A LOW-COMPLEXITY NEAR-ML DECODING TECHNIQUE VIA REDUCED DIMENSION LIST STACK ALGORITHM

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

*University of Illinois at Urbana-Champaign
Coordinated Science Laboratory
1308 West Main St. Urbana, IL 61801, USA
**Korea University
***Seoul National University

ABSTRACT
In this paper, we propose a near maximum likelihood (ML) decoding technique, which reduces the computational complexity of the exact ML decoding algorithm. The computations needed for the tree search in the ML decoding is simplified by reducing the dimension of the search space prior to the tree search. In order to compensate performance loss due to the dimension reduction, a list stack algorithm (LSA) is considered, which produces a list of the top \( K \) closest points. The combination of both approaches, called reduced dimension list stack algorithm (RD-LSA), is shown to provide flexibility and offers a performance-complexity trade-off. Simulations performed for V-BLAST transmission demonstrate that significant complexity reduction can be achieved compared to the sphere decoding algorithm (SDA) while keeping the performance loss below an acceptable level.

Index Terms— Maximum likelihood, Dimension reduction, MIMO, Tree search, Sphere decoding

1. INTRODUCTION
In many communication systems, the relationship between the transmitted symbol vector and received signal vector is described by
\[
y = Hx + w, \tag{1}
\]
where \( x \) is \( n_t \)-dimensional vector whose entries are chosen from finite symbol alphabet, \( y \) and \( w \) are \( n_r \)-dimensional received signal and noise vectors, respectively, \( H \) is \( n_r \times n_t \) channel matrix. Multiple-input-multiple-output (MIMO) systems can be described by the above model by assuming that \( H \) is the complex channel matrix.

In detecting the transmitted point \( x \), the maximum likelihood (ML) decoding problem is formulated as
\[
\hat{x}^{\text{ml}} = \arg \min_{x \in \mathcal{F}^{n_t}} \| y - Hx \|^2, \tag{2}
\]
where \( \hat{x}^{\text{ml}} \) is an ML solution and \( \mathcal{F} \) is the given alphabet. In this case, ML problem equals to a closest lattice point search (CLPS) problem subject to the finite alphabet constraint [2]. Although the ML decoding provides the significant decoding performance gain compared to the sub-optimal detectors such as V-BLAST detector [5], it demands high computational complexity due to the exponential growth of search space (complexity \( \propto O(|\mathcal{F}|^{n_t}) \)).

Recently, various search algorithms have been proposed to lower the complexity in ML decoding. Popular example includes sphere-constrained tree search algorithm so called sphere decoding algorithm (SDA) [1] and its variation such as the Schnorr-Euchner enumeration based search [2], the probabilistic ordered search [6], the sequential Fano decoder [3] and the stack algorithm [4].

This paper is motivated by the fact that the complexity of the tree search can be reduced significantly by decreasing the dimension of the search space. This dimension reduction, achieved by minimum mean square error (MMSE) estimation technique in the preprocessing stage (see Fig. 1), allows the tree search algorithm to explore the reduced depth (stage) of tree, leading to substantial complexity reduction. In order to compensate the performance loss due to the dimension reduction, we employ 1) list-ML decoding technique that finds multiple candidates instead of single output and 2) additional postprocessing stage for re-estimation of candidates found in the list tree search. In doing so, the proposed algorithm, called reduced dimension list stack algorithm (RD-LSA), reduces the computational complexity significantly over SDA while maintaining near ML performance.

This paper is organized as follows. After brief system description in Section 2, we present the proposed RD-LSA al-
2. SYSTEM DESCRIPTION

For data transmission, we assume *Mary-quadrature amplitude modulation* (M-QAM). It is straightforward to include other modulation schemes in our framework. From now on, we proceed our discussion mainly with a MIMO system with *nt* transmit antennas and *nr* receive antennas. It will be easily extended to other communication systems such as a lattice decoding and single-input-single-output (SISO) frequency selective channel equalization. We assume that a symbol being transmitted is normalized with the unit energy and the symbols are uncorrelated, i.e., $E[xx^H] = I_{nt}$. We assume a quasi-static Rayleigh fading channel. The noise vector, *w* is assumed to be i.i.d. circular symmetric Gaussian, and $E[ww^H] = \sigma_w^2 I_{nr}$. We assume that the receiver has exact knowledge of channel state, *H* and noise variance, $\sigma_w^2$. The signal power to noise power ratio (SNR) is defined as

$$\text{SNR} = 10 \log_{10} \frac{nt}{\sigma_w^2},$$

where total transmit power per channel use is used to define SNR.

