Towards the Performance of ML and the Complexity of MMSE - A Hybrid Approach

Byonghyo Shim  
EECS Dept., Korea University  
Seoul, Korea, 136-701  
Email: bshim@korea.ac.kr

Jun Won Choi  
ECE Dept., University of Illinois  
Urbana, IL, 61801  
Email: jwchoi@uiuc.edu

Insung Kang  
Qualcomm Inc.  
5775 Morehouse Dr., San Diego, CA  
Email: insungk@qualcomm.com

Abstract—In this paper, we present a near ML-achieving sphere search technique that reduces the number of search operations significantly over existing sphere decoding (SD) algorithms. While the SD algorithm relies only on causal symbols in evaluating path metric, proposed method accounts for the contribution of non-causal symbols with the aid of per-path minimum mean square error (MMSE) symbol estimation. The ML and MMSE combined cost metric results in the tight necessary condition for sphere decision and hence expedites the pruning of subtrees unlikely to be survived. From the simulations performed over multi-input multi-output (MIMO) wireless channels, it is shown that the computational complexity of the proposed approach is substantially smaller than the existing SD algorithms while providing negligible performance loss.

I. INTRODUCTION

Maximum likelihood (ML) detection of vector with finite alphabet symbols requires a search for the entire symbol combinations. For problems having a nice matrix structure such as circular or Toeplitz, efficient algorithms such as eigenvalue decomposition or Viterbi decoding can be applied, and hence the solution is found with a reasonable complexity expressed as $O(n^p)$ where $p$ is positive small integer. However, when no usable feature is available, the detection problem becomes NP-hard.

Recently, this problem has been revisited owing to a low complexity decoding method proposed by Fincke and Pohst [1], [2] is summarized as Sphere Decoding (SD) algorithm. The SD algorithm performs a constrained lattice point search using a hypersphere with radius $c_0$, and consequently, the number of points visited by the search can be reduced significantly over an exhaustive search. Due to the computation savings, the SD algorithm has received considerable attention as a detection scheme for multi-input multi-output (MIMO) wireless systems [6].

Although the sphere search guarantees an exact ML-solution, still a considerable amount of computation is required, which limits the application of SD algorithm, especially in the real-time application [9]. Previous approaches relaxing this strict condition to reduce the computational complexity include the statistical pruning by radius selection [11] and the estimation of the unvisited node in a probabilistic sense [12], [13]. Our method is distinct from these approaches in the sense that the contribution of unvisited nodes is computed via per path minimum mean square error (MMSE) symbol estimation and then added to the cost metric. The resulting cost metric provides a tight bound of the SD search, leading to the reduction in the number of nodes visited. Furthermore, we propose a radius control scheme that avoids the scenario to select loose initial radius as well as the search failure. The simulations performed over the MIMO channels demonstrate that the proposed method provides significant reduction in complexity while maintaining near-ML performance.

This paper is organized as follows. In Section II, we briefly review the SD algorithm based ML-detection and MMSE-estimation method. In Section III, we discuss the proposed method that tightens the sphere decision by the modification of the cost metric and the sphere radius. Section IV presents simulation results for MIMO systems and Section V concludes this paper.

II. ML-DETECTION AND MMSE-ESTIMATION

A. ML-detection via Sphere Search

Fincke and Pohst algorithm [1], [2] is summarized as follows: Let $c_0$ be the radius square of an $n$-dimensional hypersphere $S(r, \sqrt{c_0})$ centered at $r$. The ML solution is a nearest lattice point to an observation $r$. Instead of searching all lattice points in $\Lambda$, Fincke and Pohst algorithm searches the lattice points $Hs$ inside of the sphere, i.e., $S(r, \sqrt{c_0}) \cap \Lambda$, where $H$ is a $n \times m$ system matrix and $s$ is an $m \times 1$ finite alphabet symbol vector. This necessary condition for searching the ML solution is

