

# Reduced complexity FSD algorithm based on noise variance

Xinyu Mao, Shubo Ren and Haige Xiang  
*Institute of Modern Communications*  
*School Electrical Engineering and Computer Science*  
*Peking University*  
*Beijing 100871, China*  
{xymao, renshubo & xianghg}@pku.edu.cn

**Abstract**—multiple-input multiple-output (MIMO) system has very high spectrum efficiency. However, detection is a major challenge for the utilization of MIMO system. But even the fixed sphere decoding (FSD), which is known for its simplicity in calculation, requests too much computation in high order modulation and large number antenna system, especially for mobile battery-operated devices. In this paper, a reduced FSD algorithm is proposed to simplify the calculation complexity of the FSD while maintaining the performance at the same time. Simulation results show the effect of the proposed algorithm. Especially the results in a 4×4 64QAM system show that up to 81.2% calculation can be saved while the performance drop is less than 0.1dB when SNR=30.

**Index Terms**- multiple-input multiple-out (MIMO) systems; fixed sphere decoding (FSD); sphere decoding (SD)

## I. INTRODUCTION

Multiple-input multiple-output (MIMO) systems have been accepted as one of the most significant technology to improve the spectrum efficiency in wireless communications [1][2]. Nevertheless the detection of the MIMO is a big challenge for the application of MIMO systems. Maximum likelihood (ML) detection is able to provide the optimal performance, yet its calculation complexity is too big a burden for system especially with high modulation order and large antenna number [3]. The complexity of zero

forcing (ZF) detection or minimum mean square error (MMSE) is low, but the performance is always bad. Several algorithms have been created to balance the conflict between the performance and the complexity. The V-BLAST algorithm has a little more calculation complexity than the ZF and MMSE algorithm, but its performance promotes a lot, especially combining with the ordering. The class of sphere decoding (SD) algorithm has been widely acknowledged as the one of the most promising algorithms [4][5]. K-best sphere decoding (K-best SD) algorithm is able to work in parallel and the computation complexity is constant, so that it attracts much attention [6]. However, to attain high performance with K-best SD, large K is needed, which causes the calculation complexity arising. The fixed sphere decoding (FSD) algorithm has been proposed to lower the complexity while maintain high performance and the parallel calculating ability of the K-best SD [7][8]. But, it is still necessary to further lower the complexity of the FSD without significant performance drop to meet the demand of the development of wireless communication. In this paper, a new algorithm to further reduce the complexity of the FSD is proposed.

The rest of the paper is organized as follows: Section II reviews the MIMO system. Section III reviews the related algorithm. The new algorithm is proposed in section IV. Its simulation results are shown in section V. At last, the conclusion is drawn in section VI.

## II. MIMO SYSTEM

Consider a complex-valued baseband MIMO system

with  $N_t$  transmit antennas and  $N_r$  receive antennas, where  $N_r \geq N_t$ , like Fig. 1. Assuming it is

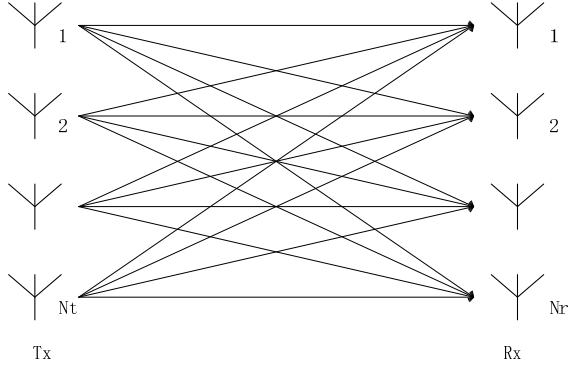


Fig. 1. MIMO system

symbol-synchronous at the receiver, the received symbol can be written as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (1)$$

where  $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_{N_t}]^T$  denotes the  $N_t$ -vector of the transmit symbol,  $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_{N_r}]^T$  is the  $N_r$ -vector of the received symbol and  $\mathbf{n} = [n_1 \ n_2 \ \dots \ n_{N_r}]^T$  is the  $N_r$ -vector of the independent and identically distributed additive white complex Gaussian noise.  $\mathbf{H}$  denotes the  $N_r \times N_t$  Rayleigh fading channel matrix with  $h_{ij} \sim \text{CN}(0,1)$ . For sake of simplicity, we set  $N_r = N_t = N/2$ . The ML detection is given by

$$\hat{\mathbf{x}}_{ML} = \arg \min_{\mathbf{x} \in \mathfrak{S}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (2)$$

where  $\mathfrak{S}$  denotes the  $N/2$ -dimensional constellation lattice, like Fig. 2.

