Partition-Based Hybrid Decoding (PHD): A Class of ML Decoding Schemes for MIMO Signals Based on Tree Partitioning and Combined Depth- and Breadth-First Search

J. I. Park[1], Y. Lee[2], S. Yoon[*3]

[1,2,3] College of Information and Communication Engineering
Sungkyunwan University
Suwon, Korea
*syoon@skku.edu

ABSTRACT
In this paper, we propose a hybrid maximum likelihood (ML) decoding scheme for multiple-input multiple-output (MIMO) systems. After partitioning the searching tree into several stages, the proposed scheme adopts the combination of depth- and breadth-first search methods in an organized way. Taking the number of stages, the size of signal constellation, and the number of antennas as the parameter of the scheme, we provide extensive simulation results for various MIMO communication conditions. Numerical results indicate that, when the depth- and breadth-first search methods are employed appropriately, the proposed scheme exhibits substantially lower computational complexity than conventional ML decoders while maintaining the ML bit error performance.

Keywords: Hybrid decoding, multiple input multiple output (MIMO), maximum likelihood detection, tree partitioning.

1. Introduction

It is well known that a significant gain in the spectral efficiency can be secured if we utilize the multiple-input multiple-output (MIMO) system when compared to the use of a single-input single-output system. The MIMO system is one of the promising techniques for next-generation communication systems because of its increased capacity without requiring consumption of excess frequency spectrum [1]. In particular, the spectral efficiency of the MIMO systems stems from the spatial multiplexing and increases as the number of antennas employed increases. Due to the increased interference among the transmit antennas or mobility of the devices [2], on the other hand, a more challenging decoding task at the receiver side is an unavoidable expense in making use of the advantages of MIMO systems comprehensively.

The problem of designing computationally efficient MIMO decoders has been addressed in many studies. Among the representative examples of computationally efficient decoders are such suboptimal decoders as the zero-forcing, nulling and cancelling, and nulling and cancelling with optimal ordering schemes [3]. With the bit error rate (BER) not equal to the ML performance, the suboptimal decoders are not appropriate in some situations to fully utilize the advantages inherent in the MIMO systems despite their low computational complexity.

Theoretically, a maximum likelihood (ML) decoder achieves the optimal BER performance for MIMO systems. The conventional full-search ML decoder inspecting all the lattice points in a tree requires a computational complexity that grows excessively fast as the number of antennas or the size of signal constellation increases. To alleviate the computational complexity of the conventional full-search ML decoder while maintaining the optimal BER performance, the sphere decoder (SD) has been proposed and analyzed in [4]. By searching over only the lattice points lying inside a hypersphere centered at the received signal point, SD reduces the search space and consequently the required computational complexity. Specifically, it is reported that, when the signal-to-noise ratio (SNR) is high and the number of antennas is small, the computational complexity of SD is comparable to that of suboptimal decoders. However, the computational complexity of SD increases significantly as the number of antennas increases or the SNR becomes low [5]. The breadth-first signal decoder (BSIDE), searching the closest lattice point based on a breadth-first search method, is proposed and shown to exhibit a lower computational complexity than SD and the optimal BER performance [6]. Nonetheless, it seems that BSIDE does not provide sufficient complexity reduction when the size of signal constellation is large. Recently, several low complexity decoding schemes were proposed employing a partial ML decoder with the successive interference cancellation detector [7] and with the chase detector [8]. However, the schemes in [7] and [8] cannot provide the optimal detection performance.

In this paper, we propose a novel ML decoding scheme, called the partition-based hybrid decoding (PHD), which first partitions the tree into several stages, and then applies depth- or breadth-first search method appropriately in each of the stages to find the ML solution. By determining a set of candidate nodes leading to the ML solution at the end of each of the stages (except for the last stage) and inspecting the more-likely candidates first, PHD can reduce the computational complexity substantially. The computational gain of PHD is shown to be more significant when the SNR is lower and when the number of antennas is larger.

The rest of this paper is organized as follows. The MIMO system model assumed in this paper is described in Section 2 and details of PHD are presented in Section 3. The computational complexity and BER performance of BSIDE, PHD, and SD are explored and compared in Section 4. Finally, Section 5 summarizes this paper.

2. System model

Let us consider a MIMO system with N_t transmit and N_r receive antennas as shown in Figure 1. At the transmitter side, the input data stream is divided into N_t substreams, each of which is then transmitted through a dedicated transmit antenna over a flat Rayleigh fading channel. The channel is assumed to be constant over one transmission period but may change from one transmission to another.

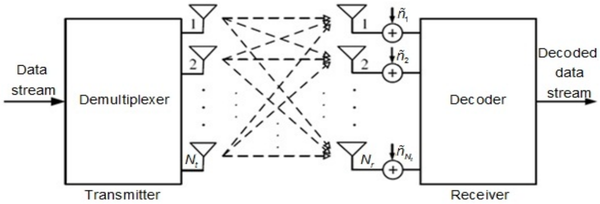


Figure 1. An example of a MIMO system with N_t Transmit and N_r receive antennas.

Then, the baseband complex received signal $\underline{\tilde{r}} = \left[\tilde{r}_1, \tilde{r}_2, \cdots, \tilde{r}_{N_r}\right]^T$ can be formulated as

$$\underline{\tilde{r}} = \tilde{H}\underline{\tilde{s}} + \underline{\tilde{n}}, \tag{1}$$

where $\tilde{H}$ is the $N_r \times N_t$ complex channel transfer matrix, $\underline{\tilde{s}} = \left[\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_{N_t}\right]^T$ is an $N_t \times 1$ complex transmitted signal vector, and $\underline{\tilde{n}} = \left[\tilde{n}_1, \tilde{n}_2, \cdots, \tilde{n}_{N_t}\right]^T$ is an complex noise vector. Here, the superscript T signifies the transpose of a matrix. The elements $\{\tilde{h}_{i,j}\}$ of the complex channel transfer matrix $\tilde{H}$ are independent and identically distributed (i.i.d) complex Gaussian random variables with mean zero and variance one: we assume that all the complex channel transfer coefficients $\{\tilde{h}_{i,j}\}$ have perfectly been estimated before the beginning of the decoding procedure at the receiver. The additive noise components $\{\tilde{n}_i\}$ are i.i.d complex Gaussian random variables with mean zero and variance $\sigma^2/2$ per dimension. We assume that the complex transmitted signals $\{\tilde{s}_i\}$ are drawn from the constellation of an L^2-quadrature amplitude modulation (QAM), where L is a power of 2.

