International Symposium on Information Theory and its Applications, ISITA2004
Parma, Italy, October 10–13, 2004

Two Methods for Decreasing the Computational Complexity of the MIMO ML Decoder

Takayuki FUKATANI†, Ryutaroh MATSUMOTO‡, and Tomohiko UYEMATSU‡

†Tokuda Lab., Dept. of Computer Science
Tokyo Institute of Technology, 152-8552 Japan.
‡Dept. of Communications and Integrated Systems,
Tokyo Institute of Technology, 152-8552 Japan.
Email: ryutaroh@rmatsumoto.org, uyematsu@ieee.org

Abstract

We propose use of QR decomposition with sort and Dijkstra's algorithm for decreasing the computational complexity of the sphere decoder that is used for ML detection of signals on the multi-antenna fading channel. QR decomposition with sort decreases the complexity of search part of the decoder with small increase in the complexity required for preprocess part of the decoder. Dijkstra's algorithm decreases the complexity of searching part of the decoder with increase in the storage complexity. The computer simulation demonstrates that the complexity of the decoder is reduced by the proposed methods significantly.

1. Introduction

In recent years the use of multiple transmit and receive antennas over fading channels has attracted great attention, because use of multiple antennas enables very high spectral efficiency. In uncoded systems with many transmit antennas, we face with the huge number of possible transmitted signals, which makes the naive maximal likelihood (ML) decoding impractical. The decoding is usually done by suboptimal algorithms, such as the nulling and canceling algorithm [4]. However, recently an effective ML decoding algorithm was proposed as the sphere decoder [2, 7], and it is shown that the error rate of the ML decoder is significantly smaller than the nulling and canceling algorithm [2]. The drawback of the sphere decoder is that it is much slower than the nulling and canceling algorithm, and several researchers improved its efficiency, some of which will be reviewed later.

The sphere decoder can be divided in two parts. The first part computes the QR decomposition (or the Cholesky decomposition) of the fading matrix. The second part computes the ML estimate of transmitted signal from the received signal and the QR decomposition. We call the first part the preprocess part and the second part the search part.

The signal from multiple transmit antennas can be regarded as a vector. The traditional method of the search part determines components in the transmitted vector one by one. It is known that order of decisions on signal components has large impact on the computational complexity of the search part [3]. We propose the QR decomposition with sort as a method of the preprocess part. The proposed method determines an efficient order of decisions on signal components, and it significantly reduces the computational complexity of the search part with a little increase of the computational complexity in the preprocess part.

We will also show that the search part can be regarded as the shortest path problem in a weighted graph and use of Dijkstra' algorithm significantly reduces the computational complexity of the search part. We verify the reduction of computational complexity by computer simulation.

The QR decomposition with sort is modification only in the preprocess part and use of Dijkstra' algorithm is only in the search part. Thus these improvements are independent and can be used together or alone.

2. Brief review of the sphere decoder

Let $\vec{r} = H\vec{s} + \vec{n}$ be the standard lowpass equivalent description of frequency flat fading channel with t transmit antennas and r receive antennas, where $\vec{s} \in \mathbf{S}^t$ is the transmitted signal, $\vec{r}$ is the received signal, $\vec{n}$ is additive white Gaussian noise (AWGN), H is the fading matrix, $\mathbf{S}$ is the signal constellation. We assume $r \geq t$. Let $H = QR$ be a QR decomposition. Let $\vec{s'}$ be

Full version of this abstract will appear in IEICE Trans. Fundamentals.

an ML estimate of $\vec{s}$. Then

$$\begin{aligned} \|\vec{r} - H\vec{s'}\| &= \|\vec{r} - QR\vec{s'}\| \\ &= \|Q^*\vec{r} - Q^*QR\vec{s'}\| \\ &= \|Q^*\vec{r} - R\vec{s'}\| \end{aligned}$$

is of minimum value among all possible transmitted signals in $\mathbf{S}^t$. Let $R_{i,j}$ be the (i, j) entry of R, $\vec{s'} = (s'_1, \ldots, s'_t)$, and $Q^*\vec{r} = (q_1, \ldots, q_r)$.

