Iranian Journal of Science & Technology, Transaction B, Engineering, Vol. 30, No. B2
Printed in The Islamic Republic of Iran, 2006

"Research Note"

A NEW MODIFIED VITERBO-BOUTROS SPHERE DECODING ALGORITHM[*]

S. M. RAZAVIZADEH[1,2**], V. TABATABA VAKILI[2] AND P. AZMI[3]

[1] Iran Telecommunication Research Center (ITRC), Tehran, I. R. of Iran
Email: smrazavi@itrc.ac.ir
[2] Iran University of Science and Technology (IUST), Tehran, I. R. of Iran
[3] Tarbiat Modarres University (TMU), Tehran, I. R. of Iran
Email: smrazavi@itrc.ac.ir

Abstract: In this paper, a new sphere decoding algorithm is introduced which is a modification of the Viterbo-Boutros (VB) algorithm. In the proposed algorithm, the problem of the initial selection of the search sphere radius in the VB algorithm has been removed, and furthermore, the speed of the algorithm is increased. To examine the performance of the proposed algorithm, we employ it in detection of signals in code division multiple access (CDMA) and multiple-input multiple-output (MIMO) systems. Computer simulations, by using the proposed algorithm, show a significant performance improvement with respect to other suboptimal detection methods.

Keywords– Sphere decoding, lattice decoding, ML detection, CDMA, MIMO

1. INTRODUCTION

The sphere decoding technique is a fast method for solving the problem of the closest point in the lattices [1]-[5]. This method can be employed in several applications, including multi-user detection (MUD) in CDMA systems [5], space-time decoding and equalization [3]. In all of these systems, the received signal can be modeled by an *n*-dimensional lattice as follows:

$$\mathbf{r} = \mathbf{uB} + \mathbf{n} \tag{1}$$

where $\mathbf{r} \in \mathbf{R}^{\mathbf{m}}$ is a row vector and shows the received signal, $\mathbf{u} \in \mathbf{D}^{\mathbf{n}}$ $(\mathbf{D} \subset \mathbf{Z})$ is a row vector and shows the transmitted information symbols and $\mathbf{B} \in \mathbf{R}^{\mathbf{n}\times\mathbf{m}}$ is the lattice generating matrix. $\mathbf{n} \in \mathbf{R}^{\mathbf{m}}$ is the additive white Gaussian noise vector. Here, $\mathbf{R}$ is the field of real numbers, $\mathbf{Z}$ is the ring of integers and $\mathbf{D}$ is the constellation from which the information symbols $\mathbf{u}$ are selected. In this paper, we assume $m \geq n$.

The received signal in (1) is then given to the detector which must find the transmitted vector $\mathbf{u}$. The optimum detection method for signals in (1) is the maximum likelihood (ML) detection. But the ML detector needs exhaustive search on all possible data vectors (in an *m*-dimensional lattice) to find the transmitted data. This exhaustive search is very complex, especially for high dimensional constellations $\mathbf{D}$ [2]-[4]. However, the sphere decoding (SD) algorithm, by restricting the search space by a hyper-sphere of radius $\sqrt{C}$ and checking only the lattice points that lie inside this hyper-sphere, reduces the computational complexity of the search process significantly [2].

One of the most efficient sphere decoding algorithms that has been proposed in the literature is the algorithm proposed by Viterbo and Boutros (VB) in [1]. However, one important drawback in the VB algorithm and other similar SD algorithms is the choice of the initial value of the search radius. If this radius is chosen too small, there may be no solution for the algorithm (no point inside the hyper-sphere).

[*]Received by the editors May 4, 2005; final revised form March 7, 2006.
[**]Corresponding author

On the other hand, if the radius is chosen too large, the number of the checked points may be very high and the algorithm will be ineffective. Although there are some methods for deciding on the initial value of *C*, there is not yet any general method that works well in all different applications. In all proposed methods, it is still possible that no valid point is found inside the hyper-sphere. In those cases, the search procedure must be repeated with a larger value of *C*. This re-iteration considerably wastes the computations. On the other hand, in some cases, like signal detection in fading channels, the choice of *C* is more difficult [1]. In these cases, because of rapid changes in the channel condition, the initial value of *C* must be determined more frequently.