### III. FSD ALGORITHM

#### A. Tree search and SIC algorithm

The variables in (2) are complex. The analysis is clear to be understood in real domain. So we transform (2) into

$$\begin{pmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{pmatrix} = \begin{pmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \Re(\mathbf{H}) \end{pmatrix} \begin{pmatrix} \Re(\mathbf{x}) \\ \Im(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} \Re(\mathbf{n}) \\ \Im(\mathbf{n}) \end{pmatrix} \quad (3)$$

where  $\Re(\square)$  and  $\Im(\square)$  represent the real or the image part of  $\square$ . Certainly, the proposed algorithm can work

well in the complex domain.

Before further disposition, the channel matrix

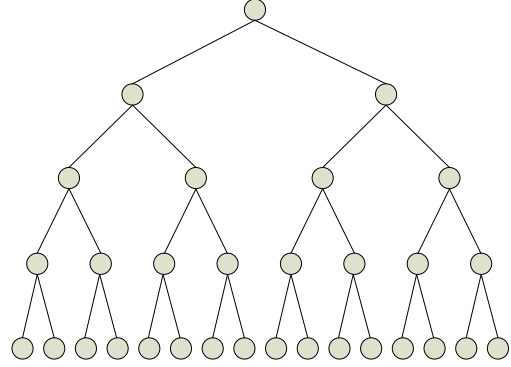


Fig. 2. Tree search and ML algorithm

should be decomposed to triangular matrix.

$$\begin{pmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \Re(\mathbf{H}) \end{pmatrix} = \mathbf{Q}\mathbf{R} \quad (4)$$

where  $\mathbf{Q}$  is a unitary matrix and  $\mathbf{R}$  is an upper triangular matrix. Then (2) is left multiplied by  $\mathbf{Q}^H$  to remove  $\mathbf{Q}$  and to make the channel matrix triangular. Equation (3) will be expressed as

$$\boldsymbol{\rho} = \mathbf{R}\mathbf{s} + \boldsymbol{\eta} \quad (5)$$

where  $\mathbf{s} = \begin{pmatrix} \Re(\mathbf{x}) \\ \Im(\mathbf{x}) \end{pmatrix}$ ,  $\boldsymbol{\rho} = \mathbf{Q}^H \begin{pmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{pmatrix}$  and

$\boldsymbol{\eta} = \mathbf{Q}^H \begin{pmatrix} \Re(\mathbf{n}) \\ \Im(\mathbf{n}) \end{pmatrix}$ . The ML algorithm can be expressed as

$$\hat{\mathbf{s}} = \arg \min_{\Omega} \|\boldsymbol{\rho} - \mathbf{R}\mathbf{s}\|^2 \quad (6)$$

where  $\Omega$  is the real counterpart of  $\mathfrak{S}$ ,

$\hat{\mathbf{s}} = (\hat{s}_1 \ \hat{s}_2 \ \dots \ \hat{s}_N)^T$  is the estimated value of transmit vector.

Equation (5) can also be written as

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_N \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1N} \\ 0 & r_{22} & \dots & r_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{NN} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{pmatrix} \quad (7)$$

From (7), it can be found that

$$\rho_N = r_{NN} s_N + n_N \quad (8)$$

So  $s_N$  can be calculated by

$$\hat{s}_N = \text{round}\left(\frac{\rho_N}{r_{NN}}\right) \quad (9)$$

where  $\hat{s}_N$  is the estimated transmit signal.

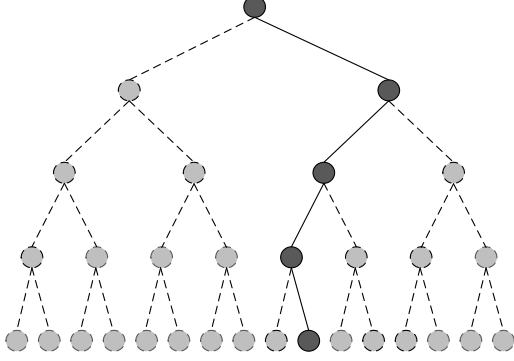


Fig. 3. SIC algorithm

Also from (7), each transmit signal can be calculated recursively.

$$\hat{s}_i = \text{round}\left(\frac{1}{r_{ii}}\left(\rho_i - \left(\sum_{k=i+1}^N r_{ik} \hat{s}_k\right)\right)\right) \quad (10)$$

So the solution of (6) can be searched layer by layer.