To simplify the description of PHD, let us now convert the complex matrix and vectors in Eq. 1 into real expressions. For convenience, let $M = 2N_t$ and $N = 2N_r$. From Eq. 1 we then have

$$\underline{r} = H\underline{s} + \underline{n}, \tag{2}$$

where $\underline{r} = \left[\Re\left\{\underline{\tilde{r}}^T\right\}\ \Im\left\{\underline{\tilde{r}}^T\right\}\right]^T = [r_1, r_2, \cdots, r_N]^T$ denotes the received signal vector,

$$H = \begin{pmatrix} \Re\{\tilde{H}\} & -\Im\{\tilde{H}\} \\ \Im\{\tilde{H}\} & \Re\{\tilde{H}\} \end{pmatrix} \tag{3}$$

denotes the real representation of the channel transfer matrix $\tilde{H}$, $\underline{s} = \left[\Re\left\{\underline{\tilde{s}}^T\right\}\ \Im\left\{\underline{\tilde{s}}^T\right\}\right]^T = [s_1, s_2, \cdots, s_M]^T$ is the transmitted signal vector in the real domain, and $\underline{n} = \left[\Re\left\{\underline{\tilde{n}}^T\right\}\ \Im\left\{\underline{\tilde{n}}^T\right\}\right]^T = [n_1, n_2, \cdots, n_N]^T$ is the noise vector. Here, $\Re\{\cdot\}$ and $\Im\{\cdot\}$ designate the real and imaginary parts, respectively. Since $\tilde{s}_i$ is drawn from an L^2-QAM constellation, we have $\underline{s} \in S_L^M$, where it is assumed that

$$S_L = \left\{-\frac{(L-1)}{2}, -\frac{(L-3)}{2}, \ldots, \frac{(L-3)}{2}, \frac{(L-1)}{2}\right\}. \tag{4}$$

Performing the QR decomposition on the channel transfer matrix H when $N \geq M$, we can rewrite Eq. 2 as [9]

$$\underline{r} = \underbrace{[Q_1\ Q_2]}_{Q} \begin{bmatrix} R \\ 0_{N-M,M} \end{bmatrix} \underline{s} + \underline{n}, \tag{5}$$

where $Q = [Q_1\ Q_2]$ is an $N \times N$ unitary matrix, $R = [r_{i,j}]$ is an $M \times M$ upper triangular matrix, and $0_{N-M,M}$ is an all-zero matrix of size $(N-M) \times M$. Multiplying $Q^T = [Q_1\ Q_2]^T$ from the left on both sides of Eq. 5, we obtain

$$\begin{bmatrix} \underline{y} \\ \underline{y}' \end{bmatrix} = \begin{bmatrix} R\underline{s} \\ 0_{N-M,M} \end{bmatrix} + \begin{bmatrix} \underline{w} \\ \underline{w}' \end{bmatrix}, \tag{6}$$

where $\underline{y} = Q_1^T \underline{r} = [y_i]$, $\underline{w} = Q_1^T \underline{n}$, and $\underline{y}' = \underline{w}' = Q_2^T \underline{r} = Q_2^T \underline{n}$. It is noteworthy that only $\underline{y}$, but not $\underline{y}'$, is dependent (or contains information) on the transmitted signal $\underline{s}$.

Let us now consider the ML decoding of a transmitted signal vector $\underline{s}$. Given a received vector $\underline{r}$ and the channel transfer matrix H, the distance between $\underline{r}$ and $H\underline{s}$ is calculated as

$$\begin{aligned} \left\|\underline{r} - H\underline{s}\right\|^2 &= \left(\underline{r} - H\underline{s}\right)^T QQ^T \left(\underline{r} - H\underline{s}\right) \\ &= \left\|Q_1^T \underline{r} - R\underline{s}\right\|^2 + \left\|Q_2^T \underline{r}\right\|^2 \\ &= \left\|\underline{y} - R\underline{s}\right\|^2 + \left\|Q_2^T \underline{r}\right\|^2 \end{aligned} \tag{7}$$

since $QQ^T = I$, where $\|\cdot\|$ denotes the Euclidean norm. As we have observed in Eq. 6, $\left\|Q_2^T \underline{r}\right\|^2 = \left\|\underline{y}'\right\|^2$ is not dependent on $\underline{s}$, and therefore, the ML solution $\hat{\underline{s}}$ is obtained eventually as

$$\hat{\underline{s}} = \arg\min_{\underline{s} \in S_L^M} \left\|\underline{r} - H\underline{s}\right\|^2 = \arg\min_{\underline{s} \in S_L^M} \left\|\underline{y} - R\underline{s}\right\|^2. \tag{8}$$

If $N < M$, on the other hand, the channel transfer matrix is first partitioned into $H = [H_1, H_2]$, where H_1 and H_2 are of sizes $N \times N$ and $N \times (M-N)$, respectively. Performing QR decomposition on H_1, we have [9]

$$\underline{r} = [QR, H_2]\underline{s} + \underline{n} \tag{9}$$

from Eq. 2, where Q is an $N \times N$ unitary matrix and R is an $N \times N$ upper triangular matrix. If we multiply Q^T from the left on both sides of Eq. 9, we obtain $\underline{y} = R\underline{s}_1 + Q^T H_2 \underline{s}_2 + \underline{n}'$, where $\underline{y} = Q^T \underline{r}$, $\underline{s}_1 = [s_1, s_2, \cdots, s_N]^T$, $\underline{s}_2 = [s_{N+1}, s_{N+2}, \cdots, s_M]^T$, and $\underline{n}' = Q^T \underline{n}$. In order to find the ML solution $\hat{\underline{s}}$, we first fix $\underline{s}_2$ and search over $\underline{s}_1$ to minimize $\left\|\underline{y} - R\underline{s}_1 - Q^T H_2 \underline{s}_2\right\|^2$ and then the procedure is repeated for every choice of $\underline{s}_2 \in S_L^{M-N}$. The ML solution $\hat{\underline{s}}$ can thus be found as

$$\hat{\underline{s}} = \arg\min_{\underline{s}_2 \in S_L^{M-N}} \min_{\underline{s}_1 \in S_L^N} \left\|\underline{y} - R\underline{s}_1 - Q^T H_2 \underline{s}_2\right\|^2. \tag{10}$$

In this paper, we assume that $N \geq M$ for simplicity: however, all the results in this paper can be readily extended to the situation $N < M$ also.