In this paper, we introduce a new algorithm which is not sensitive to the selection of initial search radius. Moreover, the complexity of the proposed method is much less than the original VB algorithm (in other words, it is considerably faster than the original VB algorithm). We introduce some parameters to control the complexity of the algorithm. To evaluate the performance of the proposed algorithm, we use it in detecting signals in CDMA and MIMO systems and illustrate simulation results in bit error rate (BER) performance of the proposed algorithm for different values of control parameters. Our simulation results show that the proposed algorithm outperforms other sub-optimum detection methods in CDMA and MIMO systems and besides, its complexity is still extremely less than the original VB sphere decoding algorithm.

2. THE VB SPHERE DECODING ALGORITHM

In general, the SD algorithm searches for the lattice points that lie inside a hyper-sphere of radius $\sqrt{C}$ as [1]-[4]

$$\left\|\mathbf{r}-\mathbf{uB}\right\|^2_{\mathbf{u}\in\mathbf{D}^n} \le C \tag{2}$$

The VB algorithm gives an efficient way for solving the following equation. If we assume $\mathbf{r}=\boldsymbol{\rho}\mathbf{B}$, then

$$\left\|\mathbf{r}-\mathbf{uB}\right\|^2 = (\mathbf{u}-\boldsymbol{\rho})^T\mathbf{B}^T\mathbf{B}(\mathbf{u}-\boldsymbol{\rho}) = (\mathbf{u}-\boldsymbol{\rho})^T\mathbf{R}^T\mathbf{R}(\mathbf{u}-\boldsymbol{\rho}) \tag{3}$$

where $\mathbf{R}$ is found by QR decomposition of the matrix $\mathbf{B}$ or by Cholesky decomposition of the Gram matrix $\mathbf{G}=\mathbf{BB}^T$ [8]. Therefore Eq. (2) can be expressed as [1]

$$\begin{aligned}\left\|\mathbf{R}(\mathbf{u}-\boldsymbol{\rho})^T\right\|^2 &= \sum_{i=1}^{n}\left(r_{ii}(u_i-\rho_i)+\sum_{j=i+1}^{n}r_{ij}(u_j-\rho_j)\right)^2\\ &= r_{nn}^2(u_n-\rho_n)^2+\left[r_{n-1,n-1}(u_{n-1}-\rho_{n-1})+r_{nn}(u_n-\rho_n)\right]^2+\cdots\le C\end{aligned} \tag{4}$$

where r_{ij} is the element *(i,j)* of matrix R. The first necessary condition for (4) is that

$$r_{nn}^2(u_n-\rho_n)^2\le C \;\Rightarrow\; \left\lceil -\frac{\sqrt{C}}{r_{nn}}+\rho_n\right\rceil \le u_n \le \left\lfloor \frac{\sqrt{C}}{r_{nn}}+\rho_n\right\rfloor \tag{5}$$

which shows the bounds on the *n*'th component of $\mathbf{u}$. $\lfloor x\rfloor$ and $\lceil x\rceil$ denote the smallest integer greater than *x* and the greatest integer smaller than *x*, respectively. Now for every u_n satisfying (5) and by using (2), the bounds on u_{n-1} are found. The *i*'th component of $\mathbf{u}$ is determined according to the previous components as [1]

$$\left\lceil \frac{\sum_{j=i+1}^{n}r_{ij}(u_j-\rho_j)-\sqrt{C-\sum_{l=i+1}^{n}\left(r_{ll}(u_l-\rho_l)+\sum_{j=l+1}^{n}r_{lj}(u_j-\rho_j)\right)^2}}{r_{ii}}+\rho_i\right\rceil \le u_i \le \left\lfloor \frac{\sum_{j=i+1}^{n}r_{ij}(u_j-\rho_j)+\sqrt{C-\sum_{l=i+1}^{n}\left(r_{ll}(u_l-\rho_l)+\sum_{j=l+1}^{n}r_{lj}(u_j-\rho_j)\right)^2}}{r_{ii}}+\rho_i\right\rfloor \tag{6}$$

This process continues until the bounds of all components of $\mathbf{u}$ are calculated, and therefore one point inside the hyper-sphere is found. Then the square distance of this point to the center (i.e. received point) is compared to the current value of the search radius. If it is smaller, then its value will be a candidate for the new search sphere radius and then the search process continues with this new radius [1].