We will informally review the original method in search part [7]. It tries to find a vector $\vec{s'} \in \mathbf{S}^t$ within a sphere with radius $\sqrt{C}$ and center $\vec{r}$, and considers the inequality

$$\|Q^*\vec{r} - R\vec{s'}\|^2 \leq C. \tag{1}$$

The t-th column in $Q^*\vec{r} - R\vec{s'}$ is $q_t - R_{t,t}s'_t$, and $|q_t - R_{t,t}s'_t|^2 + \sum_{i=t+1}^{r} |q_i|^2 \leq C$ is a necessary condition for Eq. (1). So we choose $\hat{s}_t \in \mathbf{S}$ with $|q_t - R_{t,t}\hat{s}_t|^2 \leq C$ as a candidate of t-th component in $\vec{s'}$. Considering $(t-1)$-th and t-th columns in $Q^*\vec{r} - R\vec{s'}$, we choose $\hat{s}_{t-1} \in \mathbf{S}$ with

$$\left\| \begin{pmatrix} q_{t-1} \\ q_t \end{pmatrix} - \begin{pmatrix} R_{t-1,t-1} & R_{t-1,t} \\ 0 & R_{t,t} \end{pmatrix} \begin{pmatrix} \hat{s}_{t-1} \\ \hat{s}_t \end{pmatrix} \right\|^2 + \sum_{i=t+1}^{r} |q_i|^2 \leq C$$

as a candidate of $(t-1)$-th component in $\vec{s'}$. The remaining $\hat{s}_{t-2}, \ldots, \hat{s}_1$ is determined in a similar manner. If there is no candidate $\hat{s}_i \in \mathbf{S}$, then we rechoose $\hat{s}_{i+1}$. When we determine $(\hat{s}_1, \ldots, \hat{s}_t)$, we set the new radius to $\|\vec{r} - (\hat{s}_1, \ldots, \hat{s}_t)\|$ and repeat the search part. The last found vector is the ML estimate of the transmitted signal. If there is no possible transmitted signal in the sphere with initial radius $\sqrt{C}$, then we declare erasure or increase the radius $\sqrt{C}$.

3. QR decomposition with sort

By inspecting the search part described above, we see that if we make $R_{i,i}$ large, the number of candidates for $\hat{s}_i$ decreases. It seems that decrease of the number of candidates $\hat{s}_i$ for large i makes the total computational complexity of the search part small.

The QR decomposition computes $R_{i,i}$ in increasing order of i. We propose the QR decomposition with sort that permutes columns of the decomposed matrix before each computation of $R_{i,i}$ such that $R_{i,i}$ is minimized.

The ordinary QR decomposition of H can be sketched as follows: Compute a unitary matrix Q_1 such that the first column of Q_1H is $(R_{1,1}, 0, \ldots, 0)^T$. Let H_2 be $(r-1) \times (t-1)$ submatrix of Q_1H with the first column and the first row of Q_1H removed. Compute a unitary matrix Q_2 such that the first column of Q_2H_2 is $(R_{2,2}, 0, \ldots, 0)^T$. The computation process is recursively repeated until $i = t$.

We will describe the QR decomposition with sort. Observe that in the ordinary QR decomposition $R_{1,1}$ is equal to the norm of the first column vector of H. In order to minimize $R_{1,1}$, we replace the first column of H with the column with minimum norm. Let H' be the column replaced version of H. Compute a unitary matrix Q'_1 such that the first column of Q'_1H' is $(R'_{1,1}, 0, \ldots, 0)^T$. Let $\tilde{H}_2$ be $(r-1) \times (t-1)$ submatrix of Q'_1H' with the first column and the first row of Q'_1H' removed. Replace the first column of $\tilde{H}_2$ with the column with minimum norm in $\tilde{H}_2$. Let H'_2 be the column replaced version of $\tilde{H}_2$. Compute a unitary matrix Q'_2 such that the first column of $Q'_2H'_2$ is $(R'_{2,2}, 0, \ldots, 0)^T$. The computation process is recursively repeated until $i = t$.

With this process we get a QR decomposition $\hat{Q}\hat{R}$ of the column permuted matrix $\hat{H}$ of H. If we apply the traditional search part in Section 2 with $\hat{Q}\hat{R}$, then we get more efficiently the permuted version of the ML estimate $\vec{s'}$. The ML estimate $\vec{s'}$ can be obtained by the permutation.

Other preprocess methods were proposed recently in [3]. We will compare the proposed method with [3]. The same idea of sorted QR decomposition was also used with the nulling and canceling detection in [8].

4. Search part as the shortest path problem

In this section we apply Dijkstra's algorithm to the search part to reduce the complexity of the search part with increase in the storage complexity. Dijkstra's algorithm is an efficient algorithm to find the shortest path from a point to a destination in a weighted graph [1]. In this algorithm, the vertices on the graph are searched for in increasing order of their distance from the departure.

