, T. N. Davidson, K. Wong, Z. Q. Luo, and P. C. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA," *IEEE Trans. Signal Processing*, vol. 50, no. 4, pp. 912-922, April 2002.

[23] 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.

[24] X. Li, H. C. Huang, A. Lozano, and G. J. Foschini, "Reduced-complexity detection algorithms for systems using multi-element arrays," *Proc. IEEE Global Telecommun. Conf.*, vol. 2, pp. 1072-1076, Nov. 2000.

[25] A. Elkhazin, K. N. Plataniotis, and S. Pasupathy, "Reduced-dimension MAP turbo-BLAST detection," *IEEE Trans. Commun.*, vol. 54, pp. 108-118, Jan. 2006.

[26] W. Choi, R. Negi, and J. Cioffi, "Combined ML and DFE decoding for the V-BLAST system," *Proc. IEEE ICC*, vol. 3, Nov. 2000, pp. 1243-1248.

[27] A. Wolfgang, J. Akhtman, S. Chen, and L. Hanzo, "Reduced complexity near-maximum-likelihood detection for decision feedback assisted space-time equalization," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 2407-2411, July 2007.

[28] L. G. Barbero and J. S. Thompson, "Performance analysis of a fixed-complexity sphere decoder in high-dimensional MIMO systems," *Proc. IEEE ICASSP*, Toulouse, France, May 2006.

[29] J. Jalden, L. G. Barbero, B. Ottersten, and J. S. Thompson, "Full diversity detection in MIMO systems with a fixed-complexity sphere decoder," *Proc. IEEE ICASSP*, Hawaii, USA, April 2007.
[30] B. Hochwald and S. T. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, pp. 389-399, Mar. 2003.
[31] J. Hagenauer and C. Kuhn, "The list-sequential (LISS) algorithm and its application," *IEEE Trans. Commun.*, vol. 55, pp. 918-928, May 2007.
[32] M. L. Honig, G. K. Woodward, and Y. Sun, "Adaptive iterative multiuser decision feedback detection," *IEEE Trans. Wireless Commun.*, vol. 3, no. 2, pp. 477-485, Mar. 2004.
[33] E. Biglieri, G. Taricco, and A. Tulino, "Decoding space-time codes with BLAST architectures," *IEEE Trans. Signal Processing*, vol. 50, pp. 2547-2552, Oct. 2002.
[34] J. W. Choi, B. Shim, A. C. Singer, and N. I. Cho, "A low-complexity near-ML decoding technique via reduced dimension list stack algorithm," *IEEE SAM 2008*, accepted for publication.
[35] 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.
[36] N. Prasad and M. K. Varanasi, "Analysis of decision feedback detection for MIMO rayleigh-fading channels and the optimization of power and rate allocations," *IEEE Trans. Information Theory*, vol. 50, pp. 1009-1025, June 2004.
[37] F. D. Neeser and J. L. Massey, "Proper complex random processes with application to information theory," *IEEE Trans. Information Theory*, vol. 39, pp. 1293-1302, July 1993.
[38] J. G. Proakis, "Digital communications," 4th Ed. *Mc Grow Hill*, 2001.