That is the principle of tree search and successive interference cancellation (SIC) algorithm are shown in Fig. 3. Each possible transmit signal  $s_i$  is called a node in the searching tree. Each line from the top node is called a searching rout. The solid lines and nodes mean the solution of the SIC.

### B. ML algorithm

If the solution of a certain  $s_i$  is wrong, the solutions of  $s_j$  (when  $j < i$ ) will almost always have errors. ML algorithm avoids it by calculating all the possible nodes in the tree, then selects the node which satisfies (6) from all the possible nodes. So the ML algorithm provides the best detection performance.

### C. Euclidean distance and the partial Euclidean distance

The Euclidean distance (ED) between transmit and receive symbol is defined as

$$D = \left(\sum_{j=1}^N \left(\rho_j - \sum_{k=j}^N r_{jk} \hat{s}_k\right)^2\right)^{1/2} \quad (11)$$

Define the partial Euclidean distance (PED) as

$$D_i = \left(\sum_{j=i}^N \left(\rho_j - \sum_{k=j}^N r_{jk} \hat{s}_k\right)^2\right)^{1/2} \quad (12)$$

Obviously, the solution of (6) is to find the  $\hat{\mathbf{s}}$  with the smallest ED.

### D. The SD algorithm

The main idea of the SD algorithm is to constrain the searching area to lower the complexity of the ML. A radius  $R$  around the receiver vector in the hyper-sphere is set first. So that the SD search can be represented as

$$\hat{\mathbf{x}}_{SD} = \arg \min_{\substack{\mathbf{x} \in \mathcal{N} \\ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \leq R^2}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (13)$$

After the triangular decomposition (4), (13) can be expressed as

$$\|\boldsymbol{\rho} - \mathbf{R}\hat{\mathbf{s}}\|^2 \leq R \quad (14)$$

The solution to (14) can be got recursively using a depth-first tree search. Consider (7), only a few nodes can be searched. They satisfy

$$\left(\hat{s}_i - z_i\right)^2 \leq \frac{T_i}{r_{ii}} \quad (15)$$

where

$$z_i = \frac{1}{r_{ii}} \left(\rho_i - \sum_{k=i+1}^N r_{ik} \hat{s}_k\right) \quad (16)$$

and

$$T_i = R^2 - \sum_{j=i+1}^N \left(\rho_j - r_{jj} \hat{s}_j - \sum_{k=j+1}^N r_{jk} \hat{s}_k\right)^2 \quad (17)$$

with  $z_N = \hat{s}_N$  and  $T_N = R^2$ . It is obvious that

$$T_N > T_{N-1} > \dots > T_1 \quad (18)$$

SD algorithm starts the search from the nodes on the top of the search tree to the leaf nodes (the nodes at the bottom of the tree). If a solution is found, the Euclidean distance of this route takes the place of the  $R$  as the new radius, and then new search continues on other routes until all routes are searched. If the  $R$  is set too small to find a solution in it,  $R$  should be adjusted larger to search again until a solution is found. But that is a bad situation because all calculation in the previous search is wasted. SD algorithm cannot run in parallel because the radius is

always renewed when any new solution is found.

The main idea of the K-best SD algorithm is to execute the search parallel. Also in the search tree, a certain number of routes are explored from the top to the

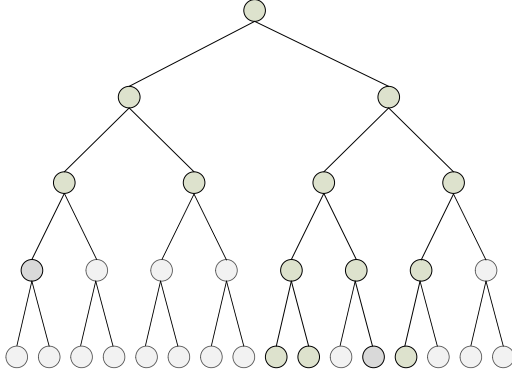


Fig. 4. K-best SD algorithm.

bottom at the same time. To lower the node number, only  $K$  nodes with the smallest PED are kept at each layer, like Fig. 4.

#### E. The K-best SD and FSD algorithm

The main idea of the K-best SD algorithm is to execute the search parallel. To lower the node number, only  $K$  nodes with the smallest PED are kept at each layer. To avoid performance dropping much, the number of  $K$  cannot be set small. That is why the complexity of the K-best SD is higher than that of the SD in the same performance. But the K-best SD attracts much attention because of the property of parallelism. Many works have been proposed to reduce the complexity of the K-best SD. One of the famous is the FSD algorithm.