3. THE MODIFIED VB SPHERE DEDOING ALGORITHM

As it was stated before, one important factor in the VB algorithm is how to choose the initial search radius C. In this section, we present a new modified VB (MVB) algorithm in which the initial value of search radius C is not important. In this algorithm, first, we select radius C very large (e.g. +∞) and with some modifications in the original sphere decoder, we try to compensate this selection and to decrease the complexity of the algorithm.

In the original VB decoder, whenever a point is found inside the sphere, the radius of the search sphere is decreased to the distance of this new point to the received vector [1]. The main idea in our proposed method is accelerating this radius decrease. In the proposed algorithm, whenever a point is found inside the sphere, the new radius is selected as $d^2 = k * \hat{d}^2$, where $\hat{d}^2$ is the squared distance of the previous founded point to the center. The scale parameter $0 \le k \le 1$ increases the rate of decreasing of the sphere radius. At the beginning, the radius C may be selected very large. In other words, in the first stage, there is no limitation or any effort on choosing the search sphere radius.

Now, the main question is how the scale parameter k is selected. In general, we can say that as small as parameter k is, the algorithm will be faster. But this may cause a problem when the algorithm reaches to the points near the received point. More precisely, if the distance of the last visited point before the answer point is less than 1/k distance of the ML solution point to the center, the ML solution point will not be selected and the previous checked point will be chosen erroneously. Hence, we must adjust parameter k adaptively.

In order to determine the lattice points inside the n-dimensional hyper-sphere, the VB algorithm splits the n-dimensional problem into n one-dimensional problems. When the possible values for the m'th dimension are determined, for each of these values the points in the (m+1)'th dimension will be determined. Hence, it can be said that the VB algorithm constructs a search tree in which the number of levels is equal to the dimension of lattice (n*)* [3]. Therefore, the complexity of the SD algorithms is determined by the number of nodes in the search tree.

In the proposed technique, the factor k is selected in proportion to the number of nodes in the first level of the search tree (corresponding to n'th component of $\mathbf{u}$). In other words, this factor is selected in proportion to the number of points which are inside the search hyper-sphere with a given radius. Applying the limits computed by (6) on the set $\mathbf{D}$ gives the possible values for the n'th component of $\mathbf{u}$ or equally the number of nodes in the first stage of the search tree.

Let t be the number of possible values in the n’th dimension. A function k(t) that has the above properties and we propose to be used for the MVB algorithm is

$$k(t) = \exp\left[-\left(\frac{1}{\alpha}t\right)^{\beta}\right] \tag{7}$$

where the parameters α and β control the complexity and the speed of the MVB algorithm. For the case of $\alpha = \infty$, the value of k is one, which corresponds to the original VB algorithm. Decreasing α from ∞, increases the speed of the algorithm. But as we will see in the next section, a very small value of α may produce some performance degradation with respect to the optimal ML detection. We can set the parameter α according to the maximum number of t for every specific application. The minimum value of t is one and its maximum value depends on the constellation from which the signals are selected. For example, in a two dimensional 64-QAM constellation, the maximum value of t is 8 (i.e. [–7, –5,–3,–1,+1,+3,+5,+7]).

The parameter β determines the rate of decrease of parameter k with respect to t. Decreasing β will increase the speed of the algorithm. But like α, if the parameter β is decreased too much, it may produce some errors and hence the performance will degrade. The optimal values of these parameters can be determined by simulations for any particular application.

Since the SD algorithm is an iterative algorithm and the number of iterations is equal to the number of visited points, the number of visited points in the algorithm before reaching to the final solution can be a criterion for the complexity of the algorithm. In the proposed method, this number decreases significantly. In addition, it should be noted that our proposed method does not introduce any additional complexity to the original algorithm due to calculating the parameter k, since the various values of k can be saved in a look-up table and then used during the algorithm procedure.

4. SIMULATION RESULTS

As it was stated before, two main applications of sphere decoding algorithm are in the detection of CDMA signals [5] and in MIMO receivers [3]. In this section the performance of the proposed algorithm is evaluated in both systems by computer simulations.