3. Proposed decoding scheme

3.1 Preliminaries

Using that the matrix R is upper triangular, the tree structure is assumed to find the ML solution $\underline{\hat{s}}$ in PHD as in other well-known decoding schemes [10]. Considering an $N_r \times N_t$ MIMO system with L^2-QAM, an L-ary tree with $M \ (= 2N_t)$ layers starting from a root is created. The *i*th node in the *k*th layer of the tree is denoted by the vector $\underline{s}_k^{(i)} = [s_{k,k}^{(i)}, s_{k+1,k}^{(i)}, \cdots, s_{M,k}^{(i)}]^T$ for $k = 1, 2, \cdots, M$ and $i = 1, 2, \cdots, L^{M-k+1}$ with $\underline{s}_{M+1}^{(1)}$ denoting the root (the only node in layer $M+1$) of the tree. Since the elements $\{s_{j,k}^{(i)}\}_{j=k}^{M}$ of $\underline{s}_k^{(i)}$ are drawn from S_L, a transmitted vector $\underline{s}$ can be represented by a node $\underline{s}_1^{(i)}$ in the first layer of the tree. Figure 2 shows an example of the tree when $M = 4$ and $L = 2$ with $S_2 = \{-\frac{1}{2}, \frac{1}{2}\}$.

Defining the (node) metric of a node $\underline{s}_k^{(i)}$ as

$$\Lambda(\underline{s}_k^{(i)}) = \sum_{j=k}^{M} \left(y_j - \sum_{l=j}^{M} r_{j,l} s_{l,k}^{(i)} \right)^2, \tag{11}$$

it is clear from Eq. 8 that the goal of the tree searching in an ML decoding scheme is to find a vector in the first layer whose node metric is the smallest among $\{\Lambda(\underline{s}_1^{(i)})\}_{i=1}^{L^M}$. In the context of the tree searching in this paper, other such commonly-used traditional terminology as the child node, descendant node, predecessor node, and subtree will be adopted with the usual definitions as in [11].

3.2 The proposed decoding scheme

The proposed decoding scheme, PHD, first partitions the tree into several stages of layers. For a tree with M layers, we can in general partition the tree into T stages with stage i containing M_i layers, where $i = 1, 2, \cdots, T$, $T \in \{1, 2, \cdots, M\}$, $M_i \in \{1, 2, \cdots, M\}$, and $\sum_{i=1}^{T} M_i = M$. Clearly, the *l*th stage is composed of layers $M - \sum_{j=1}^{l-1} M_j$, $M - \sum_{j=1}^{l-1} M_j - 1, \cdots,$ and $M - \sum_{j=1}^{l} M_j + 1$ in the tree. In the example of Figure 2 for instance, the first, second, and third stages are the collections {layer 4}, {layer 3, layer 2}, and {layer 1}, respectively.

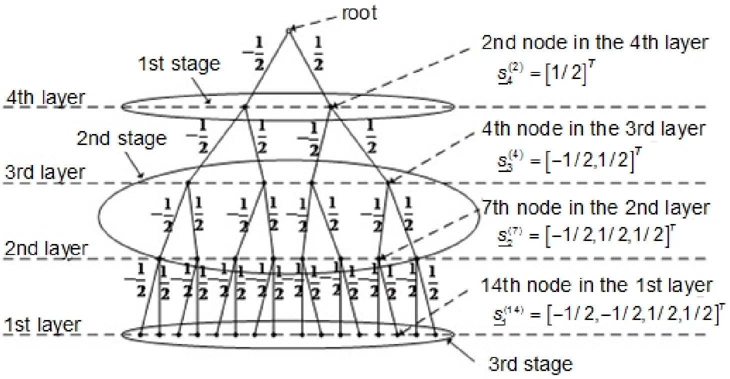

root
1st stage
$-\frac{1}{2}$ $\frac{1}{2}$
2nd node in the 4th layer
$\underline{s}_4^{(2)} = [1/2]^T$
4th layer
2nd stage
$-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$
4th node in the 3rd layer
$\underline{s}_3^{(4)} = [-1/2, 1/2]^T$
3rd layer
$-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$
7th node in the 2nd layer
$\underline{s}_2^{(7)} = [-1/2, 1/2, 1/2]^T$
2nd layer
$-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$ $-\frac{1}{2}$ $\frac{1}{2}$
14th node in the 1st layer
$\underline{s}_1^{(14)} = [-1/2, -1/2, 1/2, 1/2]^T$
1st layer
3rd stage

Figure 2. An example of the tree with $M = 4$, $L = 2$, and $S_2 = \{-\frac{1}{2}, \frac{1}{2}\}$. The tree is partitioned into 3 stages (i.e.,$T = 3$). when $\varepsilon_l = 1$, $\varepsilon_f = 0.4$, $N = 8$, and the noise is absent.

In each of the stages, a depth- or breadth-first search method can be employed appropriately, and the search is restricted to the subtrees rooted from the nodes survived in the previous stage. Here, the root $\underline{s}_{M+1}^{(1)}$ of the whole tree is assumed to be the only node survived prior to the first stage.