3. REDUCED DIMENSION LIST STACK ALGORITHM (RD-LSA)

In this section, the proposed RD-LSA is derived and described.

3.1. Overview

We let $n_1 (< nt)$ be the dimension of the reduced system and let $n_2 = nt - n_1$. The received signal, *y* can be written

$$y = Hx + w = H_1x_1 + H_2x_2 + w,$$

where we let $H_1$ be the sub-matrix constructed by the $n_1$ columns of *H* and $H_2$ be the one constructed by the remaining $n_2$ columns of *H*. We let $x_1$ and $x_2$ be the column vectors associated with $H_1$ and $H_2$.

With this notation, the ML solution can be expressed as

$$\hat{x}_1^{ml} = \arg \min_{x_1 \in F^{n_1}, x_2 \in F^{n_2}} \| y - H_2x_2 - H_1x_1 \|^2.$$

One can easily show that $\hat{x}_1^{ml}$ in (5) can be obtained by two-stage processing given by

$$\hat{x}_1^{ml} = \arg \min_{x_1 \in F^{n_1}} D(y; x_1)$$

$$D(y; x_1) = \min_{x_2 \in F^{n_2}} \| y - H_2x_2 - H_1x_1 \|^2.$$

The ML solution of $x_1$ can be obtained by minimizing the distance between *y* and $H_2x_2$ for every realization of $x_1$, and then finding the point minimizing the cost function $D(y; x_1)$ among all possible $x_1$. The complexity of finding an optimal $x_2$ for all realizations of $x_1$ is $M^{n_2}$. This job is computationally demanding for a large system with high-order modulation.

To alleviate the high complexity, $x_2$ is replaced by the estimate of $x_2$, which is derived given the observation *y* and tentative value of $x_1$, i.e., $\hat{x}_2 = f(y; x_1)$. Then, $D(y; x_1)$ can be approximated as

$$D(y; x_1) \approx \tilde{D}(y; x_1) = \| y - H_2f(y; x_1) - H_1x_1 \|^2.$$

Note that the operation of minimization over $x_2$ disappears from (6) and the corresponding approximated solution $\tilde{x}_1^{ml}$ becomes

$$\tilde{x}_1^{ml} = \arg \min_{x_1 \in F^{n_1}} \tilde{D}(y; x_1).$$

Clearly, the estimation quality of $\hat{x}_2$ would affect the performance of detecting $x_1$. In the sequel, we present the details of $x_2$ estimation, partitioning of $H_1$ and $H_2$, and the near-ML algorithm for obtaining $\tilde{x}_1^{ml}$.

3.2. Preprocessing

As a first step, we divide the symbol vector into two groups using a classification method such as the V-BLAST symbol ordering strategy [5] or probabilistic symbol ordering [6]. While symbols in the strong group (which leads to better performance when detected first) is selected as $x_1$ and rest of them is selected as $x_2$. The justification of this choice will be explained later.

As a low complexity estimator of $x_2$, the linear minimum mean square error (LMMSE) estimate suits this purpose well. The LMMSE estimate of $x_2$ minimizing the cost function $E[\| x_2 - \hat{x}_2 \|^2 | x_1]$ is given by

$$\hat{x}_2 = f(y; x_1) = F(y - H_1x_1),$$

where $F = (H_2^H H_2 + \sigma_w^2 I)^{-1} H_2^H$. We can assume the perfect cancellation of $x_1$ in obtaining $F$ since the search space $F^{n_1}$ in (8) would necessarily contain the exact solution of $x_1$. We can expect a good estimate of $x_2$ only for a true $x_1$.