Denote  $B_i$  the candidate number of the child nodes at level  $i$ . For the K-best SD,  $B_i=m$  ( $m$  is the order of the modulation). Then the total number of the visited nodes is  $K \times m$  for every layer. When the modulation order is high, the number of the visited nodes is a big number. That is a main reason why the complexity of the K-best SD is high.

The FSD explores the characteristic of the triangular matrix, and makes use of the fact that

- The diagonal elements of it have a Chi-square distribution with  $(N-i+1)$  degrees of freedom.
- The off-diagonal elements are independent complex Gaussian random variables  $r_{ij} \sim CN(0,1)$ .

Therefore, the diagonal elements satisfy

$$E[r_{N,N}^2] < E[r_{N-1,N-1}^2] < \dots < E[r_{1,1}^2] \quad (19)$$

Consider both (18) and (19), we have

$$E\left[\frac{T_N}{r_{N,N}^2}\right] > E\left[\frac{T_{N-1}}{r_{N-1,N-1}^2}\right] > \dots > E\left[\frac{T_1}{r_{1,1}^2}\right] \quad (20)$$

Consider both (15) and (20), the average limited area becomes smaller and smaller. It is sure that the average number of nodes inside the limited area becomes smaller and smaller. So it is reasonable to reduce the candidate number of the node layer by layer. By this way, the FSD algorithm can drop the complexity of the K-best SD algorithm without large performance drop. For example, in a  $4 \times 4$  system with 4-QAM modulation, the number of children nodes number per level can be  $\mathbf{B} = (B_1 \ B_2 \ B_3 \ B_4)^T = (1 \ 1 \ 2 \ 3)^T$ , like Fig. 5.

Each element of the vector means the number of the visited children nodes of this layer. For example,  $B_4 = 3$  means the number of the visited children nodes for each node in layer four is three. The algorithm starts from the layer of the largest number, so the last number of  $\mathbf{B}$  is always the largest one. The solid lines in figure mean the visited nodes. The dotted lines mean the children nodes which will be omitted. The corresponding calculation will be cut for these nodes and so the complexity will drop.

#### IV. NOISE VARIANCE BASED REDUCED FSD ALGORITHM

The complexity of the FSD is still high, so that works have done. Some works simplify the calculation by redesigning the ordering preprocessing [9][10]. [11] simplifies the calculation by reducing the numbers of kept routes. But the numbers are selected without strict proving. [12][13] simplify the calculation by setting a threshold which is calculated according to the intersection point of the probability distribution function (PDF) curves of  $\chi_N^2$  and  $\chi_{N,\gamma}^2$ , where  $\chi_N^2$  is the central chi-square distribution and  $\chi_{N,\gamma}^2$  is the non-central chi-square distribution. But that is the way to

balance the error of both  $\chi_N^2$  and  $\chi_{N,\gamma}^2$  when both two distributions are detected. So it is not the optimal threshold for only  $\chi_N^2$  detection.

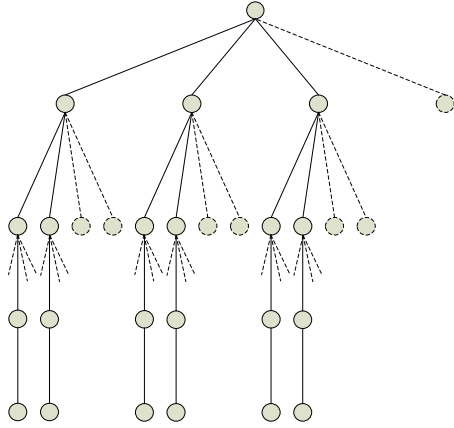


Fig. 5. FSD algorithm.

In this paper, we reduce the complexity of the FSD algorithm by explore the statistics property of the receive signal.

#### A. Principle

The noise generates the bias between the receive signal and the product of the channel matrix and the transmit signal. The value of the noise variance determines mean square of the bias. So if the variance of noise is known, a threshold can be set to cut the node whose PED larger than it, like Fig. 6. Fortunately, when a channel condition is known, we can get the power of the noise. Given  $N$  receive antennas for a system, a pre-calculated radius, which is called pruning radius, can be set.

$$\lambda = \sqrt{\beta N \sigma^2} \quad (21)$$

where  $\beta$  is a tuning parameter to balance the performance and the complexity. Obviously, the larger the tuning parameter, the larger the pruning radius, the more number of nodes is kept and the less performance drop. On the contrary, the smaller the tuning parameter, the smaller the pruning radius and the more calculation reduced. So a real system can choose the appropriate value of tuning parameter according to the QoS requirement.

#### B. The algorithm

The proposed algorithm is given as follows:

- 1) Set  $\beta$  according to the toleration of performance drop and complexity, and then get pruning radius  $\lambda$ .