3.2.1 Step 1: Determining candidates for the ML solution in the first (T-1) stages

In each of the $(T-1)$ stages in Step 1, a set of nodes to be used as the roots of the subtrees in the next stage is determined at the bottom layer of the stage by applying a depth-first or a breadth-first search method. The nodes survived are those with metrics smaller than or equal to the threshold $D_{DFE}^2 = \left\| \underline{y} - R\breve{\underline{s}} \right\|^2$, the metric of the solution $\breve{\underline{s}}$ of the decision feedback equalizer (DFE). The merit of employing the DFE metric D_{DFE}^2 as the threshold is that the DFE requires less computational complexity than other well-known algorithms utilizing QR decomposition. By using D_{DFE}^2, it is in addition guaranteed that the ML solution is not discarded during the search in Step 1 since $D_{ML}^2 = \left\| \underline{y} - R\hat{\underline{s}} \right\|^2 \le \left\| \underline{y} - R\breve{\underline{s}} \right\|^2 = D_{DFE}^2$, where the ML metric D_{ML}^2 signifies the metric of the ML solution $\hat{\underline{s}}$. Here, D_{ML}^2 measures the metric between the ML solution in the first layer and the root of a tree, or equivalently, the square of the distance between the transformed received signal $\underline{y} = Q_1^T \underline{r}$ and the closest lattice point among those constructed by the matrix R and signal constellation S_L^M.

At the end of the lth stage, a set

$$\Omega_l = \left\{ (\underline{s}_k^{(i)}, \Lambda(\underline{s}_k^{(i)})) \mid \Lambda(\underline{s}_k^{(i)}) \le D_{DFE}^2 \right\} \qquad (12)$$

of N_l pairs of nodes and their node metrics at the bottom layer (that is, the layer with $k = M - \sum_{j=1}^{l} M_j + 1$) of the stage is determined by employing appropriately a depth- or breadth-first search method for $l = 1, 2, \cdots, T-1$. Here, the numbers $\{N_l\}$ of survived nodes are such that $L^{\sum_{j=1}^{T-1} M_j} \ge N_{T-1} \ge N_{T-2} \ge \cdots \ge N_1 \ge 1$. In the $(l+1)$st stage, the search continues over the N_l subtrees rooted from the N_l nodes in Ω_l.

When a depth-first search method is employed in a stage of Step 1, the scheme traverses toward a lower layer by adding only one child node with node metric smaller than or equal to the threshold at a time until it reaches the bottom layer of the stage or until all the child nodes of a node are explored and found to have metrics larger than or equal to the threshold. Then a backtrack occurs and the search returns to the predecessor node in the lowest layer having at least one unexplored child node, at which the depth-first search method inspects an unexplored child node with metric smaller than or equal to the threshold. The procedure continues until the depth-first search method reaches back the root of the (sub)tree with no node left unexplored.

On the other hand, when a breadth-first search method is adopted in a stage of Step 1, the scheme first computes all the node metrics in the top layer and then discards nodes with the node metrics larger than the threshold. The node metrics in the layer immediately below are then computed only for the nodes stemmed from the survived nodes. Such a procedure continues until all the nodes with node metric smaller than or equal to the threshold are found in the bottom layer of the stage.

3.2.2 Step 2: Finding the node with the shortest node metric in the first layer

In the last stage, setting the threshold to D_{DFE}^2 at the beginning, the N_{T-1} subtrees stemmed from the N_{T-1} nodes in Ω_{T-1} are searched by employing one of such various ML decoders based on the depth- or breadth-first search methods as the BSIDE, list sphere decoder (LSD), and SD.

As the node in the first layer with the shortest metric will be declared the ML solution, it is desired to find the node preferably without searching all the subtrees stemmed from the nodes survived in stage $(T-1)$. To that end, PHD will search the N_{T-1} subtrees in the ascending order of the node metric of the N_{T-1} nodes in Ω_{T-1} by re-arranging the nodes in Ω_{T-1} as

$$\Omega'_{T-1} = \{(\underline{s}_k^{[i]}, \Lambda(\underline{s}_k^{[i]})) \mid \underline{s}_k^{[i]} \in \Omega_{T-1}, \\ \Lambda(\underline{s}_k^{[1]}) \le \Lambda(\underline{s}_k^{[2]}) \le \cdots \le \Lambda(\underline{s}_k^{[N_{T-1}]})\}, \tag{13}$$

where $k = M - \sum_{j=1}^{T-1} M_j + 1 = M_T + 1$. At the same time, to maximize the probability of finding the ML solution as soon as possible, PHD replaces the threshold with the metric of a node in the first layer whenever the metric is found to be smaller than the threshold.

Specifically, starting from the subtree rooted from the first node $\underline{s}_k^{[1]}$, we search for the node with the smallest metric in the first layer. When the search over the subtree rooted from $\underline{s}_k^{[n-1]}$ is completed, if the metric $\Lambda(\underline{s}_k^{[n]})$ of $\underline{s}_k^{[n]}$ is compared to be larger than the threshold (which might have been changed from the starting value D_{DFE}^2 in the meantime), the node of which the metric became the threshold is declared the ML solution and the search is terminated. This is based on the fact that all the descendant nodes of $\{\underline{s}_k^{[j]}\}_{j=n}^{N_{T-1}}$ will have node metrics larger than the shortest node metric found by that moment. The procedure of inspecting more-likely subtree (stemmed from a node with smaller metric) first and appropriately updating the threshold maximizes the probability of finding the ML solution as soon as possible, and consequently, reduces the computational complexity by preventing us from unnecessarily searching hopeless subtrees.

In contrast to conventional decoding schemes, in which only one search method is employed for the search over the entire tree, PHD enables us to employ various search methods adaptively in each of the stages of the partitioned tree when such parameters in the decoding environment as the SNR, number of antennas, and size of signal constellation change. By employing appropriate search methods in the right stages, PHD can maximally exploit the advantages of several search methods and accordingly reduce the computational complexity. In addition, as a natural advantage of the tree partitioning, PHD searches more-likely candidate subtrees first and terminates the searching process as soon as the ML solution is found, resulting in further reduction of the computational complexity.