$$c_0 \geq \| r - Hs \|^2. \quad (1)$$

In order to make the structure easy for searching, QR-decomposition of $H$ is performed as

$$H = [Q \ U] \begin{bmatrix} R \ 0 \end{bmatrix} \quad (2)$$

where $R$ is an $m \times m$ upper triangular matrix with positive diagonal elements, $0$ is an $(n-m) \times m$ zero matrix, and $Q$ and $U$ are $n \times m$ and $n \times (n-m)$ unitary matrices. Substituting (2) in (1), we have

$$d_0 \geq \| y - Rs \|^2. \quad (3)$$
where \( y = Q^T r \) and \( d_0 = c_0 - ||U^T r||^2 \). We denote \( \sqrt{d_0} \) as a modified sphere radius and \( ||y - Rs||^2 \) as a cost metric. Since \( R \) is an upper triangular matrix, \( y - Rs \) has a structure enabling the sequential calculation of path metrics and (3) becomes

\[
d_0 \geq \sum_{j=1}^{m} (y_j - \sum_{k=j}^{m} r_{j,k} s_k)^2
\]

\[
= (y_m - r_{m,m} s_m)^2 + (y_{m-1} - \sum_{k=m-1}^{m} r_{m-1,k} s_k)^2 + \cdots .
\]

Borrowing the parlance of the convolutional decoding, each term of the righthand side corresponds to a branch metric. Since it is natural to start the search from the bottom to the top layer, with a reference of the bottom layer as the first layer, the recursive relationship becomes

\[
P_m^m = B_m \quad (5)
\]

\[
P_k^m = P_{k+1}^m + B_k, \quad k = m - 1, \cdots , 1 \quad (6)
\]

\[
B_k = (y_k - \sum_{j=k}^{m} r_{k,j} s_j)^2 \quad (7)
\]

where \( P_k^m \) is the \((m - k + 1)-th \) layer path metric and \( B_k \) is the \((m - k + 1)-th \) layer branch metric.

For the first layer (the bottom layer), the necessary condition for \( Rs \) being inside the new sphere, \( S(y, \sqrt{d_0}) \), is

\[
P_m^m = (y_m - r_{m,m} s_m)^2 \leq d_0. \quad (8)
\]

In general, the condition for the \((m - k + 1)-th \) layer is \( P_k^m = P_{k+1}^m + B_k(s_k^m) \leq d_0 \). Using (7), we further have

\[
P_k^m + (y_k - \sum_{j=k+1}^{m} r_{k,j} s_j - r_{k,k} s_k)^2 \leq d_0. \quad (9)
\]

In the sequel, we refer this comparison as a sphere decision.

\section*{B. MMSE-estimation}

In a real system that the observation \( y \) is modeled as

\[
y = Hx + n, \quad (10)
\]

where \( x \) is the transmitted signal vector with zero mean and covariance \( \sigma^2_x I \) and \( n \) is a Gaussian noise \( \mathcal{N}(0, \sigma^2_n I) \), the linear MMSE estimator of \( x \) is given by [15]

\[
\hat{x} = \sigma^2_x H^T (\sigma^2_x H H^T + \sigma^2_n I)^{-1} y. \quad (11)
\]

In many cases, an alternative form obtained by the matrix inversion lemma might be computationally better

\[
\hat{x} = \sigma^2_x \left( H^T H + \frac{\sigma^2_n}{\sigma^2_x} I \right)^{-1} H^T y \quad (12)
\]

Since the MMSE-estimation is cost-efficient, it may be used to estimate symbols of unvisited nodes in the sphere search.

\section*{III. MIXTURE OF ML AND MMSE}

In this section, we first discuss the problems of sphere search in the complexity perspective and then present the proposed tightening method. Tightening of the sphere decision is performed on the cost metric \( ||y - Rs||^2 \) as well as the sphere radius \( d_0 \). In order to distinguish the symbol vectors found by sphere search and MMSE estimation, we use the notation \( s \) and \( \hat{s} \), respectively. We denote \( \bar{s} \) for the mixture of the detected and estimated symbols.