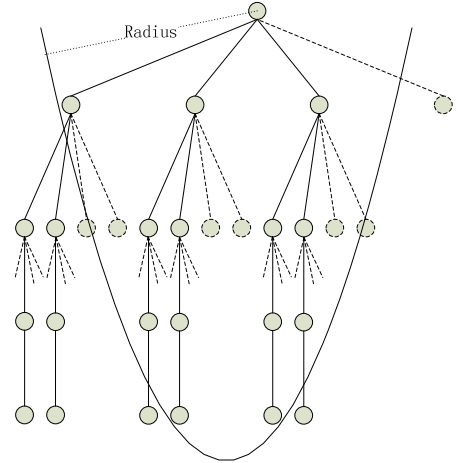


Fig. 6. Reduced FSD algorithm.

- 2) The channel matrix is QR decomposed, and  $\mathbf{y}$  is left multiplied by  $\mathbf{Q}$  to get  $\boldsymbol{\rho}$ .
- 3) Set  $le=N$
- 4) Extend the surviving nodes to their children nodes according to the presetting expend number  $B_i$ .
- 5)  $le=le-1$
- 6) If  $le>1$ , go to step 4).
- 7) If  $le=1$ , select the node that has the smallest ED as the solution of the algorithm.

#### C. Ordering

It is well known that the channel matrix ordering is a very useful preprocessing stage, which can significantly promote the performance. The V-BLAST algorithm orders the channel matrix so that the layer with the smallest noise power gain is detected first, the second detected layer with the second smallest noise power gain, and so on. This order guarantees the detection errors for the previous detected layers are smallest than that of the latter layers. The reason is because the algorithm detects layer by layer. If there happens any error in the previous layers, the later detection will be almost all wrong.

The famous V-BLAST ordering algorithm is

- for  $i=N, N-1, \dots, 1$
- $\mathbf{G}_i = \mathbf{H}_i^\dagger$

- $k_i = \arg \min_{j \in \{k_1, \dots, k_{i-1}\}} \left\| (\mathbf{G}_i)_j \right\|^2$
- $\mathbf{H}_i = (\mathbf{H}_{i-1})_{k_i}$
- *end*

where  $\mathbf{H}_i = (\mathbf{H}_{i-1})_{k_i}$  means that the  $k_i$ -th column of  $\mathbf{H}$

should be set zero and  $\mathbf{H}_i^\dagger = (\mathbf{H}_i^H \mathbf{H}_i)^{-1} \mathbf{H}_i^H$ .

There is a little difference between the above algorithm and the original V-BLAST ordering algorithm. In the latter, there are other steps for the cancellation of the interference. But the node number is not one for the former, it is impossible to cancel the interference for all nodes. Here we only keep the same part.

The previous detected layers have more impact on the performance than the later detected layers. So the number of the children nodes for the previous detected layers is often set as large as it can be. In many case,  $\mathbf{B} = (1 \ 1 \ 1 \ 4)^T$  has better performance than  $\mathbf{B} = (1 \ 1 \ 2 \ 2)^T$  or  $\mathbf{B} = (1 \ 2 \ 1 \ 2)^T$ . But in some cases, the children nodes' number of the previous detected layer is set as large as possible so that it may be equal to the modulation order, or say the number of the constellation. There is no error propagation when all possible nodes are kept in a certain layer for further detection. So it is wasteful to set the noise gain small. On the contrary, the noise gain can be set as large as possible. The steps performed in the algorithm are the following.

- *for*  $i=N, N-1, \dots, 1$
- $\mathbf{G}_i = \mathbf{H}_i^\dagger$
- $k_i = \begin{cases} \arg \max_{j \in \{k_1, \dots, k_{i-1}\}} \left\| (\mathbf{G}_i)_j \right\|^2 & \text{if } B_i = P \\ \arg \min_{j \in \{k_1, \dots, k_{i-1}\}} \left\| (\mathbf{G}_i)_j \right\|^2 & \text{if } B_i < P \end{cases}$

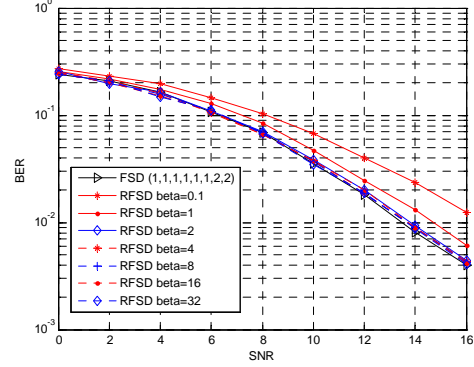


Fig. 7. Performance of RFSD in a 4×4 4QAM system.