4. Performance evaluation

4.1 Analysis of computational complexity

The number of multiplications (NOMs) is a useful index accepted widely in illustrating the computational complexity of a decoding scheme [12], In this paper also, the computational complexity of PHD is investigated and compared with those of other ML decoding schemes in terms of the number of multiplications.

At the beginning of PHD, the QR decomposition of H is performed and $\underline{y} = Q_1^T \underline{r}$, $\breve{\underline{s}}$, and $D_{DFE}^2 = \left\| \underline{y} - R\underline{s} \right\|^2$ are computed. The number of multiplications required for the QR decomposition is approximately $NM^2 - M^3/3$ [13]. Computing $\underline{y}$, D_{DFE}^2, and $\breve{\underline{s}}$ require NM multiplications, $M(M+3)/2$ multiplications, and M divisions, respectively [6]. Assuming that a division requires the same amount of computational complexity as a multiplication, the total number of multiplications required for these computations is thus

$$NM(M+1) - \frac{M^3}{3} + \frac{M(M+5)}{2}. \tag{14}$$

Note that Eq. 14 is the number of multiplications required independent of the SNR and size of signal constellation.

The computational complexity of PHD can be decided by adding the number Eq. 14 to the total number of multiplications in the T stages. The number of multiplications in a stage will of course depend on the decoding scheme employed and the number of nodes with metric smaller than or equal to the threshold in the stage. Now, the number of nodes with metric smaller than or equal to the threshold is a random variable since it is a function of the channel transfer matrix H and received vector $\underline{r}$, both of which are clearly random variables depending on the SNR. Taking in addition into account that the exact path of a decoding scheme over the tree is unpredictable, it is obvious that the number of multiplications (of any decoder) will differ from transmission to transmission. Consequently, it is impossible to acquire an exact closed-form expression for the number of multiplications of a decoding scheme in general. In this paper, therefore, the number of multiplications of various decoding schemes are obtained and compared by means of averaging over a large number of repetitions via computer simulations.

4.2 Simulation results and discussions

We now evaluate and compare the computational complexities of BSIDE, PHD, and SD in terms of the average number of multiplications over 10^6 iterations via computer simulations. For simulations, we consider Rayleigh fading channels with additive white Gaussian noise [14], [15]. The BER performance characteristics of BSIDE, PHD, and SD will also be shown when appropriate to confirm that they all have the ML performance. Additional interesting characteristics of PHD will be discussed in terms of the relationship between the configurations of stages and NOMs in several MIMO transmission scenarios.

In all the simulations herein, the SNR is defined as $E\{\|H\underline{s}\|^2\}/E\{\|\underline{n}\|^2\}$, where $E\{\cdot\}$ denotes the expectation. In specifying the structure (configuration) of PHD, we will use the notation $\mathrm{PHD}[\{A_j(M_j)\}_{j=1}^{T}]$ to denote that search method A_j is employed in the jth stage composed of M_j layers for $j=1,2,\cdots,T$. In representing search methods $\{A_j\}$, the symbols ‘BF’, ‘BS’, ‘LS’, and ‘SD’ will stand for the breadth-first search, BSIDE, LSD, and SD, respectively. It is noteworthy that among these four search methods, ‘BS’ and ‘SD’ can be used only in Step 2 of PHD, ‘BF’ can be used only in Step 1 of PHD, and ‘LS’ can be used both in Steps 1 and 2. Consequently, there may exist six possible combinations of search methods in Steps 1 and 2 of PHD. In the simulations, we set the initial searching radius of LSD and SD to D_{DFE} and assume the computational complexity of a square root is the same as that of a multiplication.

4.2.1 On the numbers of stages

Figure 3 shows the average number of multiplications of PHD as a function of the number T of stages when various search methods are adopted in PHD and $N_t = N_r = 4$. It is interesting to observe, when the search methods are all the same in the first $(T-1)$ stages, that (1) the computational complexity of PHD depends mainly on the search method and the number of layers in the last stage and (2) the changes in the number of stages and layers in the first $(T-1)$ stages barely influence the computational complexity. For example, the computational complexities of PHD[LS(6),SD(2)] and PHD[LS(2),LS(3),SD(3)] are almost the same as that of PHD[LS(2),LS(2),LS(2),SD(2)] and PHD[LS(2),LS(1),LS(1),LS(1),SD(3)], respectively. Although we have shown the results for only a limited number of combinations for the search methods, we have confirmed that other combinations also result in the same tendency.

From this observation, we can infer that, in designing or choosing the configuration of PHD, a number of contiguous stages with the same search method in Step 1 may be merged into one stage with the common search method without affecting the overall computational complexity. In other words, unless two or more search methods are employed in Step 1, partitioning a tree into two stages should be a reasonable choice.

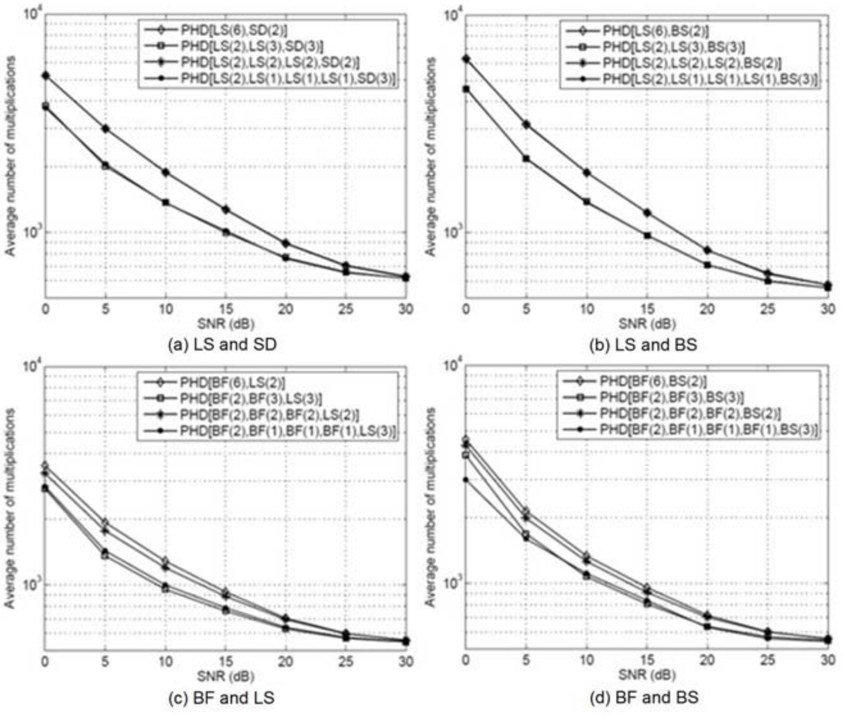
4.2.2 On the numbers of layers in stages