First, we consider a CDMA system with K=5 synchronous and equal power users in a channel with additive Gaussian noise. The spreading sequences are random with length N=15. The information symbols are selected from a 16-QAM constellation. In the receiver, we use the modified algorithm for multiuser detection of CDMA signals. Here, this technique is called the modified sphere multiuser detection (MS-MUD) method. In addition, the MS-MUD technique is compared with the multiuser detection based on the original VB algorithm (shown as S-MUD in the figures), as well as a conventional receiver and decorrelating detector [6]. The simulations have been done for the values of $\alpha = [\infty, 10, 8, 5]$ and $\beta = [8, 5, 2]$. It should be noted that, the case of $\alpha = \infty$ (k=*1*) is identical to the original VB sphere decoder (S-MUD). For other values of α or β, the performances and also the average number of visited points in the algorithm have been obtained.

Fig.1 shows the results of simulations for the above CDMA system. It is seen that in large values of α and β, the performance of MS-MUD is very close to the performance of the S-MUD method and the loss in the performance is negligible. Even for small values of α and β in which the algorithm is very fast, the performance of MS-MUD is much better than the linear decorrelating detector.

Table 1 shows the average number of visited points in the MS-MUD algorithms. This number is averaged on all iterations (of the Monte Carlo simulation) and also on all SNRs in Fig.1. As we said before, this number can be a criterion for the complexity of the algorithm. It is seen that the number of visited points in the MS-MUD technique is decreased significantly with respect to the S-MUD method. For example, for $\alpha = \beta = 8$ the number of visited points is decreased about 20% and for $\alpha = 5,\ \beta = 2$ this reduction is about 90%. In other words, the MS-MUD detector with parameters $\alpha = 5,\ \beta = 2$ has the complexity of about 10% of the original VB detector. It is worth pointing out that, in the above example CDMA system that employs 16-QAM, the optimum ML decoder (which uses comprehensive search) must examine $16^K = 16^{15}$ points in a real lattice to find the transmit vector. However, using the original VB sphere decoding multiuser detector reduces this number to about 70 points and in the MS-MUD method with $\alpha = 5,\ \beta = 2$, this number is decreased to about seven and the performance is still very outstanding.

In the second part of the simulations, we consider a MIMO system with *M=4* antennas at the transmitter and *N=6* antennas at the receiver. Like before, the information symbols are selected from a 16-

QAM constellation. The transfer matrix $\mathbf{H}$ is modeled by independent Gaussian random variables of variance 0.5 per real dimension.

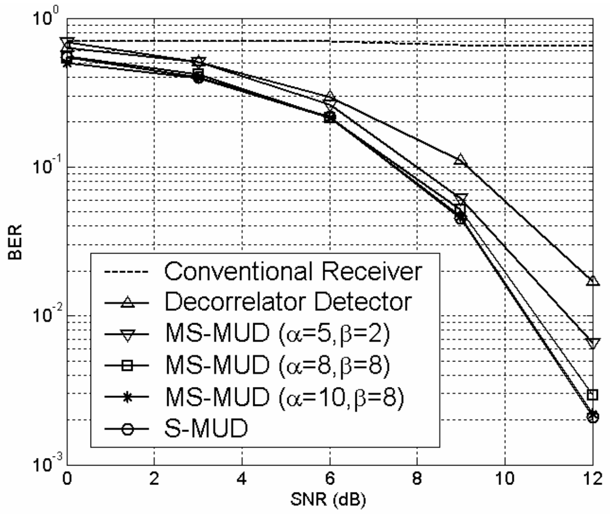

Fig.1. Performance of the MS-MUD multiuser detector in a CDMA system

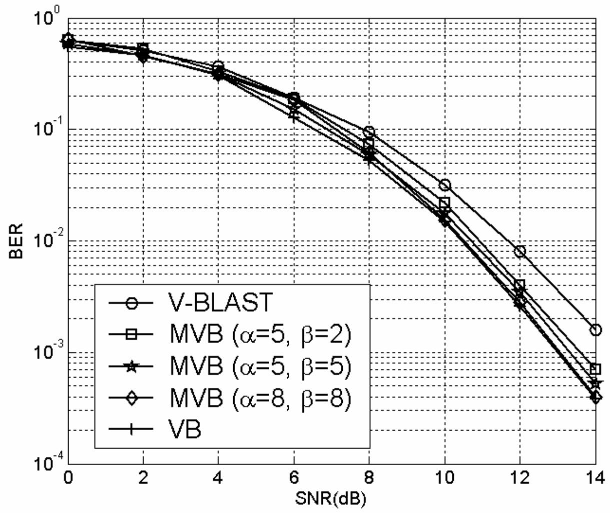

Fig.2. Performance of the MVB detector in a MIMO system

Table 1. The average number of points in the MS-MUD method in the above example of CDMA system