The decisions on $\hat{s}_i$ essentially constructs a tree where nodes at k-th level are correspond to the candidates of $\hat{s}_{t-k+1}$ [5], and the root is placed at the 0-th level. Each level in the tree has $\sharp\mathbf{S}$ nodes corresponding to each element in $\mathbf{S}$. Set the weight of the branch from the node $\hat{s}_i$ to its parent to

$$\left| q_i - \sum_{j=i}^{t} R_{i,j}\hat{s}_j \right|^2,$$

where $\hat{s}_t, \ldots, \hat{s}_{i+1}$ are assumed to be ancestors of the node $\hat{s}_i$ in the tree. The nodes having the same parent are arranged in the increasing order of the distance from left to right. Add a virtual node at $(t+1)$-th level, and connect it to all the nodes at t-th level with branch with weight zero.

If we use Dijkstra's algorithm to find the shortest path from the root to the virtual node at $(t+1)$-th level, equivalently one of nodes at t-th level, we can get the node at t-th level with the minimum

$$\sum_{i=1}^{t}\left|q_i - \sum_{j=i}^{t} R_{i,j}\hat{s}_j\right|^2 = \|Q^*\vec{r} - R\vec{\hat{s}}\|^2 - \sum_{i=t+1}^{r}|q_i|^2,$$

among all nodes at t-th level and it corresponds to the ML estimate.

Dijkstra's algorithm searches for only the nodes whose distance is smaller than the minimum distance of nodes at t-th level, but the original sphere decoder [7] searches for the node whose distance is smaller than $C - \sum_{i=t+1}^{r}|q_i|^2$ and $C - \sum_{i=t+1}^{r}|q_i|^2$ must be greater than the minimum distance of nodes at t-th level in order for ML detection succeed. So the number of searched nodes of Dijkstra's algorithm is smaller than that of the original sphere decoder. In addition, Dijkstra's algorithm does not require the radius of the sphere to be initially set, and it always finds out ML estimate without retrying to search for a lattice point with increased radius. However because we use the priority queue in Dijkstra's algorithm, the storage complexity increases.

5. Computer simulation

In this section, we show how much the complexity of search part is reduced by QR decomposition with sort and Dijkstra's algorithm, and how much the complexity of preprocessing part is increased by QR decomposition with sort.

The radius of sphere used by the traditional search part is defined so that

$$\begin{aligned}\Pr\{\text{transmit point is in sphere}\} &= \Pr\{C > \|\vec{n}\|^2\} \\ &\approx 0.99 \qquad (2)\end{aligned}$$

where C is the square of radius and $\vec{n}$ is a vector whose element is noise at each receive antenna [5]. When there is no lattice point in sphere, we increase the radius to $C+1$, and continue until a lattice point is found.

5.1. The system model

We consider the following system model.

- The number of transmit antennas is equal to the number of receive antennas.
- The fading coefficients obey the $\mathcal{CN}(0,1)$ distribution.
- The signal constellation for each transmit antenna is 64-QAM and all signals are drawn according to the uniform i.i.d. distribution.

In our simulations we use the average number of real multiplications and divisions in each processing as the measure of complexity, and in these simulations we use the complex multiplications that needs three real multiplications and seven real additions, and the complex divisions that needs five real multiplications, two real divisions, and nine real additions [6].

5.2. The computer simulations

In Figs. 1 and 2 we show the effect of various preprocess algorithms on the computational complexities of the preprocess part and the search part without Dijkstra's algorithm. In Figs. 1 and 2, SNR at each receive antenna is set to 28dB. Horizontal axis in figures represents the number of transmit antennas. In figures, SD corresponds to the ordinary QR decomposition, Norm-SD corresponds to sorting the columns of H according to norm of columns once before the QR decomposition [3], Optimal-SD corresponds to sorting columns of H so that $\min_{1\le i\le t} R_{i,i}$ is maximized among all column permutations [3], QR sort-SD corresponds to the proposed preprocess method.

When the number of transmit antenna is 8 the complexity of the search part is reduced about 55 percent from the original sphere decoder by the proposed QR decomposition with sort. However Fig. 2 shows the complexity of preprocess part increases about 10 percent.

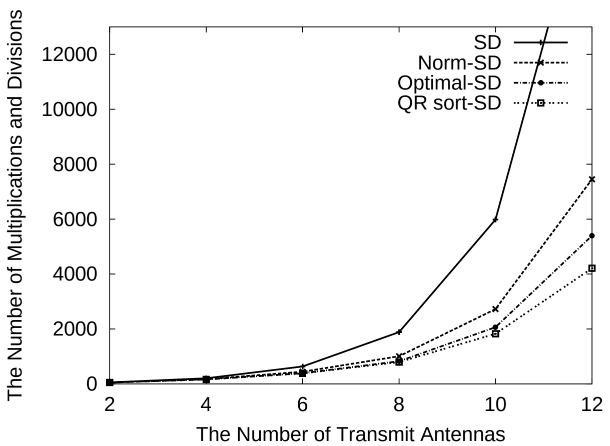

Figure 1: The complexity of search part for each receiving point

In Figs. 3 and 4 we show the comparison of the complexities of the original sphere decoder (SD), Dijkstra's algorithm (Dijkstra), and both of them using QR decomposition with sort (QR sort-SD, QR sort+Dijkstra). The number of transmit antennas is set to 8. Figure 3 shows that the complexity of search