In Figure 4, for all possible (=six) combinations of search methods in the two stages, we have shown the dependence of the computational complexity of PHD on the numbers M_1 and $M_2 = 8 - M_1$ of layers in the two stages when $N_t = N_r = 4$, $L = 4$, and $T = 2$. Clearly, the computational complexity of PHD depends on the specific configuration and varies as the search methods and numbers of layers in the two stages change: yet, the difference in the numbers of multiplications among the various configurations of PHD decreases rapidly as the SNR increases. Similar observations have been made for other values of L also, which are not included in this paper for a space reason.

Although different choices of the numbers of layers in the two stages will result in different level of computational complexity, $M_1 = M_2 = M/2$ is not an unreasonable choice in the investigation of the computational complexity of PHD in terms of the average number of multiplications. Intuitively, a larger M_1 would result in more nodes survived in the first stage, more trees to search in the second stage, and consequently, higher complexity; similarly, a larger M_2 would incur 'longer' trees in the second stage and consequently higher complexity. From now on, taking this observation into account, let us focus on PHD which partitions the tree into two stages with equal number of layers. It should be mentioned, however, that values of M_1 other than $M/2$ might be more beneficial depending on specific applications.

4.2.3 Complexity comparisons of various decoding schemes

Figure 5(a) shows the BER performance of BSIDE, LSD, PHD, and SD with two transmit and two receive antennas when the modulation order L varies. It is clearly confirmed that BSIDE, LSD, PHD, and SD all possess the ML BER performance. The average numbers of multiplications of BSIDE, LSD, PHD, and SD in three modulation schemes (4-, 16-, and 64-QAM) are shown in Figures 5(b)-(d). In these figures, it is clearly observed that PHD generally outperforms LSD and SD in most cases, and also outperforms BSIDE when the value of L is large. Furthermore, it is observed that the gain in the computational complexity of PHD over those of other schemes is generally more substantial when the SNR is low and the number of antennas is large. Similar observations can be made in Figure 6, obtained when the numbers of the transmit and receive antennas are four.



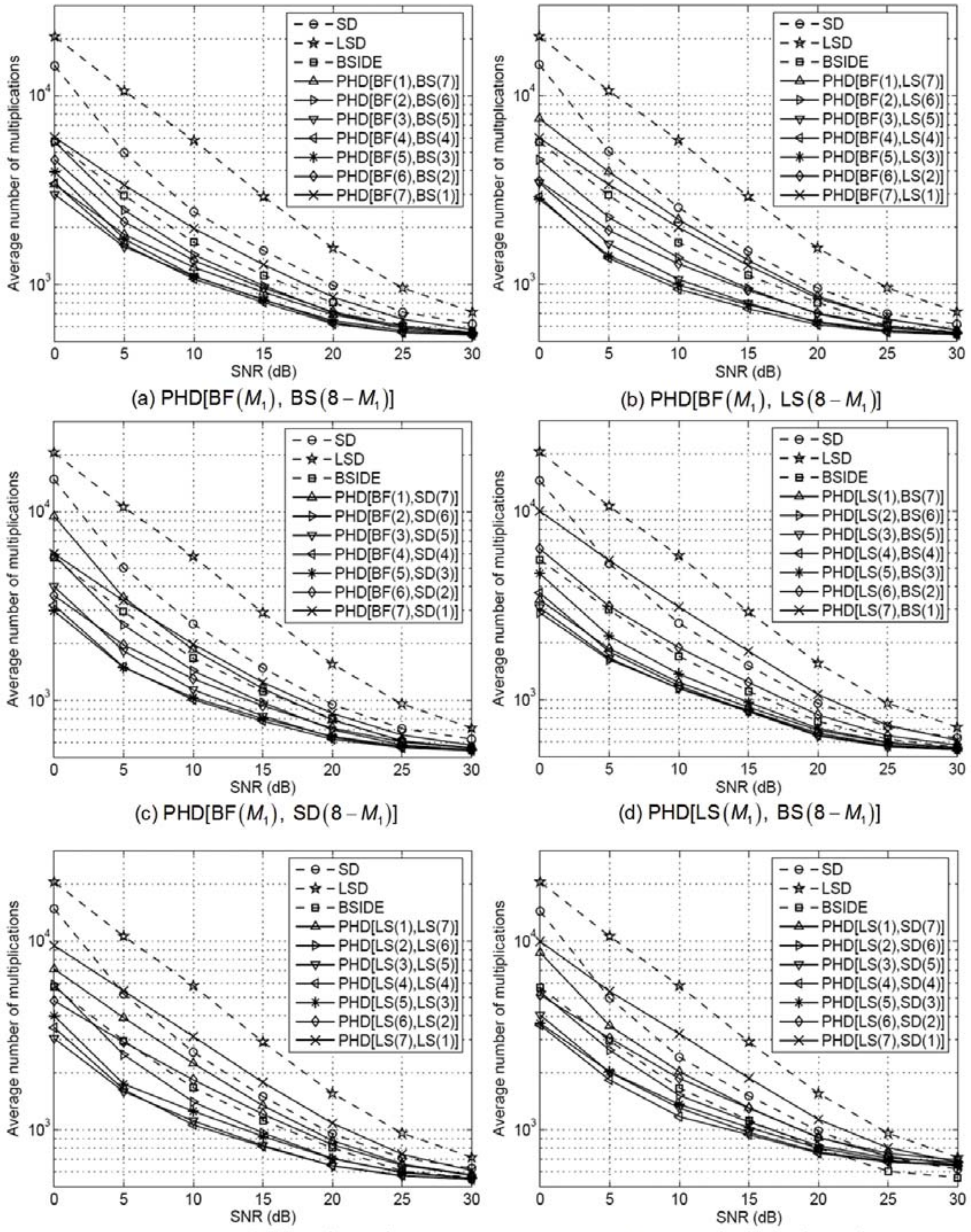
(a) LS and SD

(b) LS and BS

(c) BF and LS

(d) BF and BS

Figure 3. Average NOM of PHD for various choices of stages with search methods (a) LS and SD, (b) LS and BS, (c) BF and LS, and (d) BF and BS when $L = 4$ and $N_t = N_r = 4$.