	$\alpha=\infty$	$\alpha=10$	$\alpha=8$	$\alpha=5$
$\beta=8$	75	68	62	20
$\beta=5$	75	56	44	15
$\beta=2$	75	17	14	7

Table 2. The average number of points in the MVB algorithm in the above example of MIMO system

	$\alpha=\infty$	$\alpha=10$	$\alpha=8$	$\alpha=5$
$\beta=8$	9.8	8.7	8.1	4.3
$\beta=5$	9.8	8.2	6.8	2.9
$\beta=2$	9.8	4	2.9	1.5

In the simulations, the performance of the decoder based on the MVB algorithm is compared with the original VB algorithm as well as the V-BLAST method [7]. The simulations have been done for the values of $\alpha=[\infty,10,8,5]$ and $\beta=[8,5,2]$.

Figure 2 shows the BER performance of the MVB in different values of control parameters. We can see that for the large value of α and β, the performance of the MVB is very close to the performance of the original VB algorithm and its performance loss is negligible. For small values of α and β in which the speed of the decoder is very high, the performance of the MVB degrades a little with respect to the VB detector but it is still much better than the V-BLAST method.

Table 2 illustrates the average number of visited points in the MVB algorithm. As mentioned earlier, this number is averaged on all iterations (of the Monte Carlo simulation) and also on all SNRs in Fig. 2.

We can see that the number of visited points is decreased significantly with respect to the original VB sphere decoder ($\alpha=\infty$). For example, for the case of $\alpha=\beta=8$, the number of visited points is decreased about 20% and for $\alpha=5$, $\beta=2$ this reduction is more than 80%. That is, the MVB with these parameter values has a complexity of about 0.1 of the original VB algorithm or it is 10 times faster.

5. CONCLUSION

In this paper, we proposed a modified sphere decoding algorithm based on the Viterbo-Boutros (VB) algorithm. The complexity of the proposed algorithm is much less than the original VB algorithm. Furthermore, in the proposed method the initial value selection of the search radius is not important. As an application for the proposed method, we considered the sphere decoding of CDMA and MIMO systems.

The computer simulation results showed that the proposed method can outperform other sub-optimum detection methods in both CDMA and MIMO systems. We also compared the complexity of the proposed decoder in both systems and showed that its complexity is much less than the original VB sphere decoding algorithm.

REFERENCES

1. Viterbo, E. & Boutros, J. (1999). A universal lattice code decoder for fading channels. *IEEE Trans. Inform. Theory, 45*, 1639-1642.
2. Agrell, E., Eriksson, T., Vardy, A. & Zeger, K. (2002). Closest point search in lattices. *IEEE Trans. Inform. Theory, 48*, 2201-2214.
3. Damen, M. O., El Gamal, H. & Caire, G. (2003). On maximum-likelihood detection and the search for the closest lattice point. *IEEE Trans. on Info. Theory, 49*(10).
4. Hassibi, B. & Vikalo, H. (2005). On sphere decoding algorithm. I. Expected complexity, *IEEE Transactions on Signal Processing, 53*(8), 2806–2818.
5. Brunel, L. & Boutros, J. J. (2003). Lattice decoding for joint detection in direct-sequence CDMA systems. *IEEE Trans. Inform. Theory, 49*, 1030-1037.
6. Verdu, S. (1998). *Multiuser detection*. Cambridge University Press.
7. Wolniansky, P. W., Foschini, G. J., Golden, G. D. & Valenzuela, R. A. (1998). V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel. *Proceeding of International Symposium on Signals, Systems, and Electronics*., New York, NY, USA, 295-300.
8. Horn, R. & Johnson, C. (1989). *Matrix analysis*. Cambridge University Press.