- $\mathbf{H}_i = (\mathbf{H}_{i-1})_{k_i}$
- *end*

where  $P$  is the maximum possible number of nodes, the number of the constellation. We choose this ordering algorithm in the proposed algorithm.

## V. SIMULATION RESULTS

The proposed algorithm is compared with the FSD by the bit error rate (BER) and the computation complexity. Data are simulated in a 4×4 Rayleigh flat fading plus additive white Gaussian noise MIMO channel.

Fig. 7~12 show the performance and the complexity of the FSD and the noise variance based reduced FSD algorithm (RFSD in figures). Fig. 13~18 show those of the ordering FSD and the noise variance based reduced ordering FSD algorithm (ROFSD in figures).

All the performance figures show the impacts on BER of different  $\beta$ . When  $\beta$  is large enough, the BER performance of the proposed algorithm is very close to that of the FSD. For example, When  $\beta = 32, 16, 8$ , the performance drop is hard to distinguish for the 64QAM modulation for both ordering and non-ordering FSD. In the figures showing the ordering algorithm, the difference between the proposed algorithm with  $\beta$  large enough and the ML algorithm is very small.

All the complexity figures illustrate the complexity of the proposed algorithm and the impacts on BER of different  $\beta$ . It can be found that, the number of reduced node is associated with the SNR and  $\beta$ . The percent of reduced node number arises according to the increase of SNR and drop of  $\beta$ . When the SNR is high enough, different lines tend to the same. When the SNR is high

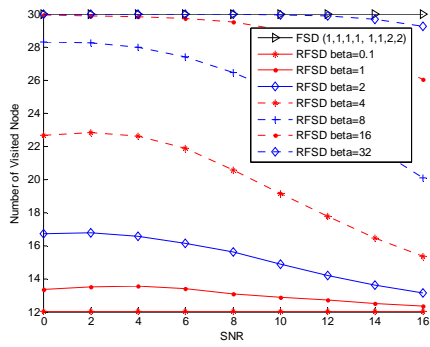


Fig. 8. Complexity of RFSD in a 4x4 4QAM system.

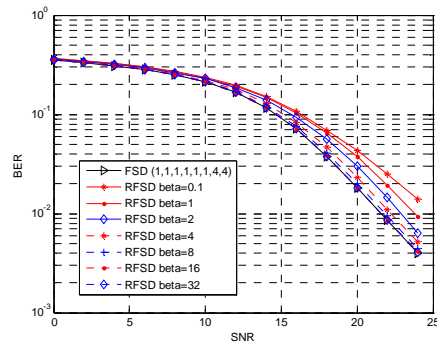


Fig. 9. Performance of RFSD in a 4x4 16QAM system.

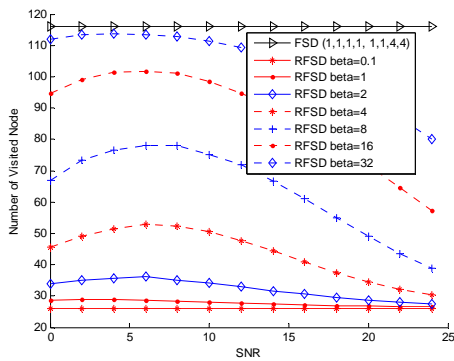


Fig. 10. Complexity of RFSD in a 4x4 16QAM system

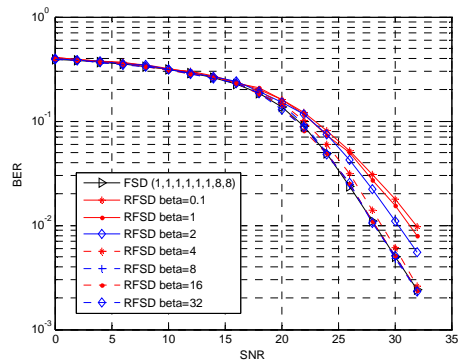


Fig. 11. Performance of RFSD in a 4x4 64QAM system.