(a) PHD[BF(M_1), BS$(8-M_1)$]

(b) PHD[BF(M_1), LS$(8-M_1)$]

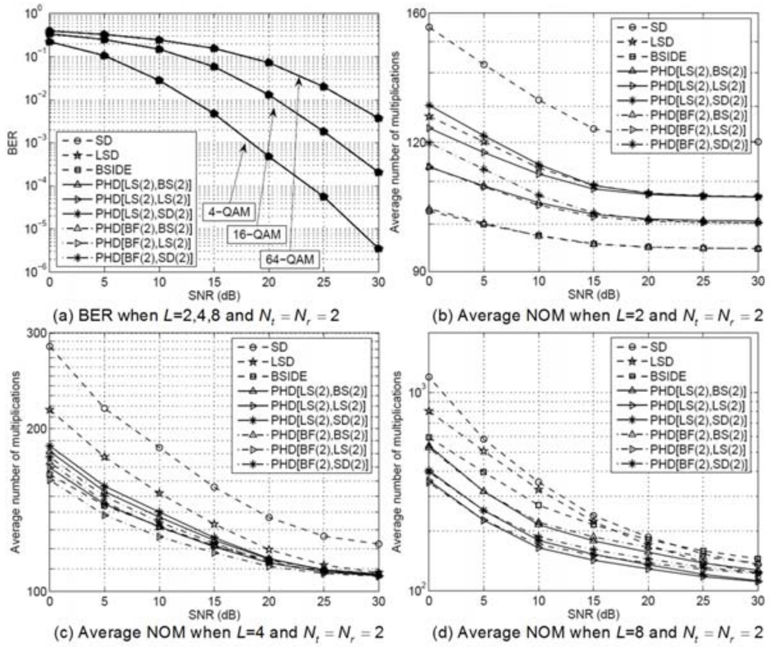
(c) PHD[BF(M_1), SD$(8-M_1)$]

(d) PHD[LS(M_1), BS$(8-M_1)$]

(e) PHD[LS(M_1), LS$(8-M_1)$]

(f) PHD[LS(M_1), SD$(8-M_1)$]

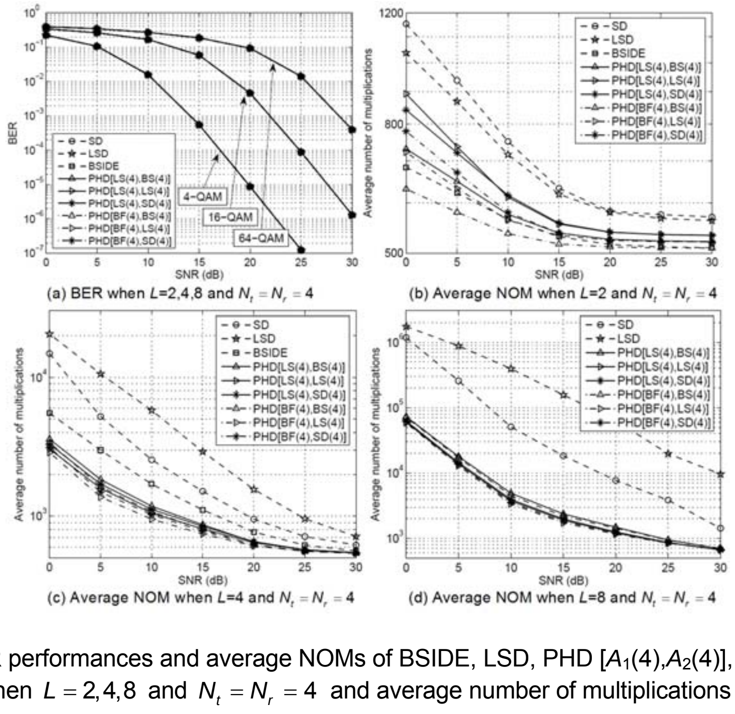
Figure 4. Average NOMs of proposed PHD schemes when $L = 4,\ N_t = N_r = 4,$ and the value of M_1 varies.



(a) BER when L=2,4,8 and $N_t = N_r = 2$ (b) Average NOM when L=2 and $N_t = N_r = 2$

(c) Average NOM when L=4 and $N_t = N_r = 2$ (d) Average NOM when L=8 and $N_t = N_r = 2$

Figure 5. BER performances and average NOMs of BSIDE, LSD, PHD [A_1(2),A_2(2)], and SD: BER performances (a) when $L = 2,4,8$ and $N_t = N_r = 2$ and average number of multiplications (b) when $L = 2$ and $N_t = N_r = 2$, (c) when $L = 4$ and $N_t = N_r = 2$, and (d) when $L = 8$ and $N_t = N_r = 2$.



(a) BER when L=2,4,8 and $N_t = N_r = 4$ (b) Average NOM when L=2 and $N_t = N_r = 4$

(c) Average NOM when L=4 and $N_t = N_r = 4$ (d) Average NOM when L=8 and $N_t = N_r = 4$

Figure 6. BER performances and average NOMs of BSIDE, LSD, PHD [A_1(4),A_2(4)], and SD: BER performances (a) when $L = 2,4,8$ and $N_t = N_r = 4$ and average number of multiplications (b) when $L = 2$ and $N_t = N_r = 4$, (c) when $L = 4$ and $N_t = N_r = 4$, and (d) when $L = 8$ and $N_t = N_r = 4$.