\subsection*{A. Problems of Sphere Decoding}

The sphere decoding can be well explained as a tree search algorithm and therefore the cost metric increases monotonically from the root (top) to the bottom node [6], [13]. As described in the previous section, the condition to prune the branches at node \( k \) is

\[
P_k^m \leq d_0.
\]

In other words, the sphere decision at the node \( k \) is

\[
B_k + B_{k+1} + \cdots + B_m \leq d_0. \quad (13)
\]

Recall that the decision based on the sphere constraint should be

\[
||y - Rs||^2 = B_1 + \cdots + B_k + \cdots + B_m \leq d_0. \quad (14)
\]

Since the branch metrics of unvisited node \( k - 1, \cdots , 1 \) are unavailable and thus not being used, the sphere condition in (13) is loose inevitably. Even though the path metric is close to but smaller than \( d_0 \) and thus we are sure that the path would be discarded from the search soon, we cannot prune the path just because the path metric is still smaller than \( d_0 \). In general, the problems of the SD algorithm can be summarized as follows:

- **Loose comparison:** Since only partial nodes \((m, \cdots , k + 1)\) are being used, the pruning at the early node is quite ineffective (e.g., \( B_m \leq d_0 \) at the root node). As the search moves on to the bottom node, the sphere comparison becomes tight.

- **Inefficient use of information:** Due to the causality in the search, observations of the unvisited node \((y_k, \cdots , y_2, y_1)\) are not exploited in the sphere decision of node \( k \).

Although the SD algorithm finds an exact ML solution, due to these reasons, a considerable amount of computation is wasted. In the next subsections, we propose a tightening scheme for cost metric and sphere radius, which will result in significant computation savings.

\subsection*{B. MMSE-estimation of Unvisited Subtree}

With a reference to the currently visiting node \( k \), the QR-transformed vector \( y = Rs + n \) can be partitioned into

\[
y = R_l s_u + R_r s_d + n \quad (15)
\]
Clearly, the estimation of an alternative form

where \( s_d = [s_k \cdots s_{m-1} s_m]^T \), \( s_u = [s_1 s_2 \cdots s_{k-1}]^T \), and

\[ R = [R_u \ R_r]. \]

Furthermore, with \( y = \begin{bmatrix} y_u \\ y_d \end{bmatrix} \) and \( R_r = \begin{bmatrix} R_{ru} \\ R_{rd} \end{bmatrix} \), we have

\[
y_u = R_{ru} s_u + R_{rd} s_d + n_u \tag{16}
\]

\[
y_d = R_{rd} s_d + n_d \tag{17}
\]

Refer the Fig. 1 for detail. Since the node \( k \) is currently visited, we have symbols \( \hat{s}_d \) determined by the path up to current node. The corresponding path metric from the node \( m \) to \( k \) is

\[
P_{km}^m = \| y_d - R_{rd} \hat{s}_d \|^2. \tag{18}
\]

\( \hat{s}_d \) will be helpful to estimate the symbols of the unvisited node. Suppose that the detected symbols are accurate (\( \hat{s}_d = s_d \)), then the cancellation of the reconstructed signal \( R_{ru} \hat{s}_d \) would enhance the system SNR in a subsequent detection

\[
y_u' = y_u - R_{ru} \hat{s}_d = R_{lu} s_u + R_{ru} (s_d - \hat{s}_d) + n_u. \tag{19}
\]