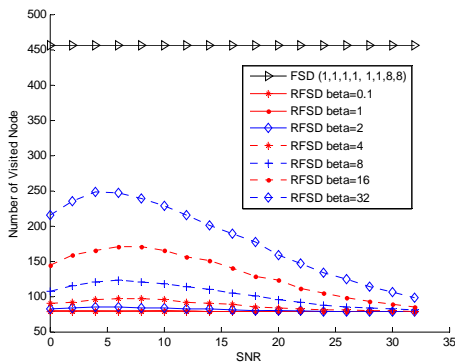


Fig. 12. Complexity of RFSD in a 4x4 64QAM system

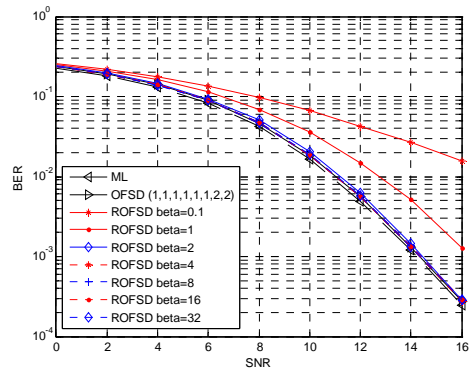


Fig. 13. Performance of order RFSD in a 4x4 4QAM system.

enough, different lines tend to the same. In Fig. 18, when SNR=30, the percent of the reduced nodes are about 82.9% when  $\beta = 0.1, 1$  or 2. When  $\beta = 4, 8$ , it becomes to about 82.4%, 81.2%. The percent turns to 77.6% or 69.1% when  $\beta = 16$  or 32. From the complexity figures, it can be concluded that the reduced calculation becomes large when the modulation order arising.

### VI. CONCLUSION

This article proposed a new algorithm to reduce the calculation complexity of the FSD algorithm. According to the noise variance, a threshold can be set. A node who's PED larger than it can be cut during the processing of tree search. To balance the performance and the complexity, a tuning parameter can be multiplied into the

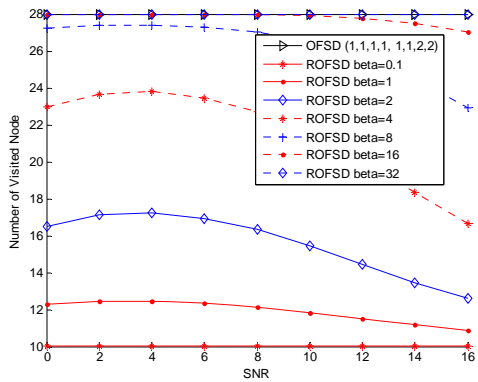


Fig. 14. Complexity of order RFSD in a 4x4 4QAM system

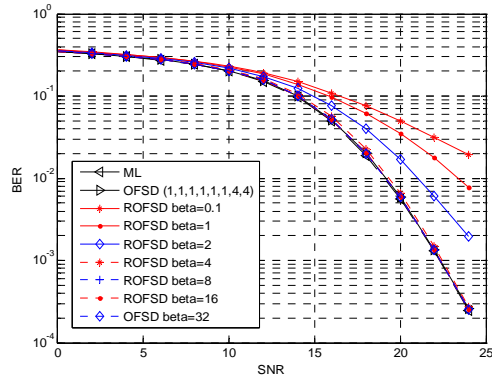


Fig. 15. Performance of order RFSD in a 4x4 16QAM system.

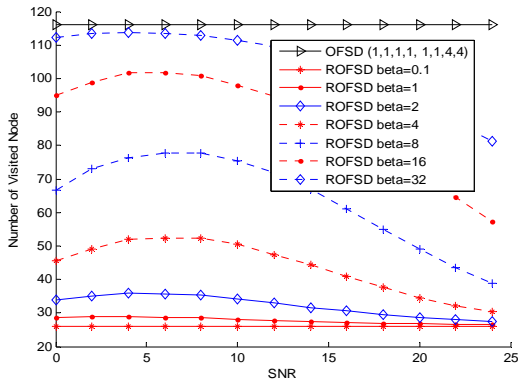


Fig. 16. Complexity of order RFSD in a 4x4 16QAM system

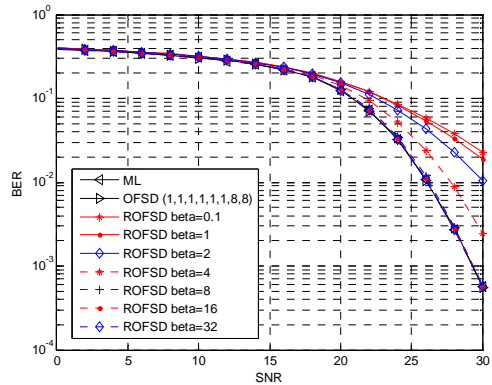


Fig. 17. Performance of order RFSD in a 4x4 64QAM system.