From (7) and (9), $\tilde{D}(y; x_1)$ becomes

$$\tilde{D}(y; x_1) = \| y - H_2F(y - H_1x_1) - H_1x_1 \|^2 = \| Z(y - H_1x_1) \|^2,$$

where

$$Z = I - H_2F = \sigma_w^2 (H_2^H H_2 + \sigma_w^2 I)^{-1}.$$

From (8) and (10), it can be shown that $\tilde{x}_1^{ml}$ is given by

$$\tilde{x}_1^{ml} = \arg \min_{x_1 \in F^{n_1}} \| (Zy - ZH_1x_1) \|^2.$$
Note that although the use of LMMSE estimator may not be the best choice, the structure of LMMSE estimator allows the ML structure with the linearly transformed system \((ZH_1)\) and observation \((Zy)\).

It is worth to mention the role of matrix \(Z\) which can be seen a dimension reduction operator. The modified observation \(Zy\) can be written by

\[
Zy = ZH_1 x_1 + Z(H_2 x_2 + w). \tag{13}
\]

The second term in the right-side is viewed as the interference plus noise term in estimating \(x_1\) in the modified system. It can be easily shown that the matrix \(Z\) in (11) is the LMMSE estimator of \(w\) that minimizes the power,

\[
E \left( \|Z(H_2 x_2 + w) - w \|^2 \right). \tag{14}
\]

Hence, we can rewrite (13) by

\[
Zy = ZH_1 x_1 + w + e \tag{14}
\]

\[
\approx ZH_1 x_1 + w, \tag{15}
\]

where \(w + e = Z(H_2 x_2 + w)\). The LMMSE estimator, \(Z\) tries to retain the noise component \(w\) by suppressing the effect of the term \(H_2 x_2\). Hence, selecting weak group symbols as \(x_2\) is reasonable since the effect of \(x_2\) might be suppressed well and this choice leads to the system of strong symbols \(x_1\).

The approximation in (15) is reasonable since the impact of \(e\) (estimation error) is typically smaller than that of \(w\). Note also that the noise term, \(w + e\) in (14) is correlated and non-Gaussian, making the solution of (12) sub-optimal. In order to minimize the performance loss, we modify the ML detection algorithm such that multiple candidates are found.

### 3.3. Tree search step: list stack algorithm (LSA)

After taking the preprocessing operator \(Z\), the ML detection problem becomes

\[
\hat{x}_1^{m} = \arg \min_{x_1 \in \mathbb{F}^{n_1}} \| y' - G x_1 \|^2, \tag{16}
\]

where \(y' = Zy\) and \(G = ZH_1\). Hence, tree search algorithms can be applied to find \(x_1\) in (16). As mentioned, \(\hat{x}_1^{m}\) does not necessarily equal to the exact ML solution \(\hat{x}_1\) and thus leads to the performance loss. In order to overcome this loss, the proposed tree search step tries to find multiple points which are close to \(y\). One such example in the literature is the list SDA [7] which finds \(N\) lattice points satisfying the sphere constraint, \(\| z - G x_1 \|^2 < B^2\), where \(B\) is an adequately chosen sphere radius. The problem of list SDA is that it is hard to choose an appropriate \(B\) that contains lattice points close to but larger than \(N\) in number. If the lattice point inside the sphere is too large or too small, the computational burden of list SDA is considerable or performance degrades.

In order to lessen the computational complexity and make an algorithm irrespective of \(B\), we develop the list stack algorithm (LSA), which can limit the number of closest points found. The stack algorithm [4] is a tree search algorithms based on the best-first search strategy. It is shown in [3] that the best-first tree search generates the least number of nodes among all tree search algorithms. In searching a tree, the stack algorithm extends the node with minimum cost metric first. It stores all extended nodes and stops when it reaches the leaf node the first time, yielding the leaf node as a closest point. On the contrary, the LSA continues the node extension until it reaches the leaf level \(K\) times, thereby producing \(K\) closest points. The extension of a stack algorithm to list tree search is also found in [8] in the context of an iterative decoding.