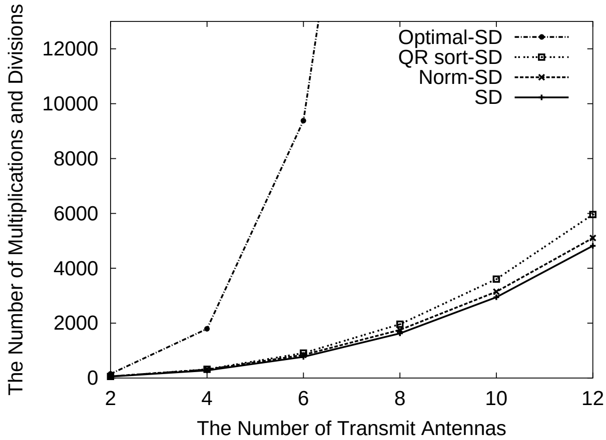

Figure 2: The complexity of preprocess part for each fading matrix

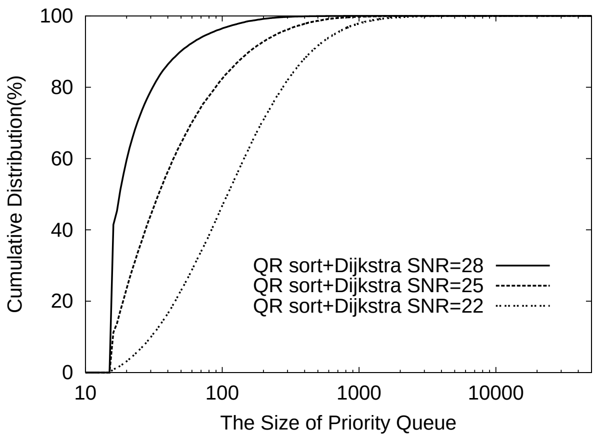

Figure 4: The cumulative distribution of the size of priority queue

part and Fig. 4 shows the cumulative distribution of the size of priority queue with QR decomposition with sort. When SNR is 26dB, the complexity of search part is reduced about 25 percent from the original sphere decoder by Dijkstra's algorithm, and is reduced about 65 percent from the original sphere decoder by combining QR decomposition with sort and Dijkstra's algorithm. Figure 3 also shows that Dijkstra's algorithm is much faster than the sphere decoder when SNR is low.

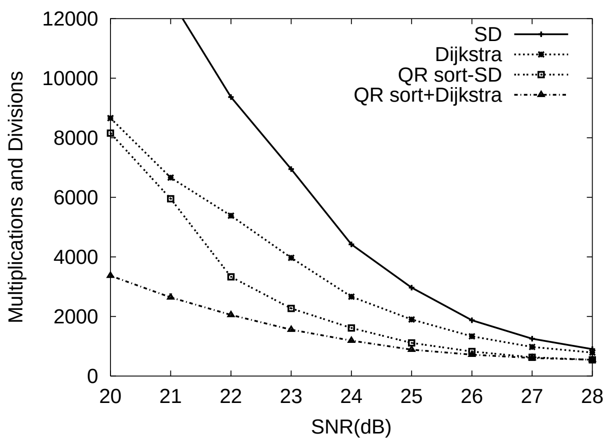

Figure 3: The complexity of search part for each receiving point

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1983.

[2] M. O. Damen, A. Chkeif, and J.-C. Belfiore. Lattice code decoder for space-time codes. *IEEE Comm. Letters*, 4(5):161–163, May 2000.

[3] M. O. Damen, H. El Gamal, and G. Caire. On maximum-likelihood detection and the search for the closest lattie point. *IEEE Trans. Inform. Theory*, 49(10):2389–2402, Oct. 2003.

[4] G. J. Foschini. Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. *Bell Labs Tech. J.*, 1(2):41–59, 1996.

[5] B. Hassibi and H. Vikalo. On the sphere decoding algorithm: Part I and II. preprint available from `http://www.systems.caltech.edu/EE/Faculty/babak/pubs/sphere.html`.

[6] D. E. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley, 2nd edition, 1981.

[7] E. Viterbo and J. Boutros. A universal lattice code decoder for fading channels. *IEEE Trans. Inform. Theory*, 45(5):1639–1642, July 1999.

[8] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer. Efficient algorithm for decoding layered space-time codes. *Electronics Lett.*, 37:1348–1350, Oct. 2001.