4.2.4 On the choice of search method in PHD with two stages

Based on the observations from the simulation results in the figures considered so far, we can make the following observations. (1) For a fixed search method in the second stage, 'BF' in the first stage produces lower computational complexity than 'LS' when $N_t = N_r = 2,4$ and $L = 2,4$: when $L = 8$, we can make the same observations at lower SNR, but 'LS' becomes the preferred choice as the SNR gets higher. (2) For a fixed search method in the first stage, 'BS' in the second stage would be the first choice when $L = 2$, and 'LS' would become our preference for the search method in the second stage as L gets larger. Such a dependence of the computational complexity of PHD on search methods is a natural consequence of the intrinsic of the depth- and breadth-first algorithms. We have summarized these observations in a somewhat quantitative manner in Tables 1 and 2.

	4-, 16-QAM	64-QAM
$N_t = N_r = 2$	BF	BF (if SNR < 5 dB), LS (if SNR > 5 dB)
$N_t = N_r = 4$	BF	BF (if SNR < 28 dB), LS (if SNR > 28 dB)

Table 1. Preferred search method in the first stage of PHD.

	4-QAM	16-QAM	64-QAM
$N_t = N_r = 2$	BS > LS ≥ SD	LS > BS ≥ SD	LS ≥ SD > BS
$N_t = N_r = 4$	BS > LS ≥ SD	LS ≥ SD ≥ BS	LS ≥ SD ≥ BS

Table 2. Preferred search methods in the second stage of PHD ($\alpha > \beta$ and $\alpha \geq \beta$ signifies that α is 'preferred' and 'preferred slightly' to β, respectively).

In passing, we would also like to mention that, if a common search method is employed in the second stage, the numbers of multiplications from various configurations of PHD are almost identical irrespective of the search methods in the first stage when the number of antennas and/or the size of signal constellation are large.

5. Concluding remark

For applications in MIMO systems, we have proposed a novel ML decoding scheme called PHD, which provides significant gain in the computational complexity compared with conventional ML decoding schemes. The proposed decoding scheme partitions the tree into several stages, in each of which a depth- and breadth-first search method is employed appropriately. The partitioning of a tree proposed in this paper plays a key role in the proposed decoding scheme, allowing us to appropriately and fully make use of the advantages of several search methods. The proposed decoding scheme has flexibility in choosing the number of stages, the number of layers in each of the stages, and the search method in each of the stages.

From simulation results, it is confirmed that the proposed decoding scheme with proper configurations has considerably lower computational complexity than conventional ML decoders while maintaining the optimal BER performance. In addition, it is observed that the computational complexity gain of the proposed decoding scheme increases as the SNR gets low and the number of antennas increases.

Acknowledgements

This research was supported by the National Research Foundation (NRF) of Korea under Grants 2012-0005066 and 2012R1A2A2A01045887 with funding from the Ministry of Education, Science and Technology (MEST), Korea, by the Information Technology Research Center (ITRC) program of the National IT Industry Promotion Agency under Grants NIPA-2012-H0301-12-1005 and NIPA-2012-H0301-12-2005 with funding from the Ministry of Knowledge Economy (MKE), Korea, and by National GNSS Research Center program of Defense Acquisition Program Administration and Agency for Defense Development.

References

[1] M. Bazdresch et al., "A family of hybrid space-time codes for MIMO wireless communications," Journal of Applied Research and Technology, vol. 10, no. 2, pp. 122-142, 2012.

[2] H.-J. Kwak and G.-T. Park, "Study on the mobility of service robots," International Journal of Engineering and Technology Innovation, vol. 2, no. 2, pp. 13-28, 2012.

[3] L. Babai, “On lovàsz’ lattice reduction and the nearest lattice point problem,” Combinatorica, vol. 6, no. 1, pp. 1-13, 1986.

[4] B. Hochwald and S. T. Brink, “Achieving near capacity on a multiple antenna channel,” IEEE Trans. Commun., vol. 51, no. 3, pp. 389-399, 2003.

[5] J. Jaldén and B. Ottersten, “On the complexity of sphere decoding in digital communications,” IEEE Trans. Sig. Process., vol. 53, no. 4, pp. 1474-1484, 2005.

[6] H. G. Kang et al., “Breadth-first signal decoder (BSIDE): A novel maximum likelihood scheme for multi-input multi-output systems,” IEEE Trans. Vehic. Technol., vol. 57, no. 2, pp. 1576-1584, 2008.

[7] L. Liu et al., “Combined maximum likelihood and SIC detection algorithm for MIMO system,” in Int. Conf. Computer Design and Applications (ICCDA), Qinhuangdao, China, 2010, pp. 561-564.

[8] L. Liu et al., “Improved detection algorithm for MIMO wireless communication system based on chase detector,” Lecture Notes in Computer Science, vol. 7473, pp. 196-202, 2012.

[9] M. O. Damen et al., “On maximum-likelihood detection and the search for the closest lattice point,” IEEE Trans. Inform. Theory, vol. 49, no. 10, pp. 2389-2402, 2003.

[10] A. D. Murugan et al., “A unified framework for tree search decoding: Rediscovering the sequential decoder,” IEEE Trans. Inform. Theory, vol. 52, no. 3, pp. 933-953, 2006.

[11] G. Valiente, “Algorithms on Trees and Graphs,” Berlin, Germany: Springer, 2002.

[12] Z. Guo and P. Nilsson, “Algorithm and implementation of the K-best sphere decoding for MIMO detection,” IEEE J. Sel. Areas Commun., vol. 24, no. 3, pp. 491-503, 2006.
[13] G. H. Golub and C. F. Van Loan, “Matrix Computations,” Baltimore, MD: Johns Hopkins University Press, 1996.

[14] D. Vybiral et al., “Devices for position detection,” Journal of Vibroengineering, vol. 13, no. 3, pp. 531-535, 2011.

[15] S. Ginzburg et al., “Design and implementation of an indoor localization system for the omnibot omni- directional platform,” Trans. Canadian Society for Mechanical Engineering, vol. 33, no. 4, pp. 715-729, 2011.