To further reduce computation, we introduce a fast stopping criterion which stops the search early by monitoring the cost metrics of already found closest points. Let the cost metric of the first found closest path be denoted as \(C\). It is desirable to find only points with small cost (distance) metrics. Hence, \(C\) is compared with the subsequent path metrics which are evaluated after founding the first closest point. Since the path metric of LSA algorithm would increase as the search continues, the search is stopped if \(PM > mC\) or \(i > K_{\text{max}}\), where \(PM\) is a path metric, \(K_{\text{max}}\) is the maximum value of \(K\), and \(m\) is a carefully chosen parameter. This stopping criterion results in a varying size list of no more than \(K_{\text{max}}\).

### 3.4. Postprocessing step

First, the postprocessing step extends the candidates produced by the LSA to full dimension and construct the final candidate list. Let us denote the \(i\)th closest point for \(x_1\) found by the LSA as \(\hat{x}_1^i\). The MMSE-decision feedback (MMSE-DF) [5] is employed to find the estimate of \(x_2\) given \(y\) and each \(\hat{x}_1^i\), i.e.,

\[
\hat{x}_2^i = \text{slicer} \left( G \left( y - H_1 \hat{x}_1^i \right) - B \hat{x}_2^i \right) \tag{17}
\]

where “slicer” denotes a function which maps an input to the nearest constellation point. The feedforward filter \(G\) and feedback filter \(B\) are determined by minimizing the MSE between \(x_2\) and \(\hat{x}_2^i\). To enable a causal feed-back operation, \(B\) should be strictly lower triangular. Such filtering can be efficiently implemented via a successive feedback cancellation mechanism [5].

Next, with the estimate, \(\hat{x}_2^i\) associated with each \(\hat{x}_1^i\), we can construct the search list, denoted \(S\)

\[
S = \left\{ \left[ \hat{x}_1^1, \hat{x}_2^1 \right], \ldots, \left[ \hat{x}_1^K, \hat{x}_2^K \right] \right\}. \tag{18}
\]

Finally, the near ML point, denoted \(\hat{x}_1^{m}\), is found by minimizing the ML distance over \(S\), i.e.,

\[
k_{\text{min}} = \arg \min_{1 \leq k \leq K} \| y - H_2 \hat{x}_2^k - H_1 \hat{x}_1^k \|^2 \tag{19}
\]

\[
\hat{x}_1^{m} = \begin{bmatrix} \hat{x}_1^{k_{\text{min}}} \\ \hat{x}_2^{k_{\text{min}}} \end{bmatrix}. \tag{20}
\]

This is the output of the RD-LSA.
4. SIMULATIONS

To perform Monte-Carlo simulations, $10^6$ information bits are generated. The information bits are mapped to 16-QAM symbols via Gray-mapping and are transmitted via V-BLAST architecture. We assume a block fading channel, in which the channel state is constant over 400 channel uses. The average number of float point operations (FLOPS) is used as a complexity measure. In the simulations, the following decoding algorithms are compared: a) the full dimension SDA (FD-SDA) [2], b) the proposed RD-LSA, and c) the V-BLAST detector [5]. Note that Algorithm II in [2] is chosen as FD-SDA. This FD-SDA guarantees exact ML decoding performance.

Fig 2 provides a plot of the bit error rate (BER) and average complexity between 13 dB and 24 dB SNR for the (6, 6) antenna configuration. We reduce the system dimension to $n_1 = 4$. We set the size of list $K$ to 4 and $n_2$ to 3 for the stopping criterion. As shown in the figure, the average complexity is reduced by around 40% at $10^{-2}$ BER and by 15% at $10^{-3}$ BER. On the contrary, the performance gap between FD-SDA and RD-LSA is less than 0.5 dB over the SNR range of interest. It was observed that the computation reduction increases significantly for larger problem size and higher modulation order, though not presented here due to space considerations.

5. CONCLUSIONS

In this paper, a dimension reduction technique for an ML tree search algorithm is studied to provide a low-complexity ML decoder. It is shown that this approach successfully reduces computational complexity, compared to the existing SDA decoder with acceptable performance loss in the application of 6-by-6 16-QAM V-BLAST communication.

6. REFERENCES