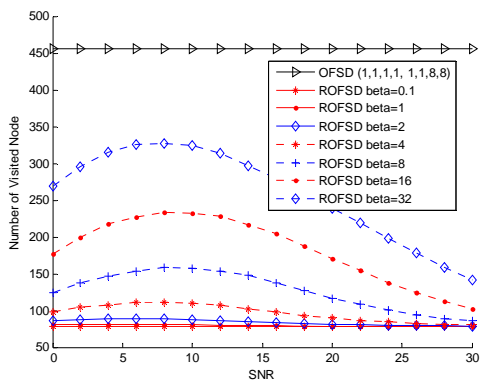


Fig. 18. Complexity of order RFSD in a 4x4 64QAM system

threshold. Simulation results in several scenes show that the algorithm can efficiently reduce the complexity of the FSD and maintain the performance.

ACKNOWLEDGEMENT

This work is supported by NSFC Grant No.60972008. Shubo Ren is the corresponding author.

REFERENCES

- [1] A. Paulraj, D. Gore, R. Nabar, and H. Bolcskei, "An overview of MIMO communications - a key to gigabit wireless," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198-218, 2004.
- [2] G. Golden, C. Foschini, R. Valenzuela, and P. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," *Electronics Letters*, vol. 35, no. 1, pp. 14-16, 1999.
- [3] Xu Zhu and R. Murch, "Performance analysis of maximum likelihood detection in a MIMO antenna system," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 187-191, Feb. 2002.
- [4] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *Information Theory, IEEE Transactions on*, vol. 45, no. 5, pp. 1639-1642, 1999.
- [5] A. Murugan, H. El Gamal, M. Damen, and G. Caire, "A unified framework for tree search decoding: rediscovering the sequential decoder," *Information Theory, IEEE Transactions on*, vol. 52, no. 3, pp. 933-953, 2006.



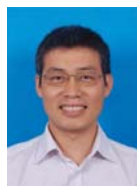
- [6] Kwan-wai Wong, Chi-ying Tsui, R. Cheng, and Wai-ho Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*, pp. III-273-III-276.
- [7] L. G. Barbero and J. S. Thompson, "A Fixed-Complexity MIMO Detector Based on the Complex Sphere Decoder," in *2006 IEEE 7th Workshop on Signal Processing Advances in Wireless Communications*, pp. 1-5, 2006.
- [8] L. Barbero and J. Thompson, "Fixing the Complexity of the Sphere Decoder for MIMO Detection," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 6, pp. 2131-2142, 2008.
- [9] X. G. Dai, S. W. Cheung, and T. I. Yuk, "Simplified ordering for fixed-complexity sphere decoder," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference on ZZZ - IWCMC '10*, p. 804, 2010.
- [10] K. Lee and B. Daneshrad, "Reduced complexity two-stage approximate sorting Fixed Complexity Sphere Decoder with real-valued lattice formulation for MIMO system," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pp. 1-7, 2008.
- [11] Cong Xiong, Xin Zhang, Kai Wu, and Dacheng Yang, "A simplified fixed-complexity sphere decoder for V-BLAST systems," *Communications Letters, IEEE*, vol. 13, no. 8, pp. 582-584, 2009.
- [12] X. Wu, J. S. Thompson, and A. M. Wallace, "An improved sphere decoding scheme for mimo systems using an adaptive statistical threshold," 2009.
- [13] X. Wu and J. Thompson, "Accelerated sphere decoding for multipleinput multiple-output systems using an adaptive statistical threshold," *Signal Processing, IET*, vol. 3, no. 6, pp. 433-444, 2009.



**Xinyu Mao** was born in Yuci, Shanxi, China, in 1972. He received the Master of Engineering in Electronics from Peking University in 1999. Now he is a Ph.D. candidate in the School of Electronics Engineering and Computer Science at Peking University.

Since 1999, he is a lecturer at School of EECS at Peking University, Beijing, China. His research interests include wireless communications and communications signal processing, especially MIMO and OFDM.

Mr. Mao is a student member in IEICE.



Shubo Ren was born in Zunhua, Hebei, China, in 1976. He received the Ph.D. in Communications and Information Systems from Peking University. Now he is a Post Doctor in the School of Electronics Engineering and Computer Science at Peking University.

His major research interests include the communication theory and algorithm in B3G/4G, the key technologies in satellite communications and mobile communications.

Doctor Ren is an associate member in IEEE.



**Haige Xiang** was born in Taiping, Anhui, China, in 1941. He received the bachelor of Engineering in Electronics from Peking University in 1964.

Since 1964, he is a teacher at Electronics Department at Peking University, Beijing, China. His research interests include communications theory, communications signal processing, wireless communications, satellite communications, channel coding.

Professor Xiang is a member in IEEE.