Clearly, the estimation of \( s_u \) would be more accurate than that without cancellation. On the contrary, if the decision is incorrect (\( \hat{s}_d \neq s_d \)), the cancellation would deteriorate the SNR

\[
y_u' = y_u - R_{ru} \hat{s}_d = R_{lu} s_u + R_{ru} (s_d - \hat{s}_d) + n_u. \tag{20}
\]

so that the estimation quality would be poor. In any case, given the observation \( y_u' \), the MMSE estimate of \( s_u \) is

\[
\hat{s}_u = R_{lu}^T (R_{lu} R_{lu}^T + \sigma_n^2 I)^{-1} y_u'. \tag{21}
\]

where \( \sigma_n^2 = E[n_u n_u^T] \). As described in II.B, it is better to use an alternative form

\[
\hat{s}_u = (R_{lu}^T R_{lu} + \sigma_n^2 I)^{-1} R_{lu}^T y_u'. \tag{22}
\]

The reason of using \( \sigma_n^2 \) as a noise power is to match the best scenario in (20). Suppose the noise power is not properly matched as in (20), the reconstruction of \( R_{lu} \hat{s}_u \) will rather increase the cost metric so that it will expedite the pruning of the incorrect path. Note that this MMSE estimation for the unvisited part of the tree is performed for every node visited, i.e., per-path processing, and the paths having same parents are subject to the same MMSE estimator. Thus, the calculation of estimator coefficients in (22) can be done before the tree search and shared in a processing block as long as a channel state remains unchanged. For block fading scenarios, MMSE computation can be done only per block and hence the overhead of MMSE estimation will be moderate.

**C. Further Reduction on Complexity via Zero-Forcing (ZF)**

By employing a ZF estimation, further reduction of complexity can be achieved. Since ZF estimator assumes \( n_u = 0 \), it shows best performance for high SNR regime. The ZF estimate of \( s_u \) is

\[
\hat{s}_u = (R_{lu}^T R_{lu})^{-1} R_{lu}^T y_u'. \tag{23}
\]

Rewriting (23), we have

\[
A \hat{s}_u = b \tag{24}
\]

where \( A = (R_{lu}^T R_{lu}) \) and \( b = R_{lu}^T y_u' \). It is interesting to note that \( R_{lu} \) is a cholesky decomposition of \( A \) [16]. By denoting \( \hat{x}_u = R_{lu} \hat{s}_u \), (24) becomes

\[
R_{lu}^T \hat{x}_u = b \tag{25}
\]

Since \( R_{lu}^T \) is a lower triangular matrix, \( \hat{x}_u \) can be obtained by successive substitution operations from the top to the bottom row. Once \( \hat{x}_u \) is obtained, we obtain \( \hat{s}_u \) by solving

\[
R_{lu} \hat{s}_u = \hat{x}_u. \tag{26}
\]

Similar procedure can be applied by successive substitution operations from the bottom to the top row. In summary, by the cascade of successive substitutions, \( \hat{s}_u \) can be obtained without painful matrix inversion. Since the successive substitution operation of the triangular matrix has a linear complexity \( O(n) \), this case is much better in complexity than that of fully populated matrix expressed as \( O(n^3) \) [16]. It is worth pointing out that noise amplification of ZF algorithm in low SNR scenario rather helps the rejection of incorrect path by increasing the path metric (refer the scenario in (20)).

**D. Cost Metric Tightening**

Using (15), (16), and (17), the actual ML cost metric becomes

\[
\| y - R s \|^2 = \| y_u - R_{lu} s_u - R_{ru} s_d \|^2 + \| y_d - R_{rd} s_d \|^2. \tag{27}
\]

Since \( \hat{s}_d \) is determined from sphere search and \( \hat{s}_u \) can also be computed by the MMSE-estimation in III-B, the ML cost metric becomes

\[
\| y - R s \|^2 = \hat{P}_1^{k-1} + P_k^m \tag{27}
\]

where \( \hat{P}_1^{k-1} = \| y_u - R_{lu} \hat{s}_u - R_{ru} \hat{s}_d \|^2 \). Since \( \hat{P}_1^{k-1} \) is non-negative, using \( P_1^{k-1} + P_k^m \) as a sphere metric instead of \( P_k^m \) tightens the sphere decision.
IV. SIMULATION AND DISCUSSIONS

In this section, we observe the performance and complexity of the proposed method over the MMSE equalizer, SE enumeration based SD algorithm, as well as the near-ML approaches including increasing radii algorithm (IRA) [11] and probabilistic tree pruning SD (PTP-SD) [13]. The simulation setup is based on $L^2$-QAM transmission over MIMO systems in quasi-static Rayleigh fading channel, where $H$ matrix is modeled as independent Gaussian random variables. We use $6 \times 6$ MIMO system and the levels of QAM are $L = 8$ (64-QAM). The PTP-SD and the reference SD algorithm in our study employs the depth-first search and the SE enumeration. Hence, we set the initial radius to $d_0 = \infty$ and update it whenever a new candidate $s$ is found. In addition, we set the pruning probability $\epsilon = 0.1$ for the PTP-SD and $P_t = 0.9$ for the proposed algorithm. As a metric for measuring performance and complexity, we employ a symbol error rate (SER) and an average number of nodes visited. We set our target SER to $10^{-2}$ and run at least 20,000 channel realizations for each SNR point. Fig. 2 compares the SER and the average numbers of nodes visited per channel use for IRA, PTP-SD, reference SD, and the proposed algorithm. While the performance gap from the exact ML detector is negligible for all methods under consideration, there is clear distinction in complexity among receiver algorithms. The IRA detector in tends to show high complexity for low SNR region while the complexity of PTP-SD does not decay fast for high SNR regime. On the contrary, the proposed detection algorithm provides the substantial reduction in complexity for the entire range of interest.

V. CONCLUSION

In this paper, we have proposed a modification of sphere search algorithm that tightens a sphere decision. Since the sphere decision at the early node of the tree is loose, we add the contribution of the unvisited node into the cost metric and then further shrink the hypersphere being searched by tightening the sphere radius. From the simulation results, we observed that the proposed method provides considerable savings in computational complexities while achieving negligible performance loss over the strict ML-detection. While our simulation study is limited to the MIMO communication systems in this paper, the proposed method can be naturally

\[
\text{Inputs: } d_0 = c_0 - ||Q^T y||^2, y', \text{ and } R
\]

\[
\text{Output: } s
\]

\[
\text{Variables: } i \text{ denote layer being examined, } k \text{ denote branch index in the layer.}
\]

\[
\text{Step 1: } \text{Set } i := m, d := d_0, \text{ and } P := 0.
\]

\[
\text{Go to next step.}
\]

\[
\text{Step 2: } \text{If } d < P, \text{ then goto step 4.}
\]

\[
\text{Else do SE-ordering, set } k := 0 \text{ and goto step 3.}
\]

\[
\text{Step 3: } k := k + 1. \text{ If } k > N_s, \text{ then goto step 4.}
\]

\[
\text{Else goto step 5.}
\]

\[
\text{Step 4: } \text{If } i = m, \text{ then terminate algorithm.}
\]

\[
\text{Else } i := i + 1 \text{ and goto step 3.}
\]

\[
\text{Step 5: } \text{If } i = 1, \text{ then compute } P_i^m \text{ and goto next step.}
\]

\[
\text{Else } i := i - 1, \text{ compute } P_i^m \text{ and } P_i^{m-1}.
\]

\[
\text{Generate cost metric } P = P_i^m + P_i^{m-1} \text{ and goto step 2.}
\]

\[
\text{Step 6: } \text{Save } s. \text{ Update } d := P_i^m, i := i + 1,
\]

\[
\text{and goto step 3.}
\]
extended into more complicated systems such as CDMA downlink, frequency selective (ISI) communications, and soft-output applications.

ACKNOWLEDGMENT

This work was supported by the research grant from Qual-comm Inc., Korea Univ. (K0800061), and second BK 21 project.

REFERENCES